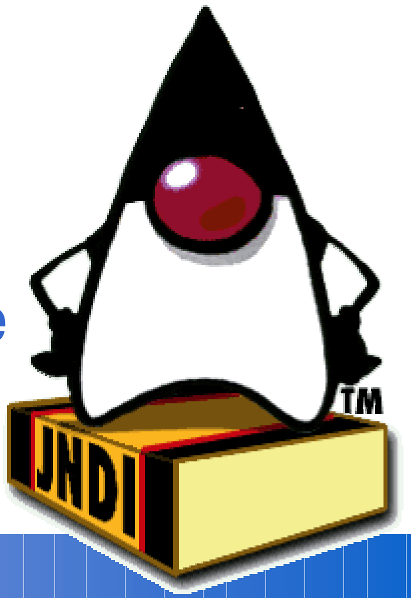
# JNDI

## Java Naming and Directory Interface



Didier DONSEZ

Université Joseph Fourier

IMA – IMAG/LSR/ADELE

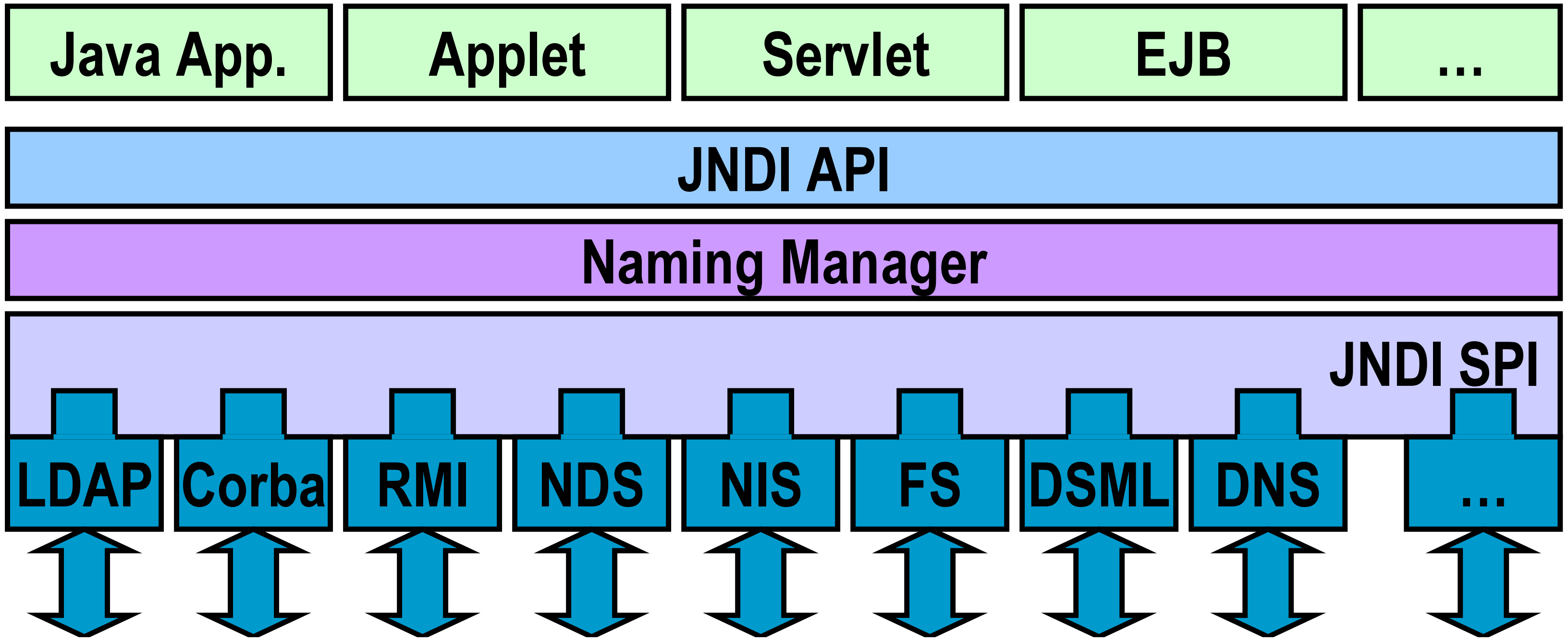
Didier.Donsez@imag.fr

http://www-adele.imag.fr/~donsez

## Sommaire

## JNDI - Principe

- Fournir un API (java) uniforme
  à des services de nommage ou d ’annuaire
  - utilisation de pilotes SPI dynamiquement chargeables
  - LDAP, DNS, NIS, NDS, RMI, CORBA, … et FileSystems
- Architecture



