NIC ARGENTINA

Desmitificando DNS

DNS no es *tan* difícil

(aunque tampoco es *tan* fácil)

Dirección Nacional del Registro de Dominios de Internet
Secretaría Legal y Técnica

Mariano Absatz
NIC Argentina
Noviembre 2016

¿Nombres o números?

- La gente utiliza **nombres** para referirse a equipos y servicios en internet (www.nic.ar, correo.yahoo.com, ftp.cabase.org.ar)

- Las conexiones para transmitir los paquetes entre equipos requieren **direcciones (números) IP** (200.108.145.10, 2801:140:5::10, 72.30.203.4, 2001:4998:58:2c03::3005,66:192.185.4)

- Se necesita *algo* para transformar (mapear, traducir) los nombres (amigables para con los humanos) en números (necesarios para las máquinas)

DNS: De nombres a números

- Al principio se utilizaba un archivo (HOSTS.TXT) mantenido por el *Stanford Research Institute* (SRI) que se distribuía entre todos los hosts conectados a la red. A principios de los 1980's, esto era inmanejable, tanto por el tamaño del archivo, cuanto (y especialmente) por la agregación de información de diversas entidades

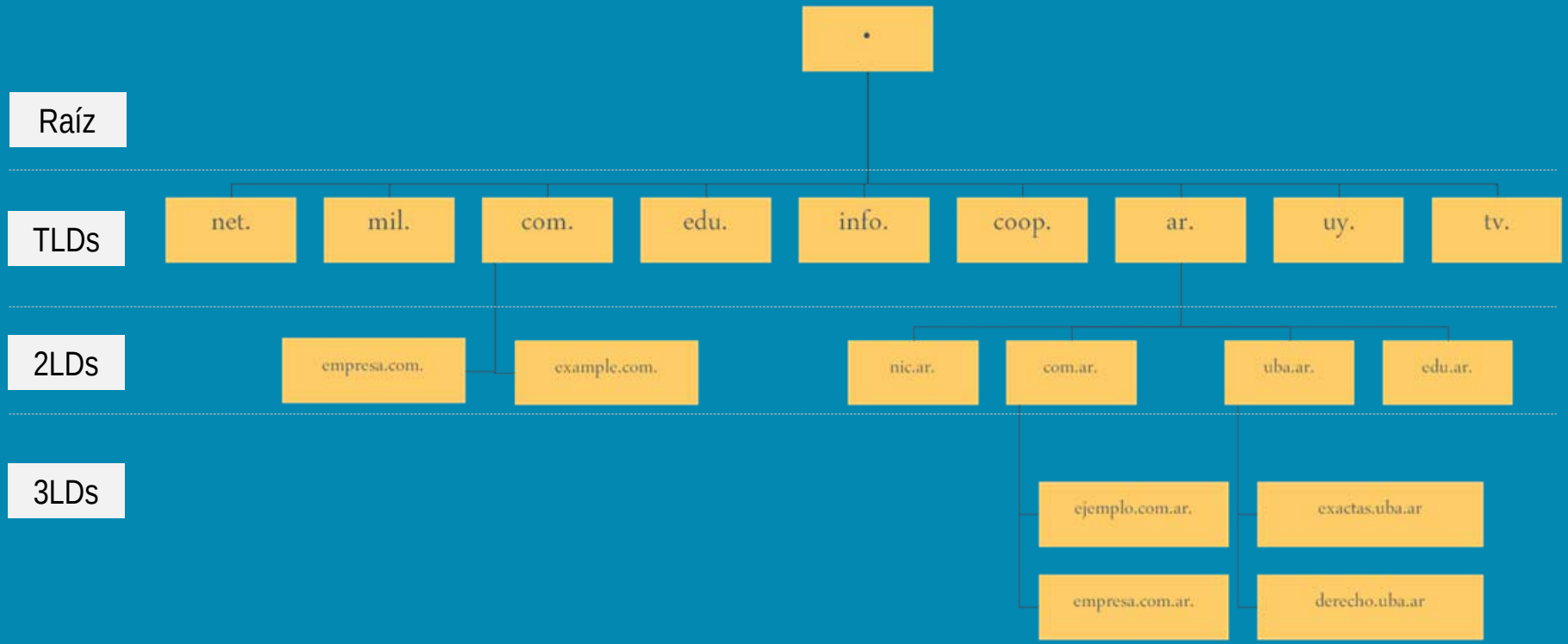
- En 1983 se crea el Sistema de Nombres de Dominios – *Domain Name System* (DNS)

- En 1984 se escribe la primera implementación de servidor de nombres: BIND (*Berkeley Internet Name Daemon*)

- En 1987 se actualiza la especificación del protocolo DNS

- A fines de los 1980's/principios de los 1990's DNS reemplaza completamente el viejo archivo HOSTS.TXT

Estructura global del DNS



DNS: 1 protocolo / 2 servicios / 3 roles

Servicio / rol:
Repositorio público de nombres

- Publicación de datos
- Autoritativo
- “Dueño” de la información
- *Authoritative name server*

