9. DNSの仕組みと管理

Contents

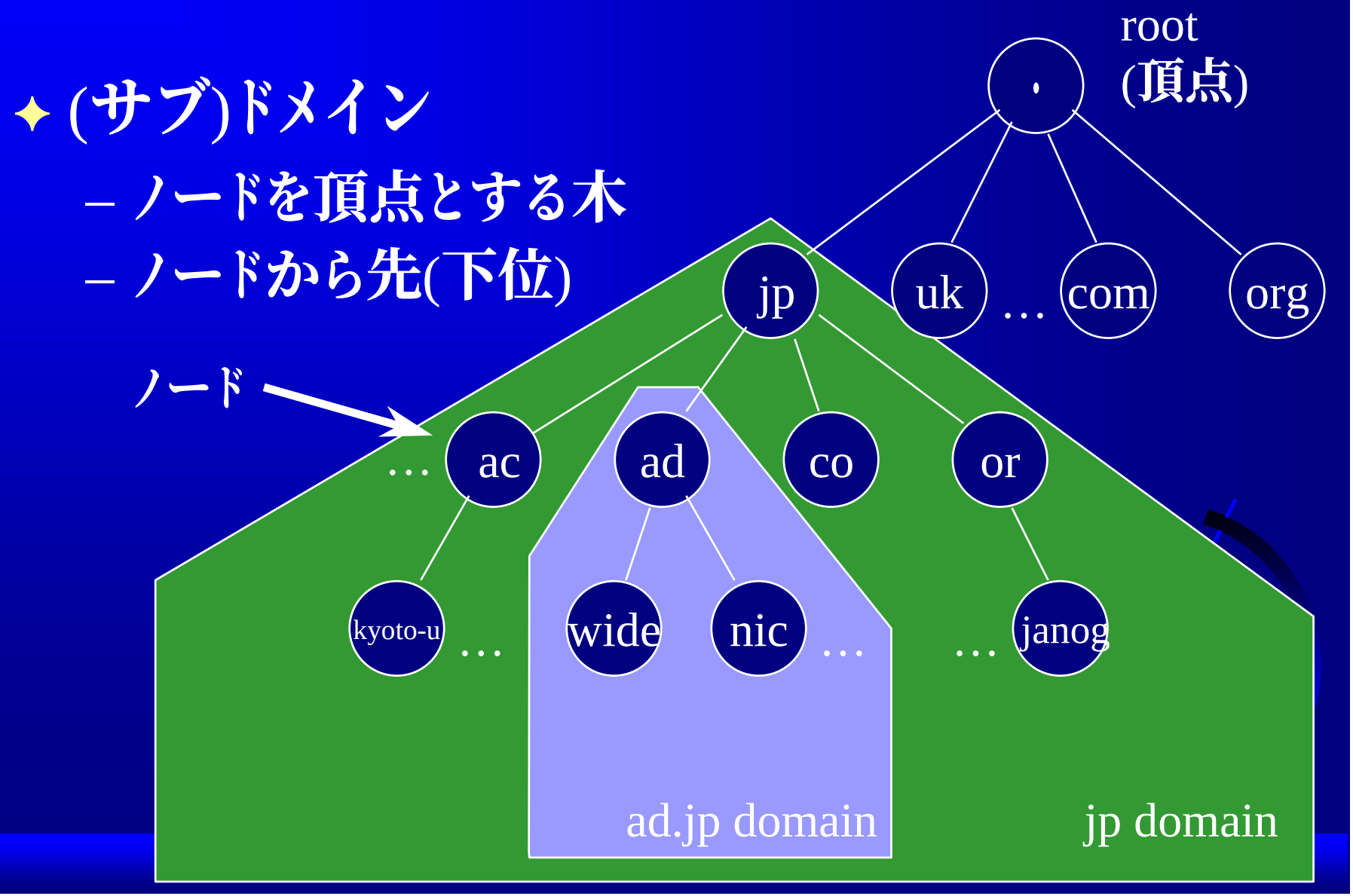
- ドメインとゾーン
- サーバの種類
- サーバの設定
- レコード詳説
- CIDRと逆引き
- IPv6
- アドレスの補完
- Wildcard MX
- エラーの一覧
- デバッグ用ツール
- DNSの今後