# JNDI - APIs

## Installation

- inclus dans J2 v1.3
- Java Standard Extension dans J1.1 et J2 v1.2

## Packages

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

## SPI : Service Providers

- ens de classes implémentant javax.naming.spi
- SPI préinstallés dans J2 v1.3
  - Lightweight Directory Access Protocol (LDAP)
  - CORBA services (COS) naming service
  - Java Remote Method Invocation (RMI) Registry

## JNDI – ContextFactory (i)

- FileSystem
  - com.sun.jndi.fscontext.FSContextFactory
  - com.sun.jndi.fscontext.RefFSContextFactory
- Lightweight Directory Access Protocol (LDAP)
  - com.sun.jndi.ldap.LdapCtxFactory
- CORBA services (COS) naming service
- Java Remote Method Invocation (RMI) Registry
  - com.sun.jndi.rmi.registry.RegistryContextFactory
- NIS
  - com.sun.jndi.nis.NISCtxFactory
- NDS
  - com.novell.naming.service.nds.NdsInitialContextFactory

## JNDI – ContextFactory (ii)

- DNS
- DSML

## JNDI
## Création du contexte LDAP

```
String login="Directory Manager";
String password="motdepasse";

Hashtable env = new Hashtable();

env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");

env.put(Context.SECURITY_AUTHENTICATION, "simple");
env.put(Context.SECURITY_PRINCIPAL, "cn="+login);
env.put(Context.SECURITY_CREDENTIALS, password);

env.put(Context.PROVIDER_URL, "ldap://localhost:389/o=JNDITutorial");

Context ctx = new InitialContext(env);
```

# JNDI
# Exemple 1 avec LDAP

```
Attributes answer = ctx.getAttributes("cn=Ted Geisel, ou=People");
for (NamingEnumeration ae = answer.getAll(); ae.hasMore();) {
        Attribute attr = (Attribute)ae.next();
        System.out.println("attribute: " + attr.getID());
        /* print each value */
        for (NamingEnumeration e = attr.getAll(); e.hasMore();
           System.out.println("value: " + e.next()));
    }
```

# JNDI
## Exemple 1 avec LDAP

```
# java GetattrsAll
        attribute: sn
        value: Geisel
        attribute: objectclass
        value: top
        value: person
        value: organizationalPerson
        value: inetOrgPerson
        attribute: jpegphoto
        value: [B@1dacd78b
        attribute: mail
        value: Ted.Geisel@JNDITutorial.com
        attribute: facsimiletelephonenumber
        value: +1 408 555 2329
        attribute: telephonenumber
        value: +1 408 555 5252
        attribute: cn
        value: Ted Geisel
```

# JNDI
# Exemple 2 avec LDAP

```
// Set up environment for creating initial context
Hashtable env = new Hashtable();
env.put(Context.INITIAL_CONTEXT_FACTORY, "com.sun.jndi.ldap.LdapCtxFactory");
env.put(Context.PROVIDER_URL, "ldap://localhost:389/o=JNDITutorial");
Context ctx = new InitialContext(env);
// Specify the ids of the attributes to return
String[] attrIDs = {"sn", "telephonenumber", "golfhandicap", "mail"};
// Get the attributes requested
Attributes answer = ctx.getAttributes("cn=Ted Geisel, ou=People", attrIDs);
for (NamingEnumeration ae = answer.getAll(); ae.hasMore();) {
        Attribute attr = (Attribute)ae.next();
        System.out.println("attribute: " + attr.getID());
        /* print each value */
        for (NamingEnumeration e = attr.getAll(); e.hasMore();
          System.out.println("value: " + e.next()));
    }
```

## JNDI
## Exemple 2 avec LDAP

```
# java Getattrs
        attribute: sn
        value: Geisel
        attribute: mail
        value: Ted.Geisel@JNDITutorial.com
        attribute: telephonenumber
        value: +1 408 555 5252
```

# JNDI
# Exemple 3 avec LDAP

```
// Specify the changes to make
        ModificationItem[] mods = new ModificationItem[3];
// Replace mail attribute with new value
        mods[0] = new ModificationItem(DirContext.REPLACE_ATTRIBUTE,
           new BasicAttribute("mail", "geisel@wizards.com"));
// Add additional value to "telephonenumber"
        mods[1] = new ModificationItem(DirContext.ADD_ATTRIBUTE,
           new BasicAttribute("telephonenumber", "+1 555 555 5555"));
// Remove jpegphoto
        mods[2] = new ModificationItem(DirContext.REMOVE_ATTRIBUTE,
           new BasicAttribute("jpegphoto"));
// Perform requested modifications on named object
        ctx.modifyAttributes(name, mods);
```