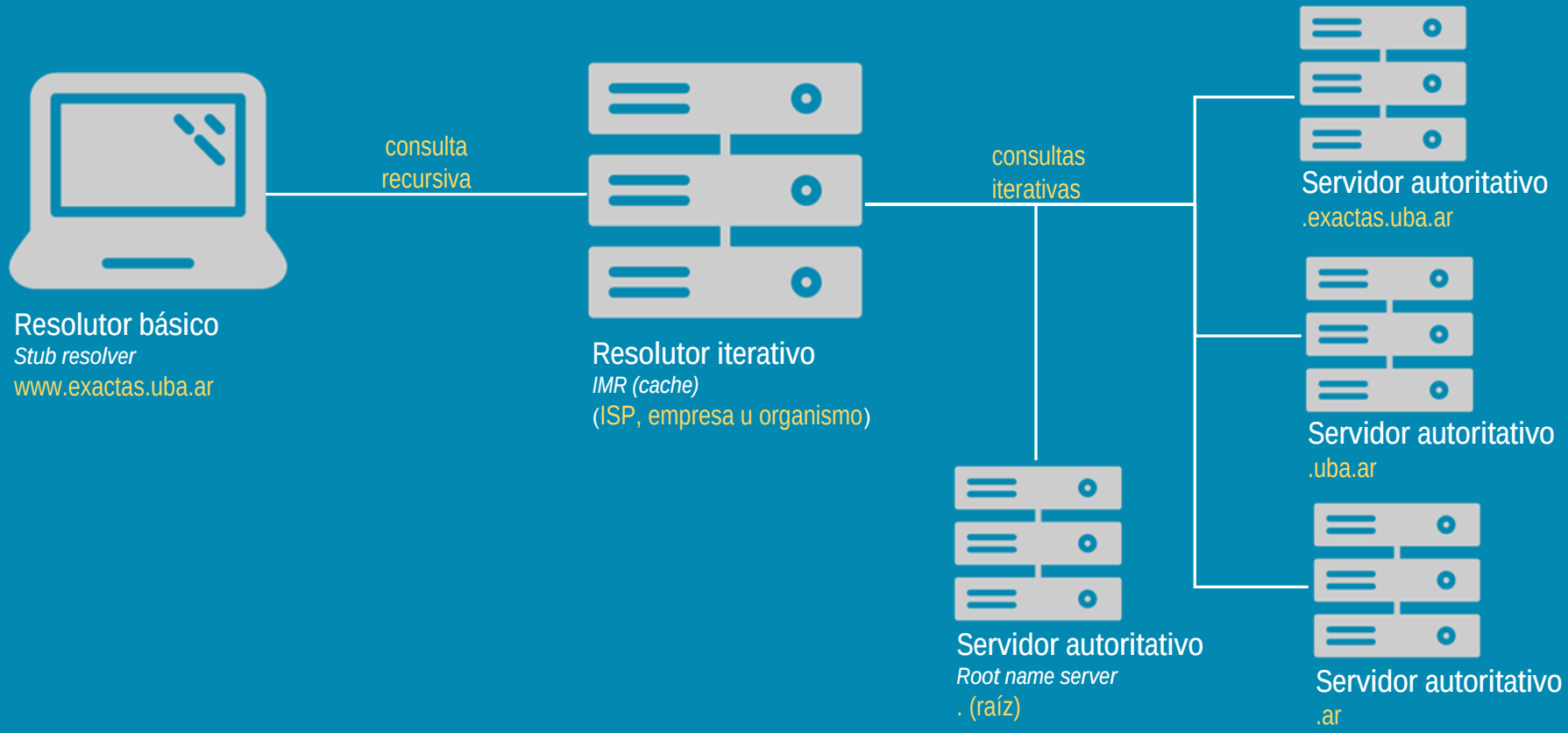
Servicio / rol:
Resolutor iterativo de nombres

- Consultas iterativas (desde la raíz) a servidores autoritativos
- *Cache* (opcional)
- *Iterative mode resolver (IMR)*

Rol:
Resolutor básico de nombres

- Consultas recursivas a resolutores iterativos preconfigurados
- Biblioteca local en el dispositivo
- *Stub resolver*

Mecanismo de resolución de nombres



Servidor de nombres autoritativo

- Funciones
  - Mapeo “nombre” ⟶ “valor” (sólo para la zona) (autoritativa)
  - Nombre inexistente (sólo para la zona) (autoritativa)
  - Referencias a subzonas (no autoritativa)
  - Error (o no contestar) si se consultan nombres fuera de la zona

- Base de datos: archivo de zona (zone file)
  - Información autoritativa (para la zona)
  - Información de delegación (no autoritativa, para subzonas delegadas)

- Clientes
  - Toda la internet

Resolutor iterativo (IMR)

- Funciones
  - Mapeo “nombre” ⟶ “valor” (para cualquier zona)
    - Consultará **iterativamente** a los servidores de nombres autoritativos necesarios hasta obtener la respuesta solicitada o decidir que el nombre no existe
  - Nombre inexistente
    - Si al hacer la consulta iterativa en algún momento recibe una respuesta de nombre inexistente
  - Error (o no contestar)
    - Si el ciente no está autorizado a consultarlo

Resolutor iterativo (IMR) (cont.)

- Base de datos
  - Archivo de "*hints*" con las direcciones IP de los servidores autoritativos de la raíz (*root servers*)
  - En algunos casos: información de reenvío (para consultar ciertos dominios a servidores específicos en lugar de los que se obtengan iterativamente desde la raíz)
  - *Cache* dinámico donde guarda la información obtenida durante las consultas (y la borra a medida que "se vence" o que no tiene más espacio)