DNS (Domain Name System)

- 広域分散データベース
- ホスト名とIPアドレスの対応表
- 自律分散管理

ドメイン・ツリー

- (サブ)ドメイン
  - ノードを頂点とする木
  - ノードから先(下位)



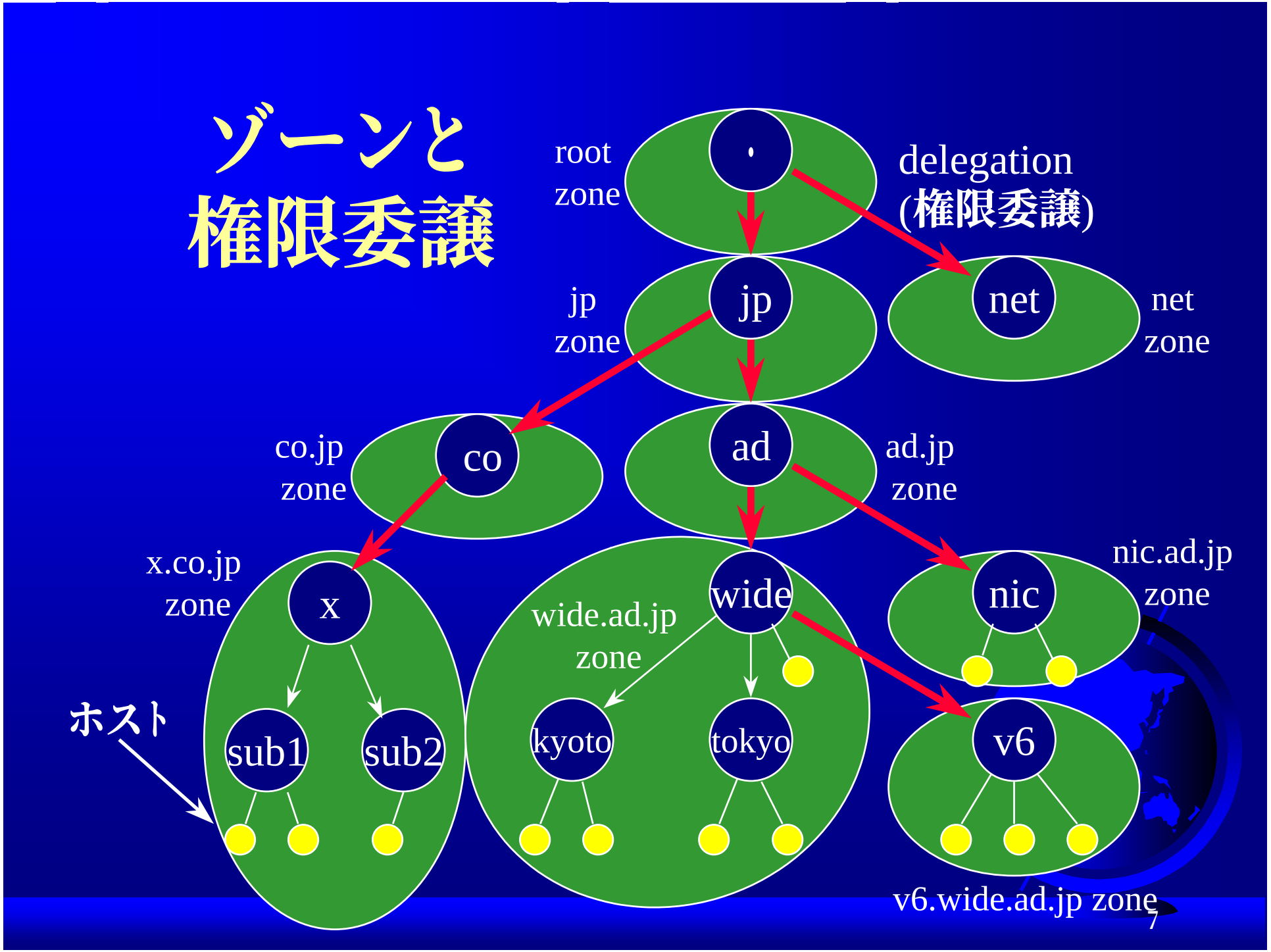
分散管理と検索

- 必要に応じてノード間の上下リンクで分割
  - ノードの下流へのリンク
    - Delegation(権限委譲)
  - TOP domain, 2nd(3rd)-level domain
    - NIC が管理
- 単方向リンク(上から下へ)
  - 上位へはrootまで戻ってから辿る
  - 全サーバはrootを知っている

