

AULA 09

Descoberta de Serviços

Prof. Windson Viana de Carvalho

DESCOBERTA DE SERVIÇO - CONCEITO



Encontrar de forma automática ou semi-automática serviços, recursos e/ou dispositivos utilizando uma rede de computadores

DESCOBERTA DE SERVIÇO - CONCEITO

Princípios Fundamentais

Interface de descrição

Serviço de publicação

Método de consulta e mapeamento

Invocação final

Service Discovery Protocol

- Protocolo responsável por definir o modelo de publicação, consulta e ligação final entre os serviços

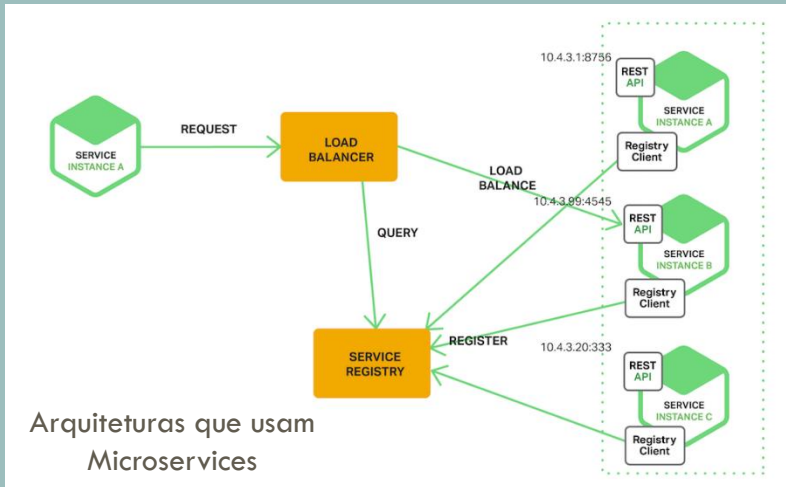
QUAL É O PROBLEMA CENTRAL?

```
public void connectToMyServer(){  
    String URL = "http://myserverXY.com"  
    ....  
}  
...
```

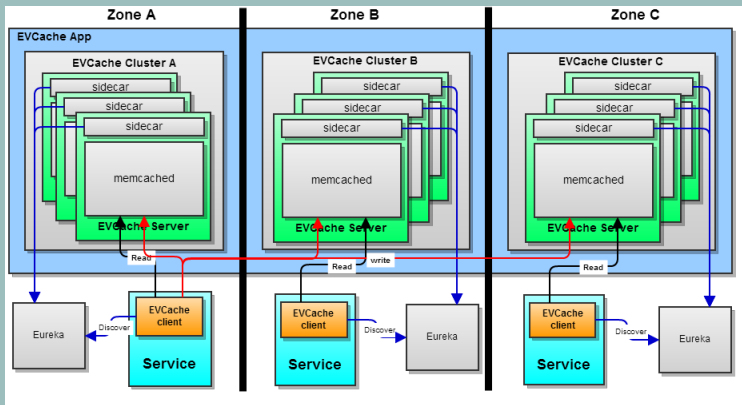
```
public void  
connectToGameServerOnOtherDevice(){  
    String DeviceIP = "200.19.190.1"  
    ....  
}
```

TRANSPARÊNCIA DE LOCALIZAÇÃO!