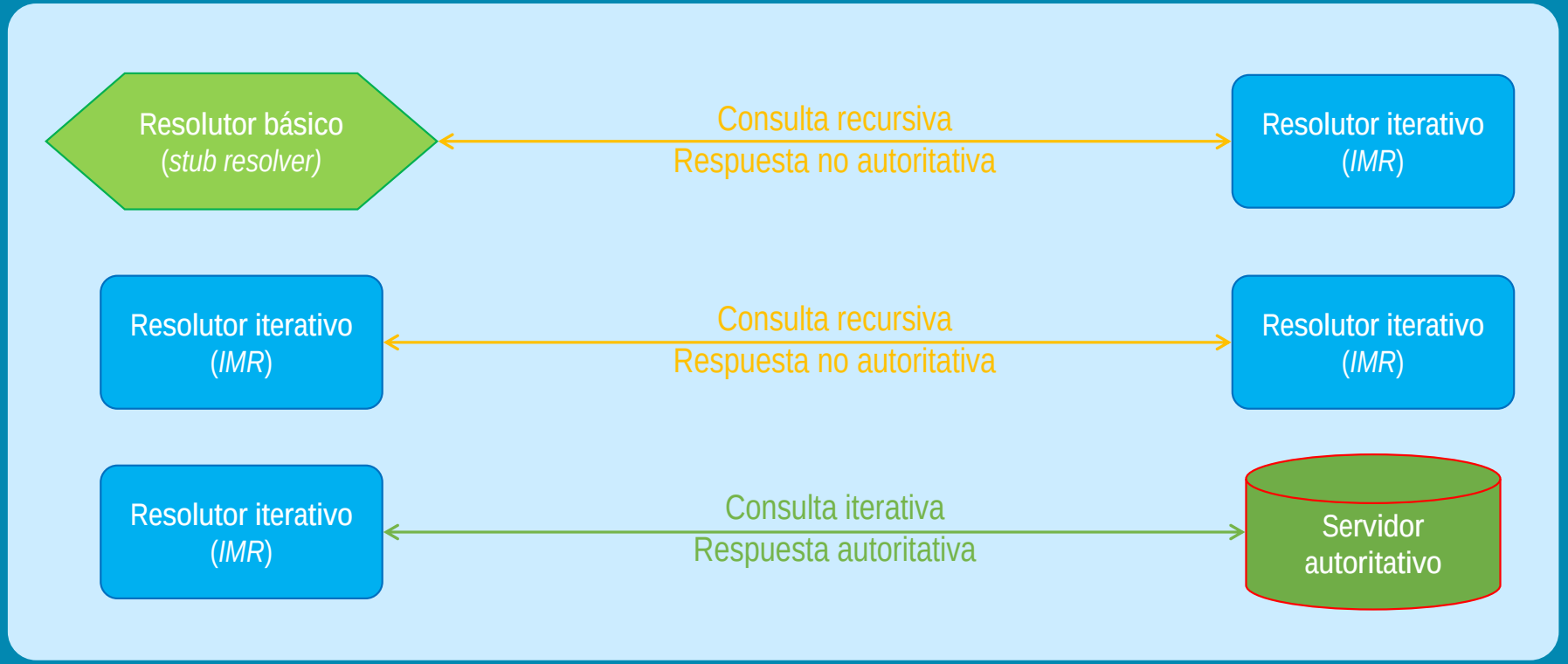
- Clientes
  - Normalmente, una organización/empresa o clientes de un servicio (por ejemplo: clientes de un ISP)
  - Existen servidores públicos que dan servicio a toda la internet (por ejemplo: Google – 8.8.8.8 / 8.8.4.4)
    - Consejo: No configurar un resolutor iterativo abierto a toda la internet (no todos somos Google)
    - Consejo: Es mejor configurar un resolutor propio que usar uno abierto ajeno

Resolutor básico (*Stub resolver*)

- Es una biblioteca de software que utilizan los programas que requieren resolución de nombres
  - gethostbyname(), gethostbyaddr(); getnameinfo(), getaddrinfo(); etc.
  - En algunos casos, también incluye un servicio local que además de resolver mediante consultas recursivas, mantiene un *cache* local
- Funciones
  - Mapeo "nombre" ⟶ "valor" (para cualquier zona)
    - Consultará **recursivamente** a los resolutores iterativos (IMR) configurados (ya sea manualmente o recibidos vía DHCP)
  - Nombre inexistente
    - Si al hacer la consulta recursiva recibe una respuesta de nombre inexistente
  - Error
    - Si no puede comunicarse con ningún servidor recursivo o si este le devuelve un error
- Base de datos
  - Archivo con las direcciones IP de los resolutores iterativos (IMR)
  - En algunos casos: *Cache* dinámico donde guarda la información obtenida durante las consultas (y la borra a medida que "se vence", cuando no tiene más espacio o cuando se reinicia el servicio)
- Clientes
  - Los programas locales que requieren resolver nombres de dominio

Transacciones DNS:

Consulta / respuesta (*query / response*)



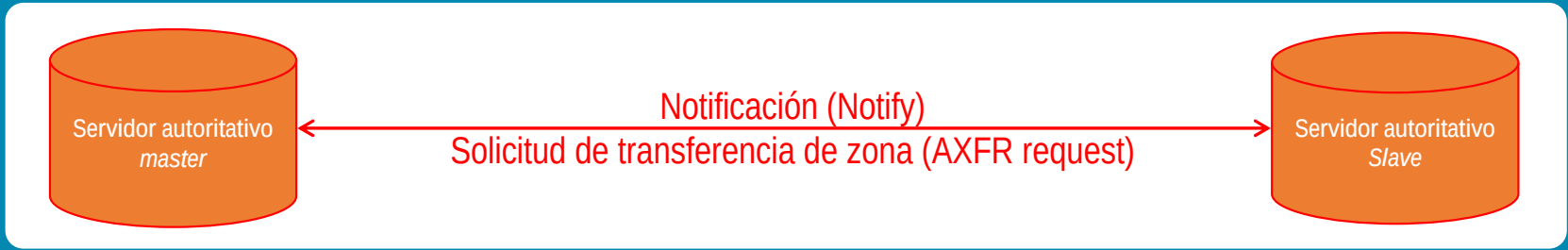
Transacciones DNS:

Consulta / respuesta (*cont.*)

- Consulta
  - Normalmente vía UDP (paquetes sueltos)
  - Más ágil y eficiente
  - Si la respuesta no cabe en un paquete, el servidor devuelve un error para que el cliente reintente vía TCP
- Respuesta
  - Conjunto de registros (RRset = Resource Record set)
  - Error
- Temas de seguridad
  - En UDP es fácil falsear el origen (ataques DDoS)
  - Validez de la información en duda
  - La sección “adicional” podría contener información no relacionada con la consulta (con el objetivo de que un IMR la “aprenda”)

Transacciones DNS:

Notificaciones / transferencias de zona



- Notificaciones (*Notify*): Desde el *master* a todos los *slaves*
  - Vía UDP
  - Temas de seguridad
    - Posible (y fácil) falsear el origen
    - Denegación de servicio (pero como antes del AXFR el *slave* verifica el número de serie con una consulta simple, no es grave)
- Solicitud de Transferencia de zona (AXFR): Iniciada desde un *slave* al *master*
  - Vía TCP, cuando recibe *notify* o cuando debe refrescar la información en base a datos del SOA
  - Solicita **todos** los RR de una zona
  - Temas de seguridad
    - Divulgación de información
    - Consumo de ancho de banda