ゾーンとドメイン

- 必ずしもノード単位で分割管理の必要なし
- ゾーン
  - 共同管理される隣接ノードの集合
  - 必ずしもドメインとは一致しない
    - 1ゾーンで複数ドメインを管理
- データの管理単位
  - 部所単位/地域単位に分割
  - 1つのネームサーバに対応
- 末端ではドメインと一致

ゾーンと権限委譲


root
zone
.
delegation
(権限委譲)
jp
zone
jp
net
net
zone
co.jp
zone
co
ad
ad.jp
zone
x.co.jp
zone
x
wide
wide.ad.jp
zone
nic
nic.ad.jp
zone
ホスト
sub1
sub2
kyoto
tokyo
v6
v6.wide.ad.jp zone

サーバの種類

- サービスの種類
  - データ提供用 (検索もする) / 検索専用
- データ(ゾーン)の管理
  - そこで編集 / 他からコピー
- 権限
  - Authorized / Unauthorized
- サービス対象
  - 組織外向け / 組織内向け

提供するデータ(ゾーン)の管理 (cont.)

- プライマリ(マスタ)・サーバ
  - データベース・ファイルの編集を行なう
- セカンダリ(スレーブ)・サーバ
  - プライマリ・サーバからデータをコピー
    - 別のセカンダリからでも可
      - コピーチェイン
    - コピー元サーバを複数指定可能
  - プライマリのサービス・バックアップ
  - 同時に到達不能にならない場所に配置

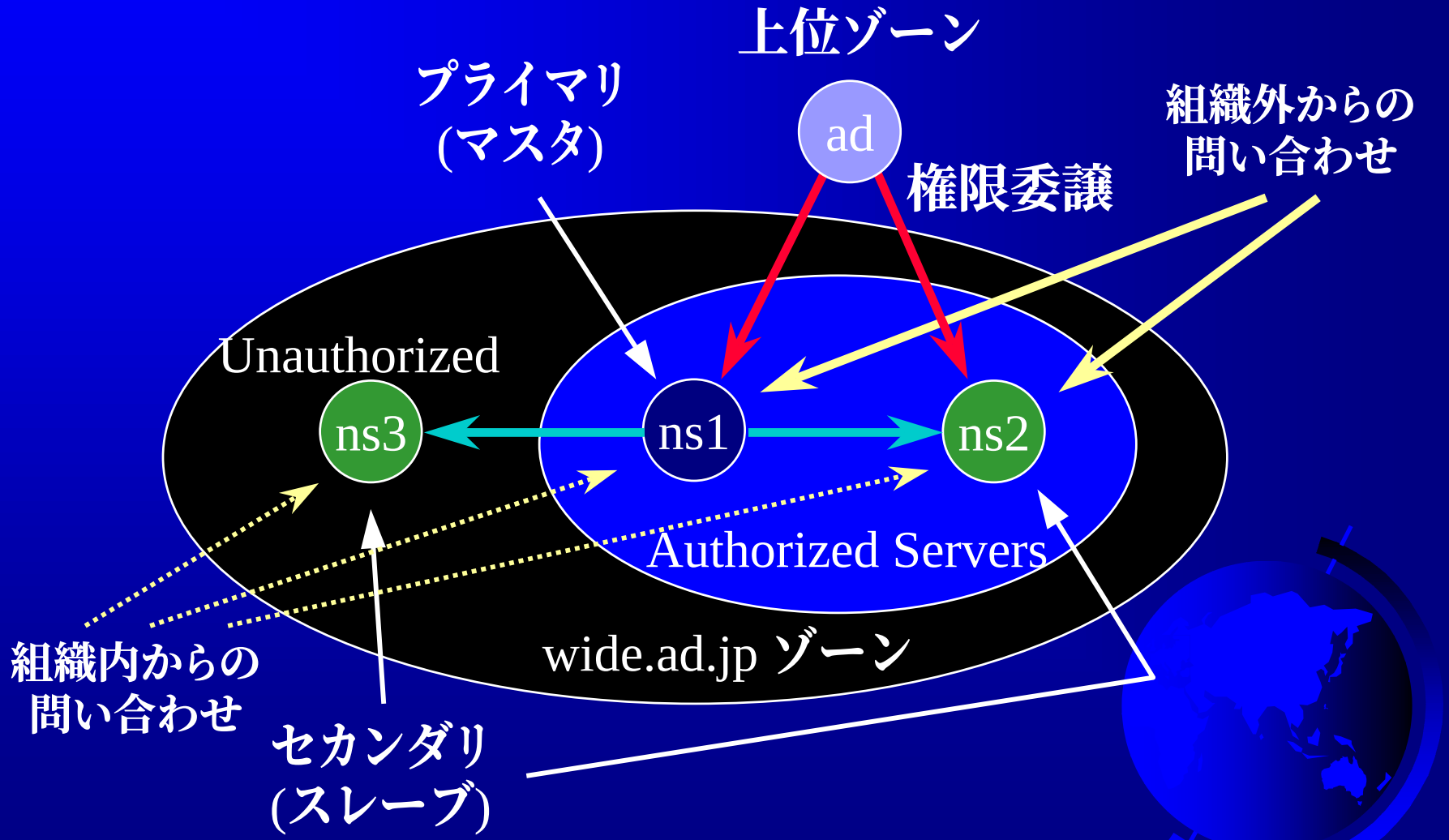
提供するデータ(ゾーン)の管理 (cont'd)