DESCOBERTA DE SERVIÇO - CENÁRIOS



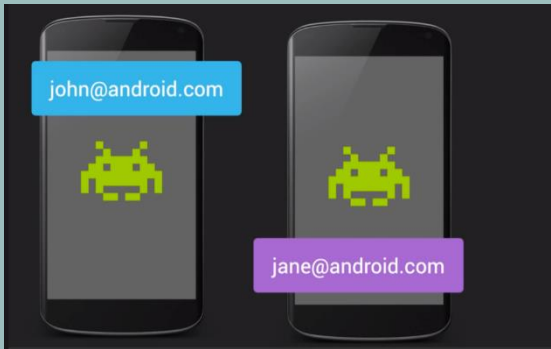
DESCOBERTA DE SERVIÇO - CENÁRIOS



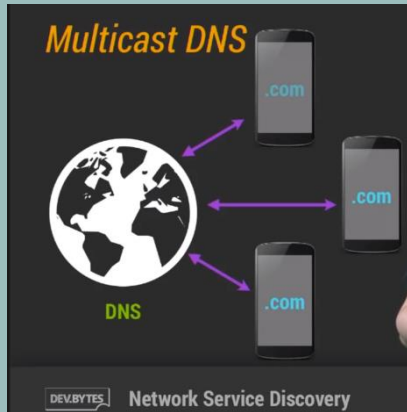
Múltiplos Servidores

NETFLIX
OSS

DESCOBERTA DE SERVIÇO - CENÁRIOS



Descoberta de dispositivos em redes locais



DESCOBERTA DE SERVIÇO - CENÁRIOS



Cenários Voláteis - Smart Home

PRINCIPAIS ELEMENTOS DA DESCOBERTA DE SERVIÇO

Arquitetura de descoberta de serviços

Escopo da descoberta de serviços

Técnicas de descrição de serviços

Mecanismo de consulta às informações de serviços

PRINCIPAIS ELEMENTOS DA DESCOBERTA DE SERVIÇO

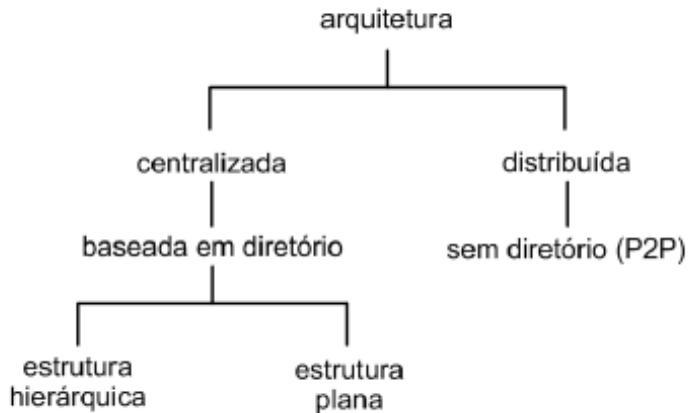
Mecanismos de requisição e anúncio de serviços

Armazenamento das informações de serviços

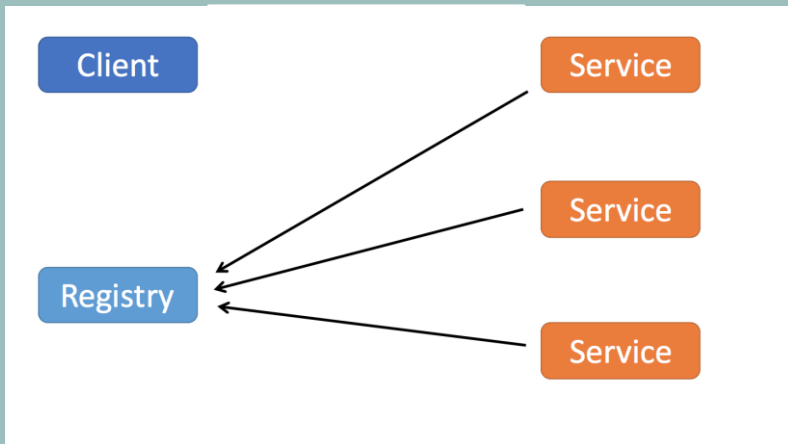
Mecanismos de seleção e invocação de serviços

Mecanismos para oferecer suporte à mobilidade e segurança

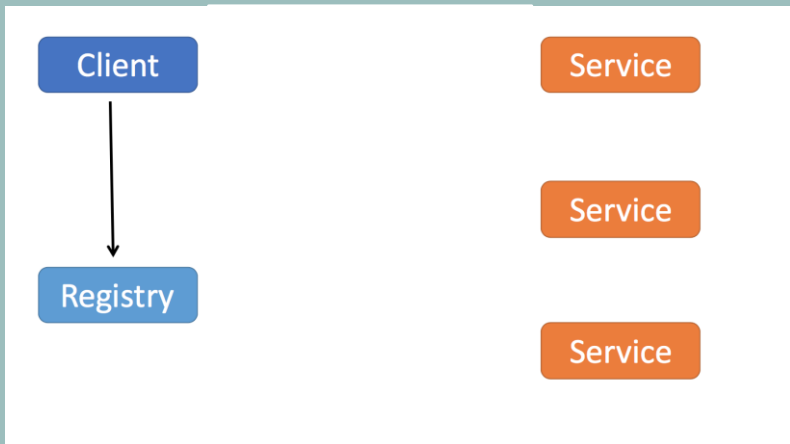
ARQUITETURA



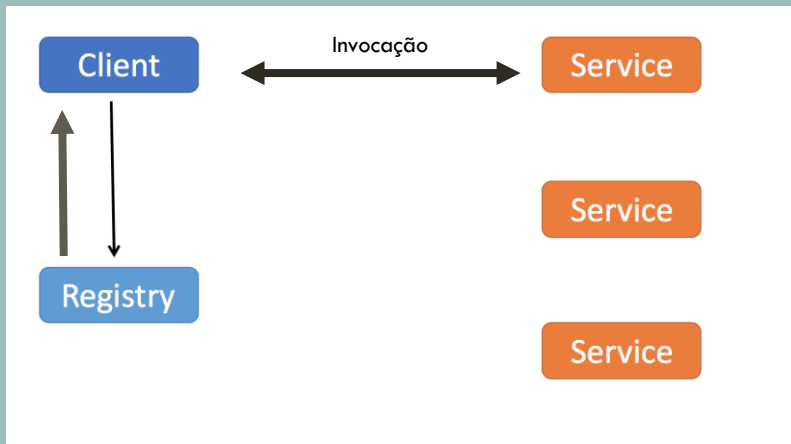
MODELO CENTRALIZADO — BASEADO EM DIRETÓRIO