Registros de Recursos más comunes

- Los objetos de una zona se llaman Registros de Recursos (*Resource Records*) o RR, los más comunes son:
- SOA: *Start Of Authority* marca el punto a partir del cual un servidor tiene autoridad (limitado por las propias delegaciones de subzonas)
- NS: *Name Server* indica los nombres de los servidores de nombres autoritativos para la zona
- A / AAAA: *Address* asocia una dirección IPv4 / IPv6 a un nombre de dominio
- CNAME: *Canonical NAME* apunta un nombre de dominio a otro nombre de dominio (una especie de “alias”)
- MX: *Mail eXchanger* asocia el nombre de un servidor de correo que brinda servicio a un dominio
- SRV: *SeRVice* asocia un servicio genérico a un nombre de dominio
- TXT: *TeXT* permite asociar una cadena de texto arbitraria a un nombre de dominio
- LOC: *LOCation* asocia una ubicación georreferencial a un nombre de dominio
- PTR: *PoinTeR* se utiliza para armar un mapeo “inverso” (obtener un nombre a partir de una dirección IP) en zonas especialmente designadas (in-addr.arpa / ip6.arpa)

Administrando servidores DNS

- El resolutor no debe*ría* nunca publicar información (actuar como autoritativo)
- El servidor autoritativo no debe*ría* nunca resolver consultas recursivas
- Configurar con mucho cuidado los nombres que sólo deben verse *localmente*

Servidores DNS

SERVIDOR AUTORITATIVO

- BIND
- *Microsoft Windows DNS Server*
- NSD
- *Nominum ANS (Authoritative Name Server)*
- PowerDNS
- *Secure64 DNS Authority*
- Knot DNS

RESOLUTOR CON *CACHE*

- BIND
- *Microsoft Windows DNS Server*
- Unbound
- *Nominum Vantio*
- PowerDNS Recursor
- *Secure64 DNS Cache*
- Knot Resolver

(Los servidores en *itálica* son software propietario y pago)

To BIND or not to BIND

PROS

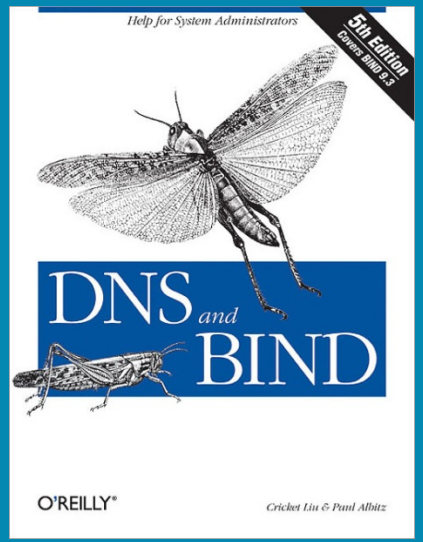
- Se puede usar como resolutor y como autoritativo
- Hay muchísima documentación
- Tiene mucha funcionalidad
- Es la implementación de referencia del protocolo DNS

CONTRAS

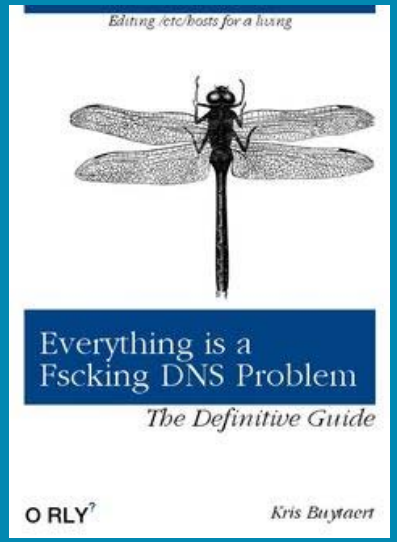
- Se puede usar como resolutor y como autoritativo
- El mismo *daemon* realiza ambas tareas
- Es *muy* grande
- La configuración es muy compleja
- Tiene una historia de vulnerabilidades

Brevísima bibliografía sobre DNS

- DNS and BIND, *Cricket Liu & Paul Albitz*, O'Reilly Media
- La *Biblia* de DNS y BIND
- 1ª edición: 1992 (400+ páginas)
- 5ª edición (última): 2006 (600+ páginas)



- Un libro inexistente (pero real)



Instalá un resolutor en 20 minutos

- Unbound
  - Pequeño
  - Eficiente
  - Fácil de instalar
    - Paquetes compilados en la mayoría de las distribuciones GNU/Linux y *BSD
    - Fácil de compilar
  - Fácil de configurar

Unbound en 20 minutos (0)

- Instalamos el software

  `sudo apt-get install unbound`

  – En ubuntu esto hace lo siguiente:

    - Instala el unbound (y el unbound-anchor)
    - Crea el usuario y grupo unbound para no correr como root
    - Configura el control remoto (unbound-control) para que lo pueda usar root
    - Configura unbound para que cuando el equipo *bootea* actualice el punto de confianza raíz de DNSSEC usando unbound-anchor y arranque dentro de una jaula `chroot`
    - Deja una configuración *pelada* que sólo escucha en localhost en **`/etc/unbound/unbound.conf`**
    - Deja un archivo de configuración más completo bien comentado en **`/usr/share/doc/unbound/examples/unbound.conf`**

- Hacemos la primera consulta

  `dig @::1 nic.ar aaaa +short`

  Resultado → `2801::140:5::10`

Unbound en 20 minutos (1)

- Agregamos la configuración completa a la que viene preinstalada

```
sudo  sh -c "cat /usr/share/doc/unbound/examples/unbound.conf >> /etc/unbound/unbound.conf”
```

- Verificamos que la nueva configuración sea válida

```
sudo unbound-checkconf
```

- Cargamos la nueva configuración

```
sudo unbound-control reload
```

- Hacemos la segunda consulta

```
dig @::1 cabase.org.ar a +short
```

Resultado → `200.69.5.212`

Unbound en 20 minutos (2)