- 検索要求は平等に来る
  - プライマリ･セカンダリの区別はない
- ゾーンに対する区別
  - 一つのサーバで複数のゾーンを管理
    - ゾーンAに対してはプライマリ
    - ゾーンBに対してはセカンダリ
  - サーバ個体に対する区別ではない

データの提供に関する権限

- Authorized Server
  - データをインターネットに提供
  - 上位ゾーンからのリンク(権限委譲)がある
- Unauthorized Server
  - 手元の恒常的キャッシュ
  - データを近隣クライアントに提供
  - 上位ゾーンからのリンク(権限委譲)がない
- ゾーンに対する区別

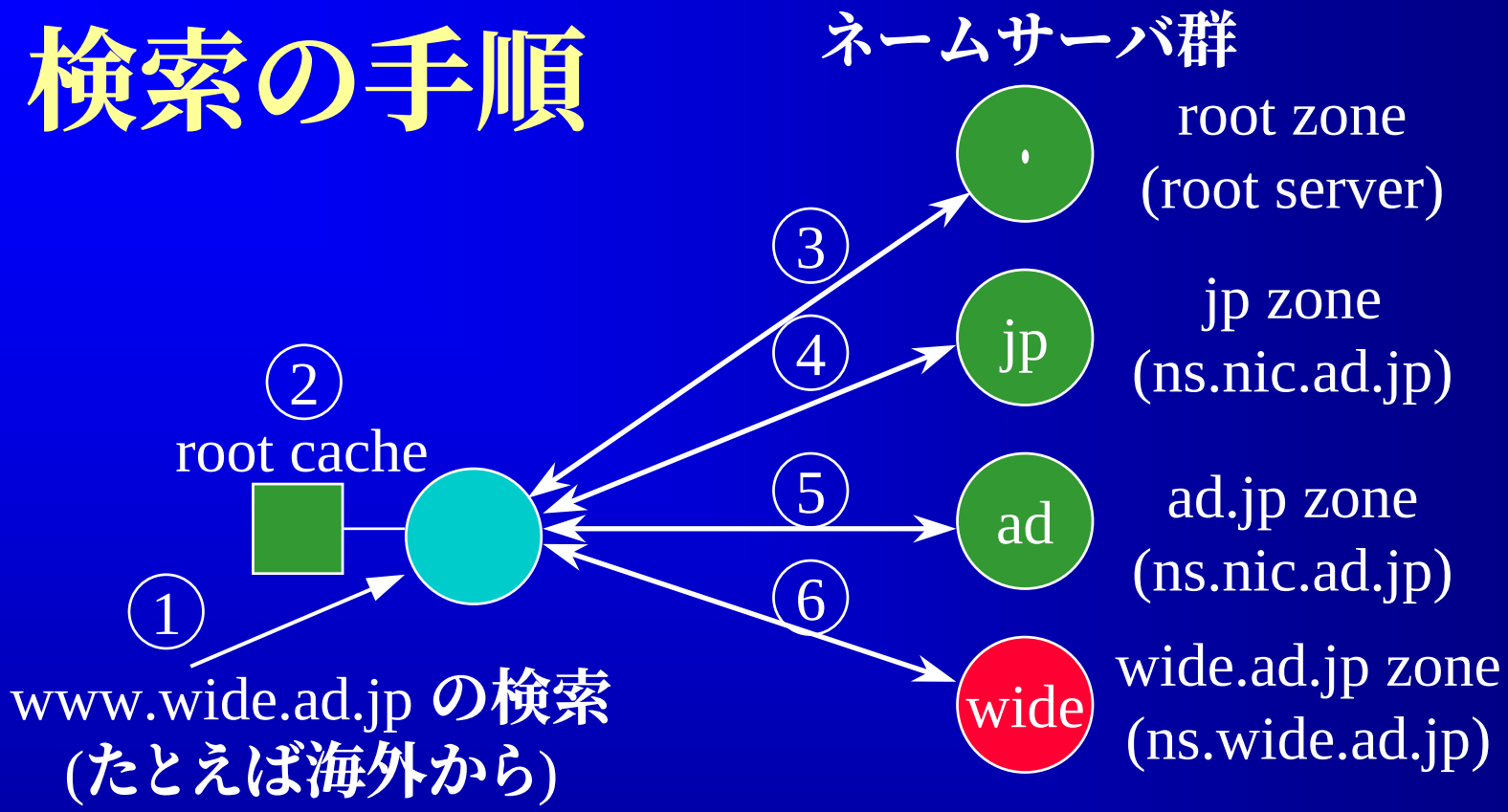
サーバの権限とゾーン


上位ゾーン
プライマリ
(マスタ)
ad
権限委譲
組織外からの
問い合わせ
Unauthorized
ns3
ns1
ns2
Authorized Servers
wide.ad.jp ゾーン
組織内からの
問い合わせ
セカンダリ
(スレーブ)

検索専用

- キャッシュサーバ
  - 一度検索したデータをしばらく記憶
    - Unauthoritative Answer として応答
  - プライマリでもセカンダリでもない
    - どのゾーンに対しても
- 参考：ネガティブ・キャッシュ
  - 該当レコードが存在しなかったことを保持 (全サーバ)

検索の手順



- root server への到達性がなければ引けない
  - 国際線の安定性問題
  - 国内に root server が必要
  - jp zone の Unauthorized Secondary に

DNS Servers

- Berkeley Internet Name Domain (BIND) Server
  - bind 4.9.6
  - bind 8.1.1
    - できるだけ最新版を
      - セキュリティ、パフォーマンス、信頼性、新機能
  - http://www.isc.org/bind.html
- Windows NT のネームサーバなど
  - 信頼性は大丈夫(?)

サーバの設定ファイル

- named.boot (bind 4)
- named.conf (bind 8)
  - named-bootconf.pl
    - named.boot からのフォーマット変換ツール
    - bind 8 に添付

BIND では ‘; ’ がコメントの開始

sample of named.boot (bind 4)

```
directory /etc/namedb
; 起動時に知っておくべきデータ (ルートサーバ情報)
cache      .                          root.cache
; localhost に関する情報
primary    localhost                  localhost.zone
primary    0.0.127.in-addr.arpa       localhost.rev
; プライマリとして提供するゾーン
primary    wide.ad.jp                 wide.zone
primary    136.178.203.in-addr.arpa   wide.rev
; セカンダリとして提供するゾーン
secondary v6.wide.ad.jp               203.178.136.188     sec/v6.zone
```