MODELO CENTRALIZADO — BASEADO EM DIRETÓRIO



MODELO CENTRALIZADO — BASEADO EM DIRETÓRIO

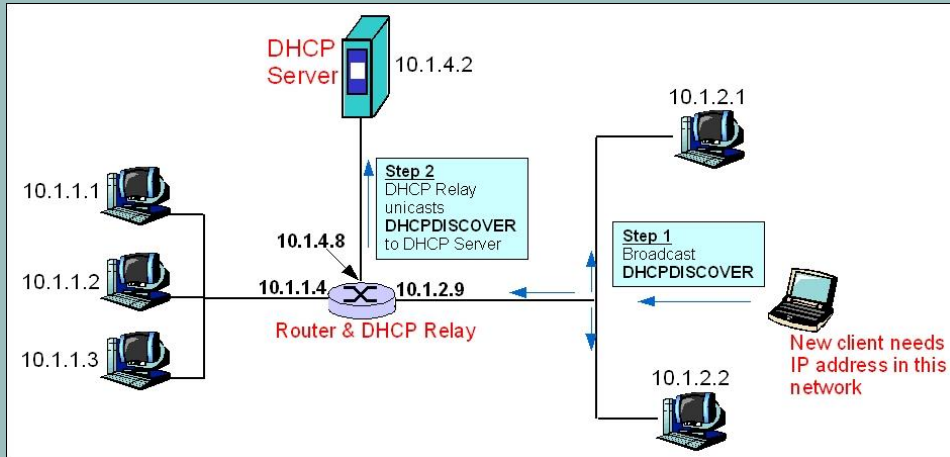


COMO ENCONTRAR O REGISTRO?

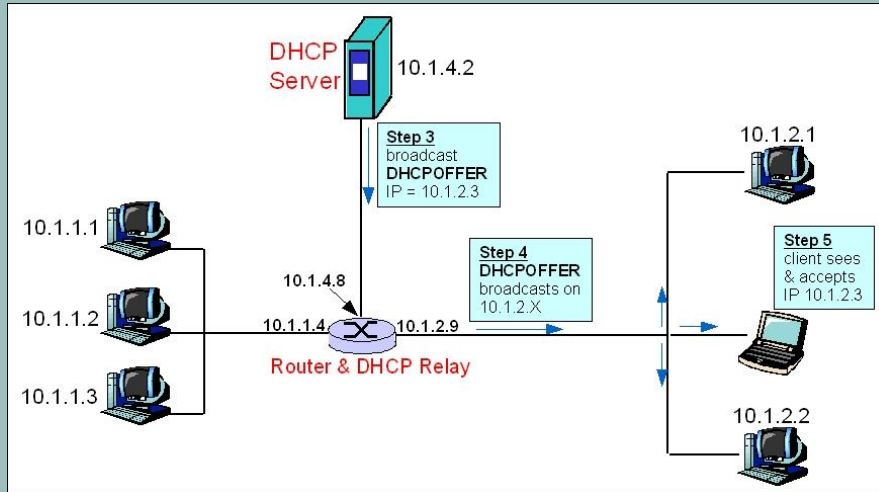
DHCP — EXEMPLO



DHCP — EXEMPLO



DHCP — EXEMPLO

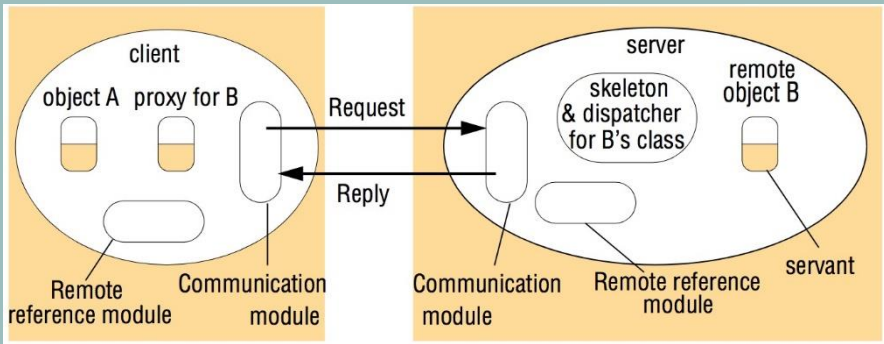


EXEMPLOS DE DESCOBERTA CENTRALIZADA

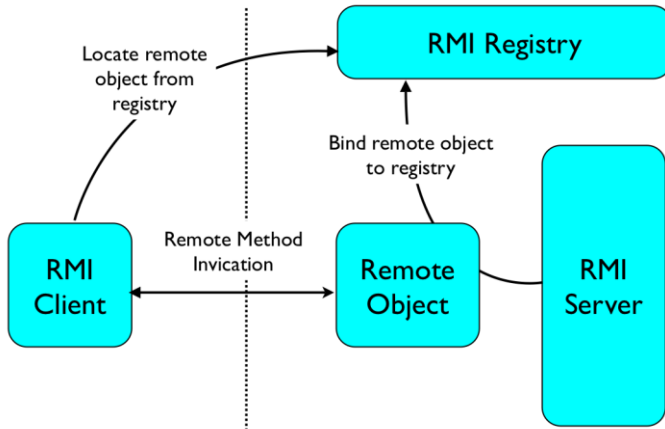


RMI — REMOTE METHOD INVOCATION

Vários objetos e módulos separados estão envolvidos na realização de uma invocação a método remoto.