- El archivo de configuración tiene un formato simple (YAML)

```
sección1:
     atributo1: valor1
     atributo2: valor2
sección2:
     atributo3: valor3
     atributo4: valor4
```

- La mayor parte de la configuración va en la sección server:

Unbound en 20 minutos (3)

Mínimos cambios para “salir andando”

- La configuración por defecto sólo escucha en localhost. Lo cambiamos para que escuche en todas las interfaces (podríamos elegir sólo algunas). Es necesario reiniciar el unbound (no alcanza con recargarlo)

```
server:
     interface: 0.0.0.0
     interface: ::0
```

- La configuración por defecto sólo atiende a localhost. Lo cambiamos para que atienda a nuestras redes

```
server:
     access-control: 10.0.0.0/8
     access-control: fe80::/10
```

- Verificamos que la nueva configuración es válida

```
sudo unbound-checkconfig
```

- Reiniciamos unbound (recordar que no alcanza con hacer un reload porque cambiaron las interfaces en las que escucha)

```
sudo unbound-control stop
sudo unbound-control start
```

Unbound en 20 minutos (4)

Más cambios para “ser prolijos”

- Unbound tiene una lista de *hints* (la lista de direcciones IP de los servidores raíz) dentro del código
- Obviamente, esta lista puede quedar desactualizada por lo que es recomendable bajarla del sitio oficial de IANA www.iana.org/domains/root/files cuando se instala el resolutor y luego periódicamente (digamos, cada 4 a 6 meses)

```
sudo wget https://www.internic.net/domain/named.root -O /etc/unbound/root.hints
```

- Lo configuramos en `/etc/unbound/unbound.conf`

```
server:
    root-hints: /etc/unbound/root.hints
```

- Verificamos que la nueva configuración es válida y la cargamos

```
sudo unbound-checkconfig && sudo unbound-control reload
```

Unbound: más información

- El archivo de configuración tiene profusión de comentarios, también: `man unbound.conf`
- La página oficial es unbound.net
- La documentación está en unbound.net/documentation/index.html
- Instalación y configuración básica: unbound.net/documentation/howto_setup.html
- Optimizaciones: unbound.net/documentation/howto_optimise.html
- Algunos consejos:
  - Habilitar el *logging*: sirve para encontrar problemas en servidores autoritativos y en clientes (los discos SSD no son muy caros ahora)
  - No usar balanceadores de carga para el resolutor (“un *cache* muy ocupado es un *cache* feliz”)
  - Para entornos de *mucha* carga, mirar dnsdist dnsdist.org

Una pequeña disgresión:

Criptografía en 10 minutos

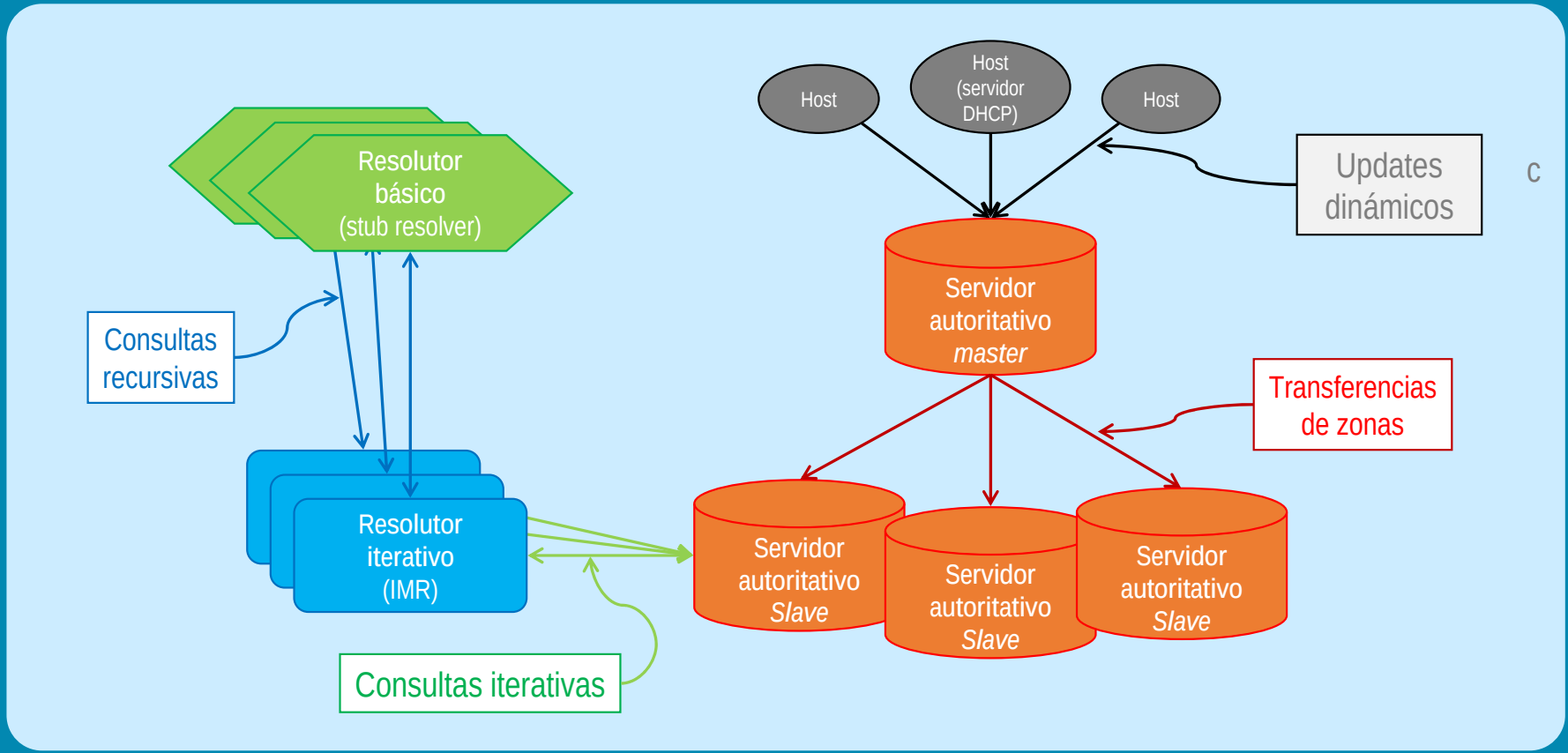
- Encripción simétrica
  - La misma clave que encripta es la que desencripta
  - *Ambas partes tienen que conocer la misma clave*
    - El problema es como hacerlo enforma secreta
  - Texto_plano ✖ clave = Texto_oculto
  - Texto_oculto ✖ clave = Texto_plano
  - Cuanto más larga la clave, más difícil de atacar por fuerza bruta (pero es más “caro” encriptar y desencriptar)
  - La longitud ideal de la clave depende del algoritmo (y la capacidad computacional de la época)