sample of named.conf (bind 8)

```
options {
        directory "/etc/namedb";
};

zone "." {
        type hint;
        file "root.cache";
};

zone "localhost" {
        type master;
        file "localhost.zone";
};
```

```
zone "0.0.127.in-addr.arpa" {
        type master;
        file "localhost.rev";
};

zone "wide.ad.jp" {
        type master;
        file "wide.zone";
};

zone "136.178.203.in-addr.arpa" {
        type master;
        file "wide.rev";
};
```

```
zone "v6.wide.ad.jp" {
        type slave;
        file "sec/v6.zone";
        masters {
                203.178.136.188;
        };
};
```

root cache

- ルートサーバに関する情報
  - ルートサーバさえ知れば全て検索可能
- ftp://ftp.rs.internic.net/domain/named.root
- 13番目が日本で稼働開始(1997/8)
  - m.root-servers.net
- Firewall の内側では
  - 内部向け root server を用意

sample of root.cache

```
; formerly NS.INTERNIC.NET
.                        3600000  IN  NS  A.ROOT-SERVERS.NET.
A.ROOT-SERVERS.NET.  3600000      A   198.41.0.4
;
; formerly NS1.ISI.EDU
.                        3600000      NS  B.ROOT-SERVERS.NET.
B.ROOT-SERVERS.NET.  3600000      A   128.9.0.107
:
:
; housed in Japan, operated by WIDE
.                        3600000      NS  M.ROOT-SERVERS.NET.
M.ROOT-SERVERS.NET. 3600000      A   202.12.27.33
```

forwarders

- 組織内から外部のアドレスの問い合わせ
  - 外部のネームサーバに問い合わせを転送
    - socks 対応 firewall などの場合
  - slave とともに指定

    forwarders 12.34.56.79 (内外両側からアクセス可能なサーバ)

    slave (options forward-only - 4.9.3 or later)
- キャッシュの有効利用
  - データを特定のサーバに集約
  - トラフィックを抑える(回線が細いときなど)

sample of localhost.zone

```
; $ORIGIN       localhost.
@       IN      SOA     ns.wide.ad.jp.      postmaster.wide.ad.jp. (
                        1           ; Serial number
                        172800      ; Refresh every 2 days
                        3600        ; Retry every hour
                        1728000     ; Expire every 20 days
                        172800 ); Minimum 2 days
;
        IN      NS      localhost .
;
        IN      A       127.0.0.1
```

sample of localhost.rev

```
; $ORIGIN       0.0.127.in-addr.arpa.
@       IN      SOA     ns.wide.ad.jp.      postmaster.wide.ad.jp. (
                        1           ; Serial number
                        172800      ; Refresh every 2 days
                        3600        ; Retry every hour
                        1728000     ; Expire every 20 days
                        172800 ); Minimum 2 days
;
        IN      NS      localhost.
;
0       IN      PTR     loopback-net.
1       IN      PTR     localhost.
```

sample of wide.zone (cont.)

```
@               IN  SOA     ns.wide.ad.jp. two.wide.ad.jp. (
                            100627   ; Serial
                            3600     ; Refresh
                            900      ; Retry
                            3600000  ; Expire
                            3600     ; Minimum
                        )
                IN  A       203.178.136.63
                IN  NS      ns
                IN  NS      ns.tokyo
                IN  MX 10   sh
ns              IN  A       203.178.136.63
ns.tokyo        IN  A       203.178.136.61
```

sample of wide.zone (cont'd)

```
sh              IN  A       203.178.137.73

www             IN  CNAME   endo
endo            IN  A       203.178.137.71

localhost       IN  CNAME   localhost.

v6              IN  NS      ns1.v6
                IN  NS      ns2.v6
ns1.v6          IN  A       163.221.11.21
ns2.v6          IN  A       203.178.136.188
```

レコード定義の基本型

key [ttl] IN r-id value1 value2 ...