DESCOBERTA DE SERVIÇO - RMI



UNINDO AS PARTES — INTERFACE REMOTA

```
//Classe compartilhada pelo
cliente e pelo servidor
package example.hello;
import java.rmi.Remote;
import
java.rmi.RemoteException;
public interface Hello extends
Remote {
    String sayHello() throws
RemoteException;
}
```

```
//Objeto remoto real
public class Server implements
Hello {
    public Server() {}
    public String sayHello() {
        return "Hello, best students are
in SMD!";
    }
....
```

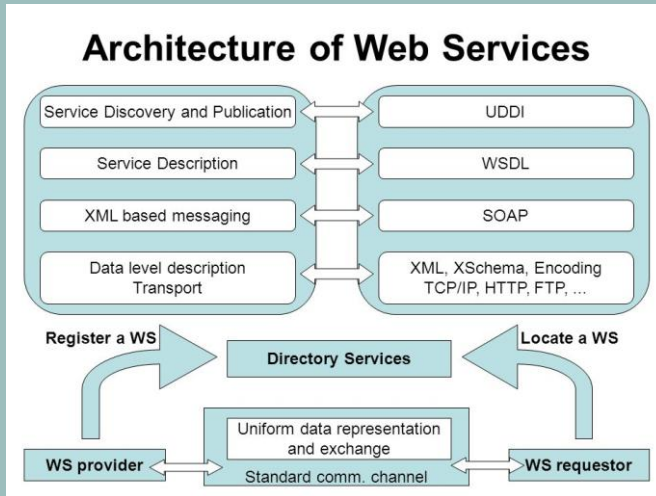
UNINDO AS PARTES - SERVIDOR

```
public static void main(String args[]) {  
    Server obj = new Server();  
    Hello stub = (example.hello.Hello)  
    UnicastRemoteObject.exportObject(obj,0);  
    // Bind the remote object's stub in the registry  
    Registry reg=null;  
    try {  
        reg = LocateRegistry.createRegistry(1099);  
    } catch(Exception e){}  
    reg.rebind("HelloService", stub);
```

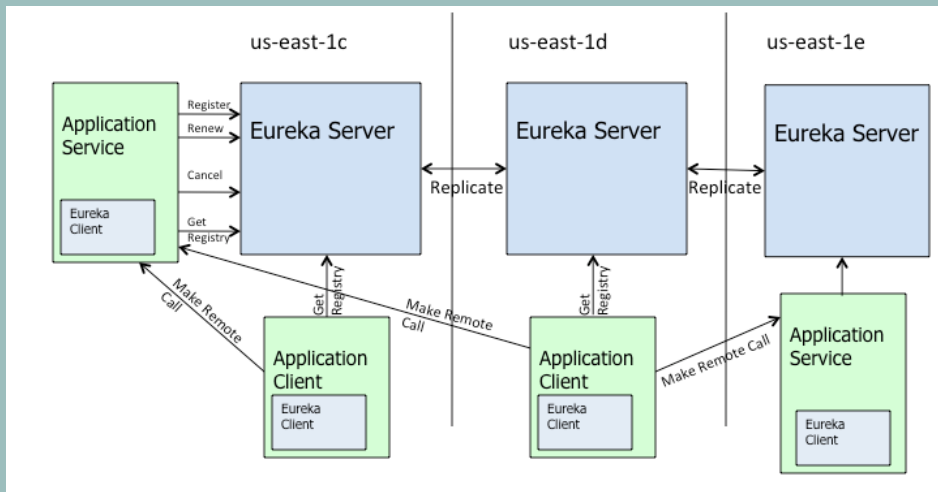
UNINDO AS PARTES - CLIENTE

```
public static void main(String[] args) {  
    String host="localhost";  
    try {  
        Registry registry = LocateRegistry.getRegistry(host);  
        Hello stub = (Hello) registry.lookup("HelloService");  
        if (stub!=null){  
            String response = stub.sayHello();  
            System.out.println("response: " + response);  
        }  
    } catch (Exception e) { }}}
```


DESCOBERTA DE SERVIÇO — WEB SERVICES W3C



EUREKA — NETFLIX OSS



EUREKA — NETFLIX OSS

Eureka é um serviço baseado em REST (Representational State Transfer)

Nuvem da AWS

Localizar serviços com o objetivo de balanceamento de carga

Failover de servidores de camada intermediária.

Fonte: <https://github.com/Netflix/eureka/wiki/Eureka-at-a-glance>

ABORDAGENS DESCENTRALIZADAS?

TAREFA 01

Quais cenários seriam interessantes de possuir uma abordagem descentralizada?