Una pequeña disgresión:

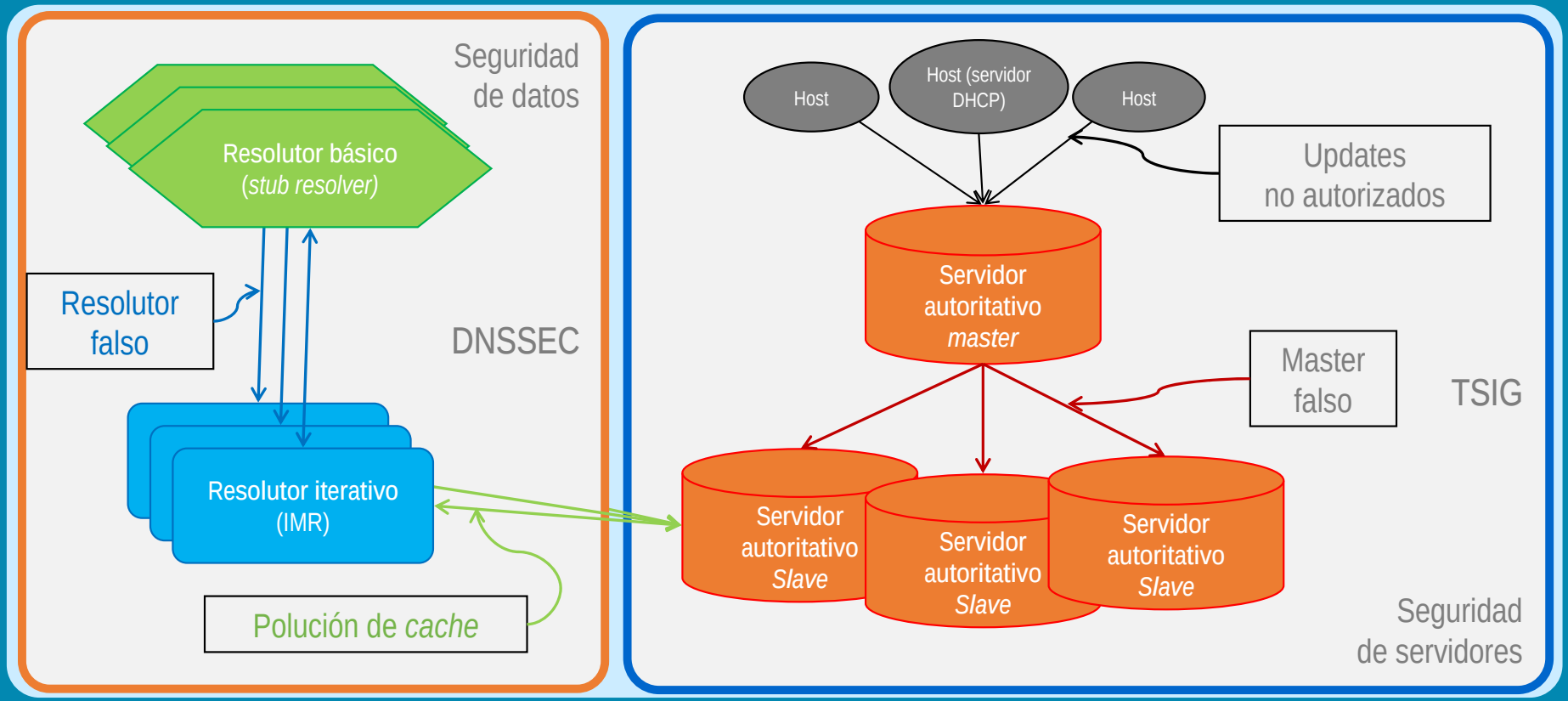
Criptografía en 10 minutos (cont.)

- Encripción asimétrica
  - Se utilizan pares de claves asociadas:
    - 1 clave “pública” / 1 clave “privada”
  - La clave privada sólo la conoce el que la genera
  - *La clave pública la puede conocer <u>todo el mundo</u>*
  - Texto_plano ✖ clave_priv = Texto_oculto
  - Texto_oculto ✖ clave_pub = Texto_plano
  - Sirve para “firmar” información:
    - Autenticación de origen
    - No repudiable
    - Sólo quien tiene la clave privada podría haber generado la información encriptada que se desencripta con la clave pública
  - Es más “cara” computacionalmente que la encripción simétrica

Transacciones DNS



Vulnerabilidades en transacciones DNS



Seguridad para DNS: TSIG / DNSSEC

- Seguridad de servidores
  - AXFR / Updates dinámicos: TSIG (claves compartidas – criptografía simétrica)
  - Notificaciones: Dirección IP de origen (bajo riesgo – opcionalmente TSIG)

- Seguridad de datos
  - Consultas / Respuestas: DNSSEC (firma digital de las respuestas – criptografía asimétrica – cadena de confianza)

TSIG: Utilización

- Se utiliza para validar las transferencias de zona (solicitadas por el *slave* y enviadas por el *master*)
  - Normalmente, el *master* y los *slaves* de una zona o bien están bajo la misma autoridad administrativa, o bien existe un convenio entre los administradores de ambos servidores
  - Las claves privadas se transmiten por mecanismos seguros ajenos al DNS (*out of band*)
  - Se debe generar una clave compartida para **cada** par de servidores *master*/*slave* (es decir, si hay 1 *master* y 4 *slaves* se utilizaran 4 claves distintas)