<左辺> <右辺>

- ttl - 省略可
  - 当該レコードのキャッシュ期間
- IN (class-ID) - Internet Domain
- r-id (resource-ID)
  - レコードの種類 (SOA, NS, A, MX, ...)
- value
  - そのレコードの値 (r-id によって形式が違う)

レコード定義の基礎知識

- 同一 key に対する定義
  - 後続の定義の key は省略可
- $ORIGIN <domain>
  - デフォルトのドメイン名の指定
  - 初期デフォルトは named.{boot,conf} の zone
- $INCLUDE <filename> [<domain>]
  - ファイルの挿入
- FQDN表記のホスト名の末尾には . を

SOA (Start Of Authority) RR

```
@  IN  SOA  <Pri-NS名>  <管理者メールアドレス>  (
        1           ; Serial
        172800   ; Refresh (2d)
        3600       ; Retry
        1728000 ; Expire (20d)
        172800   ; Minimum TTL (2d)
        )
```

- 管理者メールアドレスは @ を . に変える

SOA パラメータ (cont.)

- Serial
  - Sec-NSのデータ更新判定用
- Refresh (秒)
  - Sec-NSのSerialチェック間隔
- Retry (秒)
  - Refresh経過後のチェック間隔

SOA パラメータ (cont'd)

- Expire (秒)
  - サービス停止までのチェック不能期間
  - この状態で nslookup をすると

    *** ns.provider.ad.jp can't find x.co.jp.: Server failed
- Minimum TTL (time to live) (秒)
  - ゾーン内に定義される全レコードのデフォルト・キャッシュ期間

  (全NSに対して効果を持つ)

Serialについて

- 32ビット
- . による混乱に注意(使わない)
  - 1.01 = 100001 ("." は "000" と同値)
- 1997122501 など日付を使うと明瞭
  - 一日100回更新で4294年まで
- 上限なし(ループ):RFC1912(I)
  - 1に戻すことが可能
  - 2147483647(7fffffff)以内を2回足す

NS (Name Server) RR

- Pri-NS および Sec-NS を記述
  - 上位ゾーンでの記述が重要
    - Authorized Server
  - 上位ゾーンに記述がない
    - Unauthorized Server
- 該当する NS に対する A RR も記述
  - glue record (逆引き zone には不要)

```
$ORIGIN     ad.jp.
wide        IN  NS  ns.wide.ad.jp.    ;ad.jp.zone からの delegation
ns.wide     IN  A   203.178.136.63
```

lame (不完全な) NS

- Authorized だと思って問い合わせたら Unauthoritative answer が返ってきた
  - Delegation されている
  - Authorized NS ではない
- 実際の Authorized NS にアクセス不能な状況で、存在するはずのデータが存在しないとみなされる
  - メールが落ちる

A (Address) RR

- A RR
  - ホスト名からIPアドレスのマッピング

```
$ORIGIN         wide.ad.jp.
sh              IN A 203.178.137.73
```

MX (Mail eXchanger) RR

- MX RR
  - メールアドレスから配信先ホスト名へのマップ

```
$ORIGIN         wide.ad.jp.
@               IN MX 10 sh
```

- MX は A より優先(メールの配信)
- A を優先させたいとき
  - 1st-MX で転送

CNAME (Canonical NAME) RR

- ホストの別名定義

```
$ORIGIN  wide.ad.jp.
archie       IN CNAME sun3.tokyo
```

  - CNAME チェインはできるだけ避ける
  - 同一 key に別の種類のレコードを定義しない
  - 同一 key に複数の CNAME は定義しない
- NS, MX の右辺に CNAME で定義される名前を使わない

PTR (domain name PoinTeR) RR

- IP アドレスからホスト名へのマッピング
  - 逆引き

```
$ORIGIN      137.178.203.in-addr.arpa.
73           IN PTR sh.wide.ad.jp.
```

  - PTR レコード検索によるサービス制限
    - 引けないホストからのアクセス拒否
    - ドメイン名の確認
- うそつき問題
  - アドレス→ホスト名 の単方向だと騙れる
  - 引き直しチェック

nslookup で逆引きの確認

- ホストのIPアドレスが 1.2.3.4 のとき

  % nslookup

  > set q=ptr

  > 4.3.2.1.in-addr.arpa.