Como solucionar o problema do registro central?

Onde os recursos/serviços iriam se registrar?

Qual seria o mecanismo de endereçamento dos serviços/recursos?

Como seria a descrição do serviço?

Como funcionaria a consulta?

UPNP - Universal Plug and Play

- Extensão do modelo Plug and Play da Microsoft para dispositivos de uma rede local
- Protocolo de descoberta de serviços orientado a dispositivos.
 - Descrição do serviço no formato XML
- Uso do Simple Service Discovery Protocol para descoberta e anúncio na rede
 - Multicast UDP no endereço 239.255.255.250, porta 1900

UPNP — Universal Plug and Play

Entrada do Dispositivo na Rede

```
NOTIFY * HTTP/1.1
Host:239.255.255.250:1900
Cache-control:max-age=1800
Location:http://192.168.0.1:49152/des.xml
Nt:upnp:rootdevice
Nts:ssdp:alive
Usn:uuid:de5d6118-bfcb-918e-0000-00001eccef34::upnp:rootdevice
```

DESCRIÇÃO DO SERVIÇO/DISPOSITIVO

```
<?xml version="1.0" encoding="utf-8"?>
<root xmlns="urn:schemas-upnp-org:device-1-0">
  <specVersion>
    <major>1</major>
    <minor>1</minor>
  </specVersion>

  <device>
    <deviceType>urn:schemas-upnp-org:device:BinaryLight:1</deviceType>
    <friendlyName>Kitchen Lights</friendlyName>
    <manufacturer>OpenedHand</manufacturer>
    <modelName>Virtual Light</modelName>
    <UDN>uuid:cc93d8e6-6b8b-4f60-87ca-228c36b5b0e8</UDN>

    <serviceList>
      <service>
        <serviceType>urn:schemas-upnp-org:service:SwitchPower:1</serviceType>
        <serviceId>urn:upnp-org:serviceId:SwitchPower:1</serviceId>
        <SCPDURL>/SwitchPower1.xml</SCPDURL>
        <controlURL>/SwitchPower/Control</controlURL>
        <eventSubURL>/SwitchPower/Event</eventSubURL>
      </service>
    </serviceList>
  </device>
</root>
```


REQUISIÇÃO SOAP PARA CONTROLE

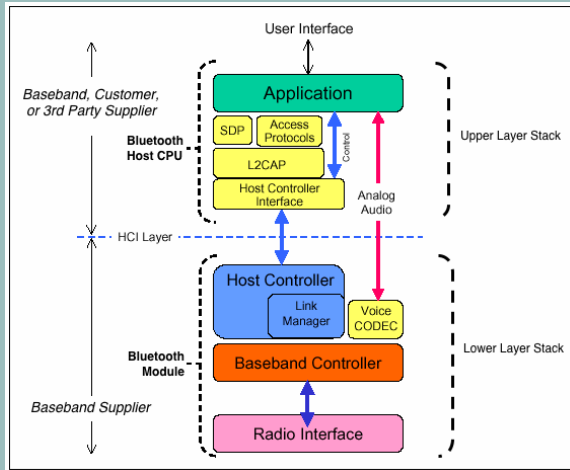
```
POST /control/url HTTP/1.1
HOST: hostname:portNumber
CONTENT-TYPE: text/xml; charset="utf-8"
CONTENT-LENGTH: length of body
USER-AGENT: OS/version UPnP/1.1 product/version
SOAPACTION: "urn:schemas-upnp-org:service:serviceType:v#actionName"
```

```
<?xml version="1.0"?>
<s:Envelope
  xmlns:s="http://schemas.xmlsoap.org/soap/envelope/"
  s:encodingStyle="http://schemas.xmlsoap.org/soap/encoding/">
  <s:Body>
    <u:actionName xmlns:u="urn:schemas-upnp-org:service:serviceType:v">
      <argumentName>in arg value</argumentName>
    </u:actionName>
  </s:Body>
</s:Envelope>
```

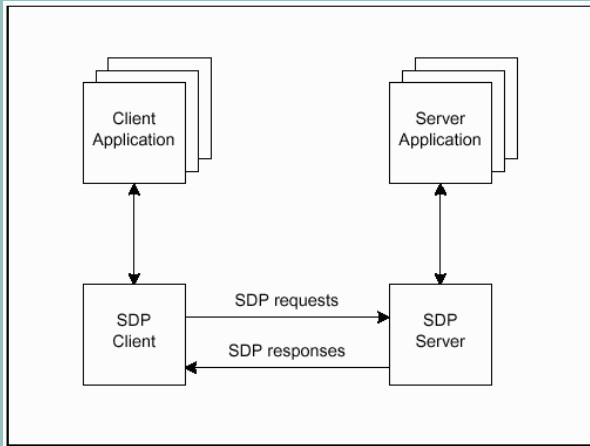


Nome e argumentos do serviço a ser invocado

DESCOBERTA DE SERVIÇO — BLUETOOTH SERVICE DISCOVERY PROTOCOL (SDP)



DESCOBERTA DE SERVIÇO — BLUETOOTH SERVICE DISCOVERY PROTOCOL (SDP)



TAREFA 02 — SOBRE O BLUETOOTH SERVICE DISCOVERY

Como funciona o processo de registro e busca do SDP ?

