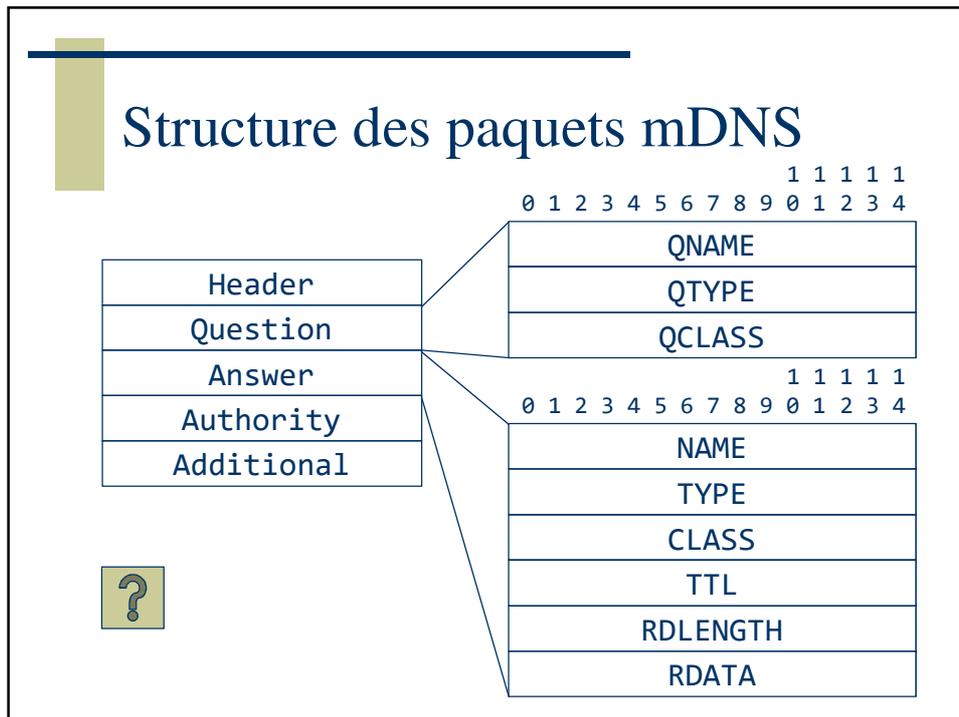

mDNS-SD

- ◆ Le protocole DNS multidiffusion (mDNS) résout les noms d'hôte en adresses IP au sein de petits réseaux qui n'incluent pas de serveur de noms local.
- ◆ mDNS peut fonctionner en conjonction avec DNS Service Discovery (DNS-SD), une technique de mise en réseau complémentaire sans configuration spécifiée dans la RFC 6763.
- ◆ Ce protocole utilise le port 5353

mDNS-SD

- ◆ Ce protocole utilise des paquets IP UDP et est implanté par les logiciels Apple Bonjour et Avahi open source, inclus dans la plupart des distributions Linux.
- ◆ Bien que l'implantation de Windows 10 se soit limitée à la découverte d'imprimantes en réseau, les versions ultérieures ont également résolu les noms d'hôte.



- ## mDNS-SD
- ◆ mDNS utilise...
 - le port 5353
 - l'adresse de groupe

224.0.0.251	en IPv4
FF02::FB	en IPv6

 - ◆ Modifier le programme MulticastReceiver pour afficher QNAME et tester...

mDNS-SD

- ◆ Quelques machines d'un réseau local proposent un service.
- ◆ Chaque machine du réseau possède une adresse IP (attribuée par le serveur DHCP)
- ◆ Comment découvrir les services disponibles, à quelle adresse, sur quel port ?
- ◆ Solution : diffusion multicast de l'adresse IP, du port...

Bonjour, Zeroconf

- ◆ Tout débute avec le protocole Bonjour d'Apple dans le réseau AppleTalk.
- ◆ Une standardisation apparaît plus tard avec ZeroConf (Zero Configuration Networking), appellation générique d'un ensemble de protocoles permettant de créer automatiquement un réseau IP utilisable sans configuration particulière ou serveurs dédiés.

implantation en Java : JmDNS

- ◆ ServiceInfo
Classe java associée au paquet DNS
- ◆ Découverte des services par un thread d'écoute des trames au bon protocole
- ◆ Enregistrement de services par diffusion

Enregistrement de type SRV

voir http://fr.wikipedia.org/wiki/Enregistrement_de_service

- ◆ Service : _http
- ◆ Protocole : _tcp ou _udp
- ◆ Nom.de.domaine
- ◆ TTL : durée de validité de la réponse en secondes
- ◆ Classe : IN pour internet
- ◆ Priorité : priorité du serveur cible
valeur faible priorité élevée
- ◆ Poids : poids relatif donné pour les services identiques
- ◆ Port : numéro de port
- ◆ Cible : nom du serveur qui fournit le service

JmDNS

JmDNS 3.4.1

[All Classes](#)