- Se utiliza cuando hay actualizaciones dinámicas de registros en la zona (*dns dynamic updates*)
  - En general esto ocurre entre un servidor DHCP y un servidor DNS *master*, también bajo una misma autoridad administrativa

- Si bien normalmente no vale la pena, se puede utilizar para autenticar los mensajes de notificación (*notify*) desde el *master* a los *slaves* (también con claves distintas para cada par de servidores, aunque puede utilizarse la misma que se utiliza para transferencia de zona entre esos servidores)

DNSSEC: DNS SECurity Extensions

- Objetivo: Validar la información que un cliente recibe de los servidores DNS
- Nuevos tipos de RRs:
  - RRSIG: *Resource Record set SIGnature* contiene la firma digital de un RRset
  - DNSKEY: *DNS KEY* contiene una clave pública (e información asociada)
  - NSEC (NSEC3): *Next SECure* contiene información de todos los tipos de RR de un nombre y un puntero al siguiente nombre (permite validar la inexistencia de un RR)
  - DS: *Delegation Signer* contiene la clave pública de una subzona (zona “hija”), firmada con la clave privada de la zona que hizo la delegación (zona “madre”)

DNSSEC: Claves para una zona

- Si bien no es obligatorio, conviene utilizar dos tipos de (pares de) claves para utilizar en una zona

- ZSK: *Zone Signing Key* – se utiliza para firmar todos los RRs de la zona
  - La ZSK nunca sale de la zona y sólo se utiliza para firmar registros propios de la zona

- KSK: *Key Signing Key* – se utiliza para firmar solamente los DNSKEYs y sirve para establecer la confianza en la zona; es la que se envía a la zona "madre" para ser firmada y puesta en un RR de tipo DS
  - El RR DNSKEY que contiene la KSK tiene un flag SEP (*Secure Entry Point*) que es el que indica que la clave contenida en el RR es el punto de confianza de la zona

DNSSEC: Cadena de confianza

- Una cadena de confianza se establece desde un “ancla o punto de confianza” (*trust anchor*)
- El punto de confianza consiste en la KSK-pub de una zona determinada
- Esto se debe **configurar** en el servidor de resolución con validación:
  - Se configura la KSK-pub asociada al nombre de la zona a la que corresponde esa KSK
  - Es un *hint* que se configura y se mantiene por fuera del protocolo DNS, del mismo modo que la lista de direcciones IP de servidores autoritativos de *root*

DNSSEC: Cadena de confianza (cont.)

- Desde julio de 2010 la zona raíz "." (*root zone*) está firmada y su punto de confianza (*trust anchor*) KSK-2010 está publicado en https://www.iana.org/dnssec
- Quienes operen servidores de resolución de nombres de dominio con validación DNSSEC deben configurar este punto de confianza (del mismo modo que configuran las direcciones IP de los servidores raíz)
  - La clave de la zona raíz se valida por fuera del protocolo DNS (*out of band*), utilizando https y certificados SSL X.509
- Si el administrador de dominio hace firmar el KSK de su zona en un DS del registro que le delegó la zona, y a su vez este extiende su cadena de confianza hasta la raíz, la zona será validable por todos los servidores de resolución que tengan configurado el punto de confianza de la raíz

DNSSEC: Isla(s) de confianza

- En el caso de que la zona “madre” no esté firmada (o no permita delegar confianza a través de RRs DS), si se quiere publicar información utilizando DNSSEC, la única posibilidad es publicar la KSK-pub como punto de confianza y convencer a quienes operen servidores de resolución con validación de que configuren dicho punto de confianza

- Si una zona **no** tiene su KSK firmada por su zona “madre” en un RR DS, pero sí firma las KSK-pub de sus zonas “hijas” que configuraron DNSSEC, esta zona y dichas “hijas” constituyen una **isla de confianza**

- Hasta 2010, toda la operacion de DNSSEC estaba basada en islas de confianza

Traspaso de la KSK raíz

- En 2016/2017 se está realizando el primer traspaso (*rollover*) de la KSK de la zona raíz (*root*), de la KSK-2010 actualmente en uso a la nueva KSK-2017

- El proceso tiene 8 fases:

  A. **Generación** del nuevo par de claves KSK-2017 en 1ª instalación de gestión de claves (2016-10)

  B. **Replicación** del nuevo par de claves a la 2ª instalación de gestión de claves (2017-02)

  C. **Firma** con KSK-2017 de los primeros datos para usar en fase D (2017-05)

  D. **Publicación** de KSK-2017 firmada con KSK-2010 y KSK-2017 (2017-08)

  E. **Traspaso**: se firma la zona raíz sólo con KSK-2017 (2017-11)

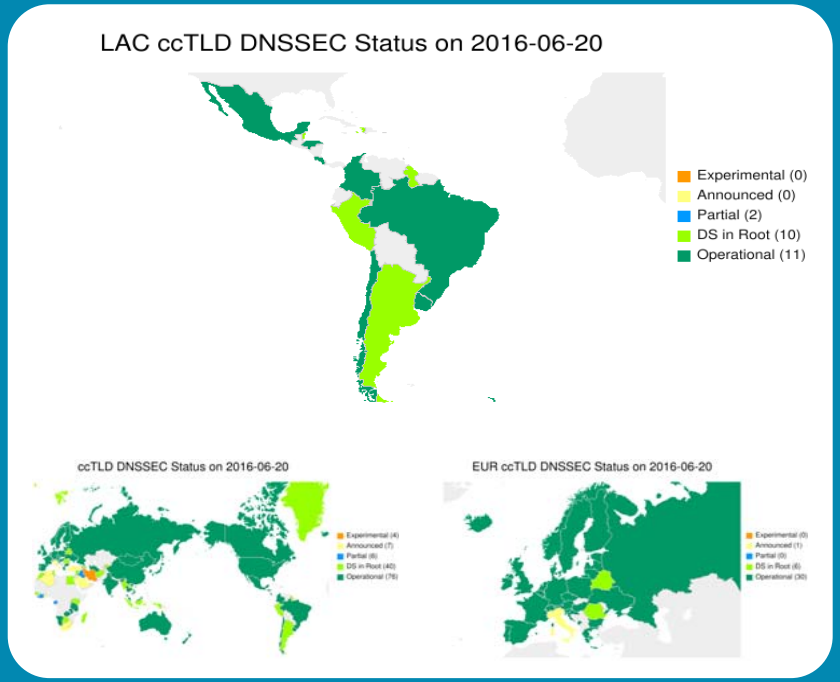
  F. **Revocación**: se quita KSK-2010 de la zona raíz (2018-02)