O que é um UUID?

Como funciona o processo de busca no Bluetooth BLE ?

O que difere os atributos ATT e UUID?

ZEROCONF — O QUE É?

Created in 1999 by the group IETF (Internet Engineering Task Force), the Zero Configuration Networking (Zeroconf) is a methodology and a special set of technologies that enable the configuration of a network and discovery of services in a simple way that an average user will not notice.

- ☐ Atribuição de endereços IP (sem um DHCP server)
- ☐ Resolução de nomes (sem um DNS server)
- ☐ Busca e listagem de serviços (sem um serviço de diretórios)
- ☐ Atribuição de endereços IP multicast, se necessário

ZEROCONF — IMPLEMENTAÇÕES



Apple Bonjour



Android Network Service
Discovery
(parcialmente)

ZEROCONF — TECNOLOGIAS DE DESCOBERTA

- ☐ IPv4 Link-Local
- ☐ Multicast DNS
- ☐ DNS Service Discovery

ZEROCONF — IPV4 LINK-LOCAL ADDRESS

- ☐ Sem servidor central
- ☐ Cada dispositivo escolhe seu próprio endereço



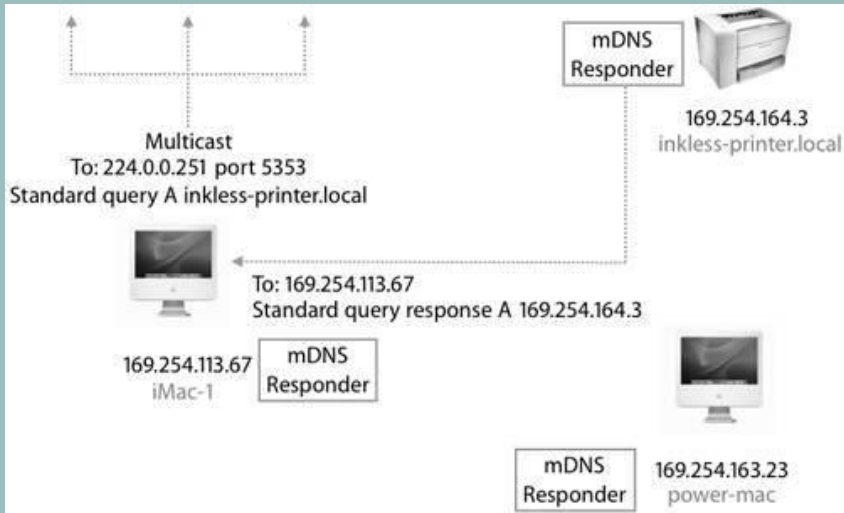
ZEROCONF — IPV4 LINK-LOCAL ADDRESS

- ☐ Endereço randômico entre 169.254.1.0 e 169.254.254.255
- ☐ IPv4LL usa ARP (Address Resolution Protocol) para checar se o IP está livre

ZEROCONF — MULTICAST DNS (MDNS)

- ☐ Sem servidor central de DNS
- ☐ Cada host escolhe seu próprio nome
 - Domínio .local é um TLD reservado para o ZeroConf
- ☐ Uso do MDNS Responder
- ☐ Consultas para saber do uso do nome e para anúncios
 - Broadcast
 - MDNS usa um grupo multicast 224.0.0.251

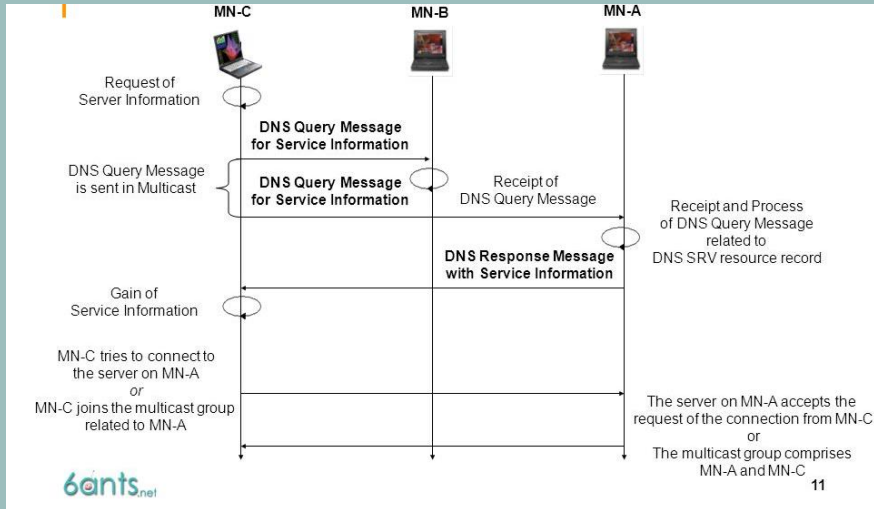
ZEROCONF — MULTICAST DNS (MDNS)