## JNDI Exemple 4 avec LDAP Basic Search

```
// Specify the attributes to match
// Ask for objects with a surname ("sn") attribute with value "Geisel"
// and which have the "mail" attribute.
        Attributes matchAttrs = new BasicAttributes(true); // ignore attribute name case
        matchAttrs.put(new BasicAttribute("sn", "Geisel"));
        matchAttrs.put(new BasicAttribute("mail"));

// Search for objects with those matching attributes
        NamingEnumeration answer = ctx.search("ou=People", matchAttrs);

        while (enum.hasMore()) {
           SearchResult sr = (SearchResult)enum.next();
           System.out.println(">>>" + sr.getName());
           printAttrs(sr.getAttributes());
        }
```

## JNDI Exemple 5 avec LDAP Search Filter

```
// Create default search controls
        SearchControls ctls = new SearchControls();

// Specify the search filter to match
// Ask for objects with attribute sn == Geisel and which have the "mail" attribute.
        String filter = "(&(sn=Geisel)(mail=*))";

// Search for objects using filter
        NamingEnumeration answer = ctx.search("ou=People", filter, ctls);
```

# JNDI Exemple 5 avec LDAP Search Filter

```
// Specify the ids of the attributes to return
        String[] attrIDs = {"sn", "telephonenumber", "golfhandicap", "mail"};
// Specify the search control
        SearchControls ctls = new SearchControls();
        ctls.setTimeLimit(1000); // limit to 1000 ms
        ctls.setReturningAttributes(attrIDs);
        ctls.setSearchScope(SearchControls.SUBTREE_SCOPE);

// Specify the search filter to match
// Ask for objects with attribute sn == Geisel and which have the "mail" attribute.
        String filter = "(&(sn=Geisel)(mail=*))";

// Search subtree for objects using filter
        NamingEnumeration answer = ctx.search("", filter, ctls);
```

## Symboles de Filtrage

| Symbol | Description |
|---|---|
| & | conjunction (i.e., and -- all in list must be true) |
| \| | disjunction (i.e., or -- one or more alternatives must be true) |
| ! | negation (i.e., not -- the item being negated must not be true) |
| = | equality (according to the matching rule of the attribute) |
| ~= | approximate equality (according to the matching rule of the attribute) |
| >= | greater than (according to the matching rule of the attribute) |
| <= | less than (according to the matching rule of the attribute) |
| =* | presence (i.e., entry must have the attribute but its value is irrelevant) |
| * | wildcard (indicates zero or more characters can occur in that position) This is to be used when specifying attribute values to match. |
| \ | escape (for escaping '*', '(', or ')' when they occur inside an attribute value) |

## DSML et JNDI

- Provider (SPI) JNDI / DSML
  - Accéder à des documents DSML
  - Manipuler et modifier leur contenu
  - Ré-exporter le contenu en DSML

## JNDI et J2EE

- TODO

## Exemple JNDI

- Navigateur-Editeur LDAP de Jarek Gawor
  - http://www.iit.edu/~gawojar/ldap
  - Pur Java et JNDI
- Tutorial JNDI
  - http://java.sun.com/products/jndi/tutorial/TOC.html
- Passerelle DSML
  - http://www.worldspot.com/dsmlgw-xml-rpc/DSMLGateway.html

## Bibliographie

- Spécifications et Tutorial JNDI
  - http://java.sun.com/products/jndi
  - http://java.sun.com/products/jndi/tutorial/TOC.html
- Rosanna Lee, Scott Seligman , "JNDI API Tutorial and Reference: Building Directory-Enabled Java Applications (The Java Series)", (May 30, 2000) , Ed Addison-Wesley Pub Co; ISBN: 0201705028
- David Flanagan, Jim Farley, William Crawford & Kris Magnusson, « Java Enterprise in a Nutshell, A Desktop Quick Reference », Edition Oreilly, 1st Edition September 1999, ISBN 1-56592-483-5
  - Chapter 6
- Andrew Patzer , "Programmation Java côté serveur : Servlets, JSP et EJB", Ed Eyrolles-Wrox, 2000, ISBN 1-861002-77-7 (sources des exemples sur www.wroxfrance.com)
  - chapitres 21 et 22
- ROB WELTMAN and TONY DAHBURA, LDAP Programming with Java, ADDISON–WESLEY, 2000, ISBN 0-201-65758-9