Packages

- [javax.jmdns](#)
- [javax.jmdns.impl](#)
- [javax.jmdns.impl.constants](#)
- [javax.jmdns.impl.tasks](#)
- [javax.jmdns.impl.tasks.resolver](#)
- [javax.jmdns.impl.tasks.state](#)

[javax.jmdns](#)

Interfaces

- [JmDNS.Delegate](#)
- [JmDNS](#)
- [JmDNS.Factory.ClassDelegate](#)
- [NetworkTopologyDiscovery](#)
- [NetworkTopologyDiscovery.Factory.ClassDelegate](#)
- [NetworkTopologyListener](#)
- [ServiceListener](#)
- [ServiceTypeListener](#)

Classes

- [JmDNS](#)
- [JmDNS.Factory](#)
- [NetworkTopologyDiscovery.Factory](#)
- [NetworkTopologyEvent](#)
- [ServiceEvent](#)
- [ServiceInfo](#)

Enums

- [ServiceInfo.Fields](#)

Un schéma de programmation

- ◆ Initialisation
 - JmDNS jmdsn = JmDNS.create();
- ◆ Découverte de service
 - jmdsn.addServiceListener("_http._tcp.local.", new MyListener());

Un écouteur(observateur) avec les méthodes

 - void serviceAdded(ServiceEvent event)
 - void serviceRemoved(ServiceEvent event)
 - void serviceResolved(ServiceEvent event)

Exemple de Listener

```
class SampleListener implements ServiceListener {
    @Override
    public void serviceAdded(ServiceEvent event) {
        System.out.println("Service added : " + event.getName() +
            "." + event.getType());
    }

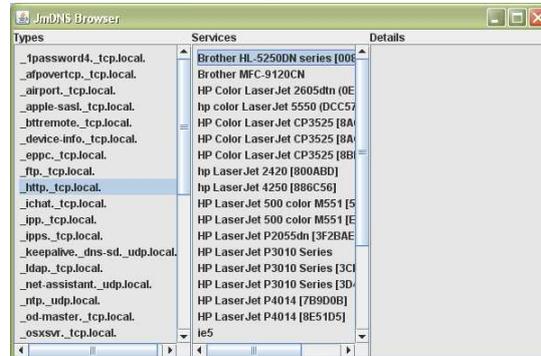
    @Override
    public void serviceRemoved(ServiceEvent event) {
        System.out.println("Service removed : " + event.getName() +
            "." + event.getType());
    }

    @Override
    public void serviceResolved(ServiceEvent event) {
        System.out.println("Service resolved: " + event.getInfo());
    }
}
```

Exemple de Listener...

- ◆ Avec "_http._tcp.local.", donne la liste de tous les serveurs web du réseau local avec leur adresse IP.
java -cp .;..\jmdns.jar DiscoverServices
java -cp .;..\jmdns.jar HttpServer8080
- ◆ Avec un navigateur, on peut s'y connecter pour le vérifier.

Navigateur prêt à l'emploi



- ◆ par
 - Arthur van Hoff
 - Werner Randelshofer
- ◆ source
 - <https://github.com/jadahl/jmdns/tree/master/src>

Navigateur prêt à l'emploi

- ◆ Compilation


```
javac -cp .;..\jmdns.jar Browser.java
```
- ◆ Lancement du navigateur


```
java -cp .;..\jmdns.jar Browser
```
- ◆ Choisir `_http._tcp.local.`
- ◆ Lancement d'un service


```
java -cp .;..\jmdns.jar HttpServer8080
```

Initialisation de JmDNS

```
JmDNS jmdns = null;
try {
    jmdns = JmDNS.create();
    jmdns = JmDNS.create(InetAddress.getLocalHost());
    jmdns = JmDNS.create(InetAddress.getLocalHost(), "unNom");
    ...
    ...
    ...
    ...
} finally {
    jmdns.close();
}
```

Quels services ? ServiceInfo

```
// en mode synchrone
// ServiceInfo[] services = jmdns.list("_http._tcp.local.");

try {
    jmdns = JmDNS.create();
    ServiceInfo[] services = jmdns.list("_http._tcp.local.");
    for(ServiceInfo si : services){
        System.out.println(si.getURLs()[0]);
    }
} finally {
    if (jmdns != null) jmdns.close();
}
```

<http://jmdns.sourceforge.net/apidocs/index.html>

<http://jmdns.sourceforge.net/apidocs/javax/jmdns/ServiceInfo.html>

Quels types ? ServiceTypeListener

```
JmDNS jmdns = JmDNS.create();
// en mode asynchrone
// jmdns.addServiceTypeListener(new TypeListener());

static class TypeListener implements ServiceTypeListener {
    @Override
    public void serviceTypeAdded (ServiceEvent event) {
        System.out.println("Service type added: " +
            event.getType());
    }

    @Override
    public void subTypeForServiceTypeAdded (ServiceEvent event) {
        System.out.println("SubType for service type added: " +
            event.getType());
    }
}
```