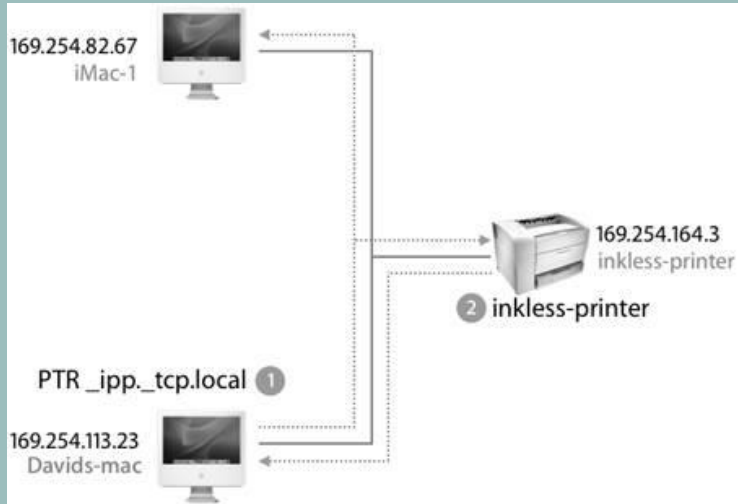
DNS — SERVICE DISCOVERY (DNS-SD)

- ☐ Busca automática de serviços na rede
 - ☐ uso MDNS ou consultas clássicas
- ☐ Mensagens do DNS-SD tem o mesmo formato de consultas DNS (UDP) padrão
- ☐ Consultas são do tipo SRV, PTR, A and TXT

PROCESSO DE DESCOBERTA



DNS — SERVICE DISCOVERY (DNS-SD)



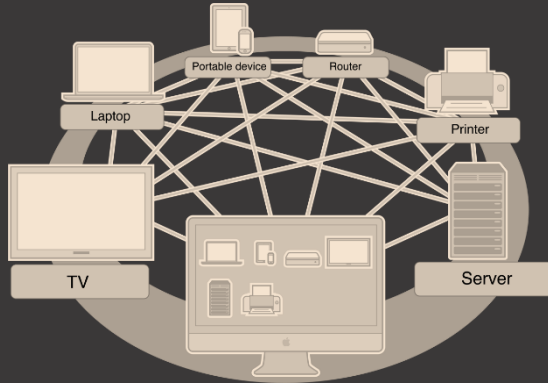
APPLE BONJOUR

Bonjour é uma implementação
da Apple de Zero-
configuration networking
Substituiu o Rendez-vous



<https://developer.apple.com/bonjour/>

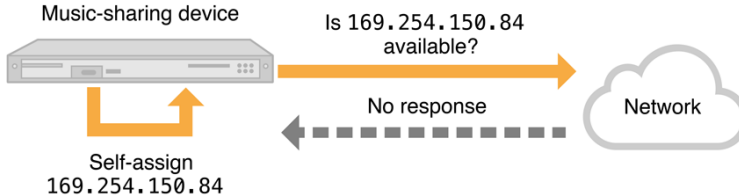
APPLE BONJOUR



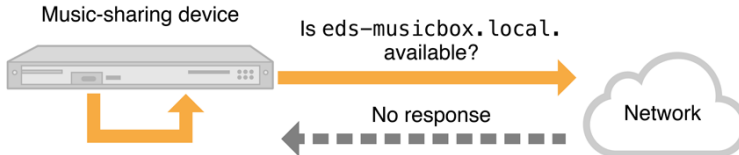
Bonjour localiza dispositivos tais como impressoras, outros computadores e os serviços que estes dispositivos ofereçam em uma rede local usando uma multicast Domain Name System (mDNS).