  G. **Borrado 1**: se borra KSK-2010 de la 1ª instalación de gestión de claves (2018-05)

  H. **Borrado 2**: se borra KSK-2010 de la 2ª instalación de gestión de claves (2018-08)

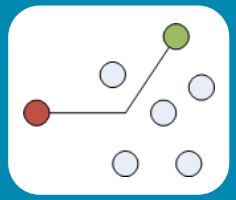
Despliegue de DNSSEC

- DNSSEC todavía está desplegándose lentamente
- Para saber si un dominio tiene DNSSEC hay que asegurarse de usar un *resolver* con validación
- Para “verlo” hay que agregar plugins en Firefox o Chrome (desarrollados por cz.nic)

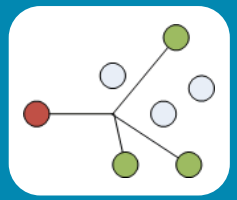


Anycast: Mejorando la confiabilidad y la *performance* de DNS

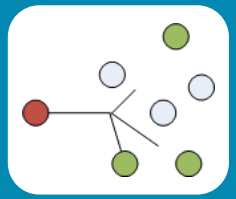
- Unicast: Comunicación uno a uno



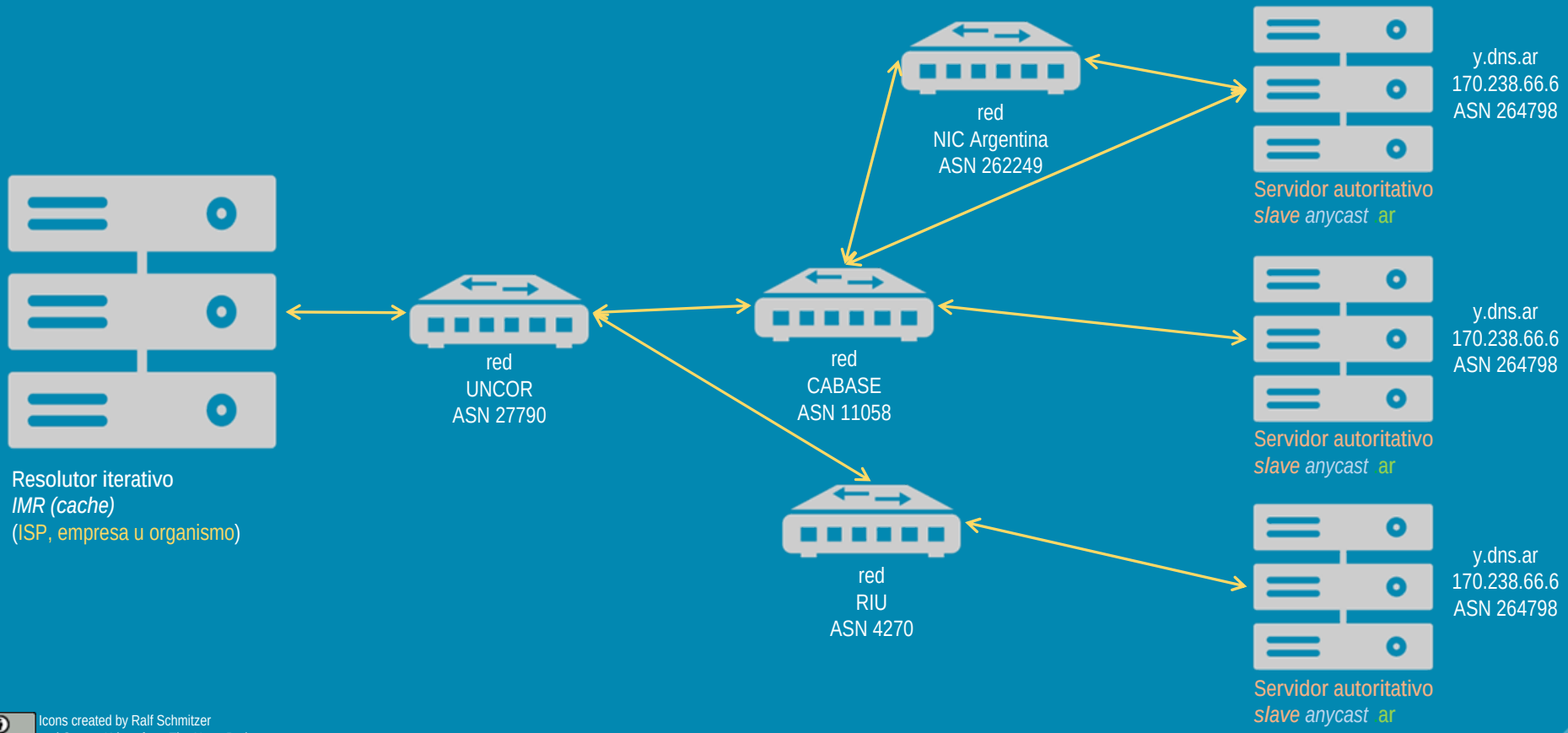
- Multicast: Comunicación uno a muchos



- Anycast: Comunicación uno a “uno (el más cercano) de muchos”



Red anycast DNS autoritativo .ar



¿Preguntas?

¡MUCHAS GRACIAS!