Publication, création d'un service

```
JmDNS jmdns = JmDNS.create();

ServiceInfo info;
String type = "_http._tcp.local." ;
String nom = "un_nom_unique_de_service",
info = ServiceInfo.create(type, nom, 8080, "index.html");

jmdns.registerService(info);

// paramètres supplémentaires
Map<String, String> map = new HashMap<>();
map.put("cle", "valeur");
info = ServiceInfo.create(type, "un_nom_de_service", 8080, map);

jmdns.registerService(info);
```

Discussion

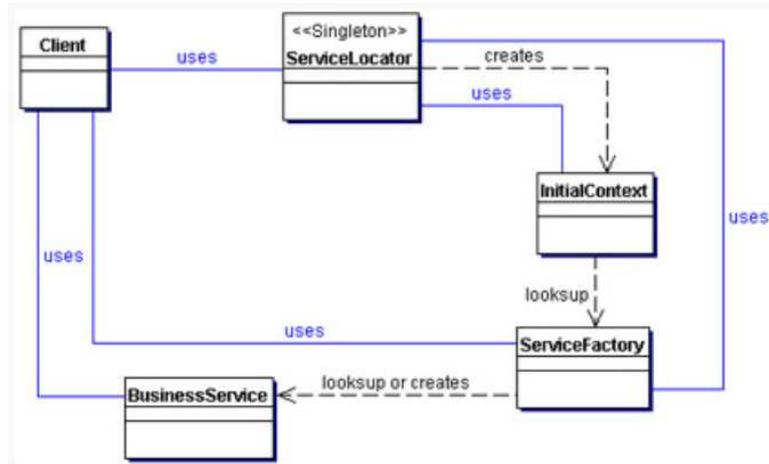
- ◆ Des thermomètres connectés offrent un service pour obtenir la température.
- ◆ mDNS leur permet de publier leur adresse IP.
- ◆ Afin d'éviter une connexion ultérieure en TCP, pourraient-ils publier la température ?

Patron ServiceLocator

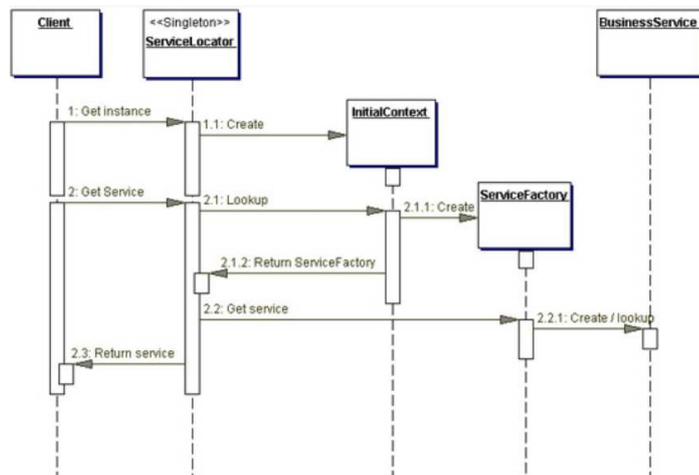
- ◆ La recherche et la création de services dans un réseau nécessitent des interfaces complexes. De multiples API sont utilisées.
- ◆ Un ServiceLocator se comporte comme une façade et masque la complexité sous-jacente.

voir <https://www.oracle.com/java/technologies/service-locator.html>

Patron ServiceLocator



Patron ServiceLocator



Un schéma de ServiceLocator

```
public final class ServiceLocator implements ...{
    private Map<String,XXX> map;
    private static ServiceLocator instance;
    static{
        instance = new ServiceLocator(); // cf. Singleton
    }
    public static ServiceLocator getInstance(){
        return instance;
    }
    public Object lookup(String name) throws Exception{
        XXX data = map.get(name);
        ...
        return data;
    }
    private ServiceLocator() {
        this.map = Collections.synchronizedMap(
            new HashMap<String, XXXX >()); // concurrence
    }
}
```

Un schéma de ServiceLocator...

```
private class XXXXListener implements ServiceListener {

    public void serviceAdded(ServiceEvent event) {
        map.put(event.getName(), xxxx);
    }

    public void serviceRemoved(ServiceEvent event) {
        map.remove(event.getName(), xxxx);
    }

    public void serviceResolved(ServiceEvent event) {
        // accès aux info, via event
    }
}
```

Internet des objets

- ◆ Découverte d'un webthing : mDNS-SD
- ◆ Accès au webthing : http:// (JSON)
- ◆ voir <https://webthings.io/about/>
- ◆ voir <https://webthings.io/framework/>
- ◆ voir <https://webthings.io/api/>

Références

- ◆ <https://lig-membres.imag.fr/donsez/dev/osgi/jmdns/api/fr/imag/adele/service/jmdns/JmDNS.html>
- ◆ <http://jmdns.sourceforge.net/apidocs/index.html>