APPLE BONJOUR - CONSULTA

1. Address selection

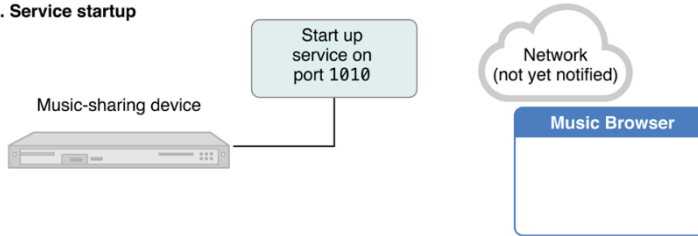


2. Name selection

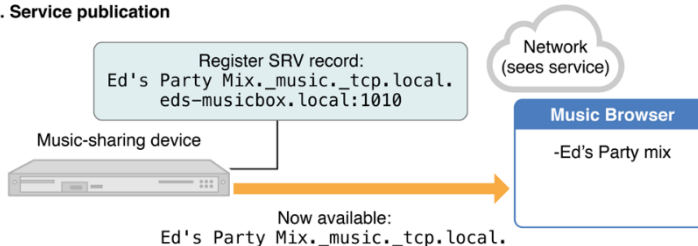


APPLE BONJOUR - CONSULTA

3. Service startup

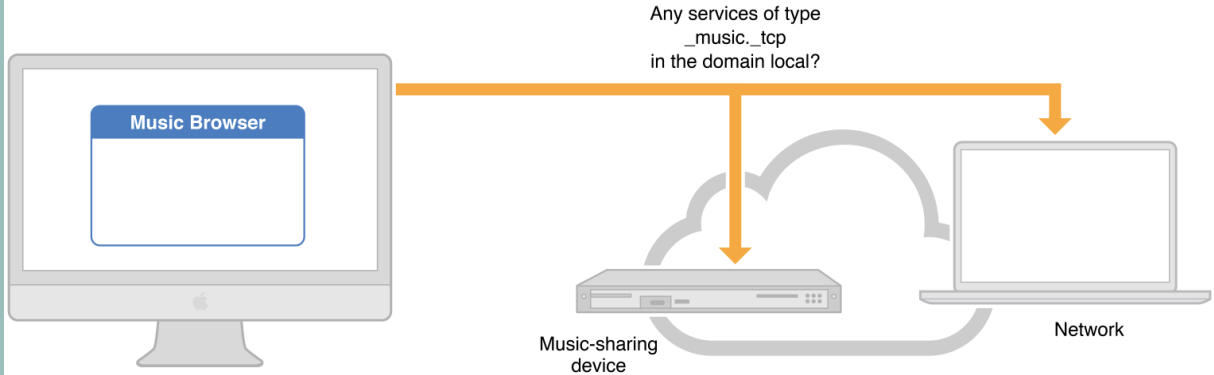


4. Service publication



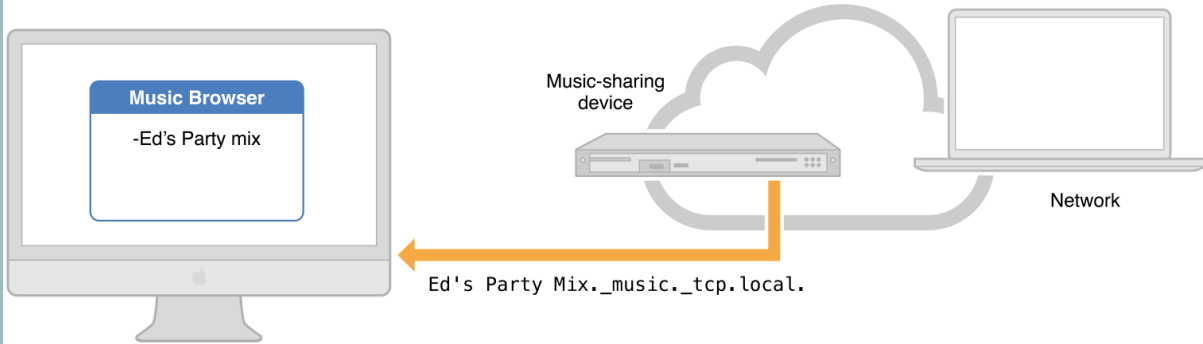
APPLE BONJOUR - DESCOBERTA

1. Query by service type



APPLE BONJOUR - DESCOBERTA

2. Response



DESCOBERTA DE SERVIÇO EM ANDROID

- O Network Service Discovery (NSD) fornece ao seu aplicativo acesso a serviços que outros dispositivos fornecem em uma rede local
 - Os dispositivos que suportam o NSD incluem impressoras, webcams, servidores HTTPS e outros dispositivos móveis.
- DNS-based Service Discovery (DNS-SD)
 - Aplicação define seu nome de descoberta e porta associada
 - https://www.youtube.com/watch?v=oi_ARV_l8Dc

REGISTRO

```
public void registerService(int port) {  
    NsdServiceInfo serviceInfo = new NsdServiceInfo();  
    serviceInfo.setServiceName("NsdChat");  
    serviceInfo.setServiceType("_http._tcp.");  
    serviceInfo.setPort(port);  
  
    mNsdManager = Context.getSystemService(Context.NSD_SERVICE);  
  
    mNsdManager.registerService(  
        serviceInfo, NsdManager.PROTOCOL_DNS_SD,  
        mRegistrationListener);  
}
```

IANA Name - Port

<http://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xml>

DESCOBERTA

```
@Override
    public void onServiceFound(NsdServiceInfo service) {
        // A service was found! Do something with it.
        Log.d(TAG, "Service discovery success" + service);
        if (!service.getServiceType().equals(SERVICE_TYPE)) {
            // Service type is the string containing the protocol and
            // transport layer for this service.
            Log.d(TAG, "Unknown Service Type: " +
service.getServiceType());
        } else if (service.getServiceName().equals(mServiceName)) {
            // The name of the service tells the user what they'd be
            // connecting to. It could be "Bob's Chat App".
            Log.d(TAG, "Same machine: " + mServiceName);
        } else if (service.getServiceName().contains("NsdChat")){
            mNsdManager.resolveService(service, mResolveListener);
        }
    }
```

Thanks!

windson@virtual.ufc.br