- 新しい (4.8.3 以降) nslookup

  % nslookup 1.2.3.4

ネットワーク名の定義

- RFC1101(?): DNS Encoding of Network Names and Other Types
- netstat -i, -r などで参照される

```
0.0.54.130.in-addr.arpa.    IN PTR kuins.kyoto-u.ac.jp.
                            IN A   255.255.0.0
kuins.kyoto-u.ac.jp.        IN PTR 0.0.54.130.in-addr.arpa.

0.0.0.224.in-addr.arpa.     IN PTR BASE-ADDRESS.MCAST.NET.
```

その他のレコード

- HINFO, TXT, WKS
  - HINFO は必ず2つ以上のパラメータを書く!
- NULL, MB, MG, MR, MINFO (experimental)
  - RFC1035(S)
- AFSDB, ISDN, RP, RT, X25
  - RFC1183(E)
- PX
  - RFC1664(E)

localhost/127.in-addr.arpa zone

- すべてのネームサーバに設定すべき
  - root server まで問い合わせるのは無駄

```
$ORIGIN      my.domain.jp.
localhost    IN CNAME localhost.
```

- 引き直しの際の不整合の防止
  - 127.0.0.1 → localhost.my.domain.jp にならないように

CIDRと逆引き管理

- class less なアドレスの割り当て
  - 192.0.2.0/25 - 組織Aに
  - 192.0.2.128/26 - 組織Bに
- 逆引きゾーンの管理単位問題
  - オクテット(8ビット)単位の権限委譲との不整合
- 解決策
  - CNAME で散らす
    - draft-ietf-dnsind-classless-inaddr-03.txt
  - NS で散らす

Classless IN-ADDR.ARPA delegation (cont.)

- **上位ゾーンからの権限委譲**

```
$ORIGIN 2.0.192.in-addr.arpa.
;  <<0-127>> /25
0/25  NS  ns.A.domain.jp.
1       IN  CNAME  1.0/25.2.0.192.in-addr.arpa.
2       IN  CNAME  2.0/25.2.0.192.in-addr.arpa.
 :
126    IN  CNAME  126.0/25.2.0.192.in-addr.arpa.
```

Classless IN-ADDR.ARPA delegation (cont'd)

- **当該ゾーンでの定義**

```
$ORIGIN 0/25.2.0.192.in-addr.arpa.
@       IN  SOA ...
        IN  NS  ns.A.domain.jp.
1       IN  PTR  host1.A.domain.jp.
2       IN  PTR  host2.A.domain.jp.
 :
126     IN  PTR  host126.A.domain.jp.
```

DNS と IPv6 (cont.)

- IPv6 アドレスは 128 ビット長
- CIDR で発生する問題をどう解決するか
  - 逆引きの区切りは 4 ビット単位
    - 16進表記
  - IPv4 と同じ方式を利用する

```
sh.v6.wide.ad.jp.                    IN  AAAA  3ffe:501:1000::1
$ORIGIN  0.0.0.1.1.0.5.0.e.f.f.3.ip6.int.
1.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0.0 IN PTR sh.v6.wide.ad.jp.
```

DNS と IPv6 (cont'd)

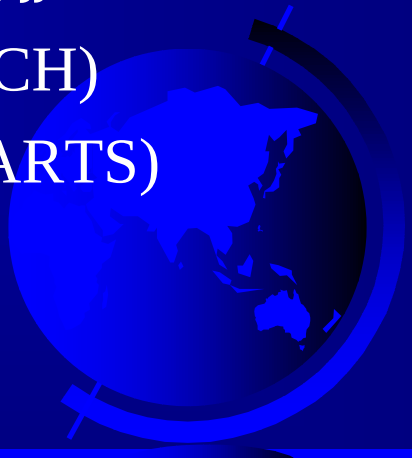
- 古いnamedによるセカンダリ
  - ゾーンの転送がうまくいかない
  - 少なくとも bind 4.9.3以降にする
- 古い sendmail(?)
  - AAAA RRを拾うと v4 宛てのメールが落ちる
    - additional information で送られてくる
- まだ運用系に定義しない方が安全か

メールアドレスの補完(cont.)

- MX RR と A RR を用いる
  - ワイルドカードMX問題
- /etc/resolv.conf に定義

domain sub.x.co.jp

  - search sub.x.co.jp x.co.jp co.jp と同値
    - 遡って3階層分調べる (MAXDFLSRCH)
    - 最短は2レベル (LOCALDOMAINPARTS)
      - JP domain の実状にあわない
  - RFC1535(I) で暗黙の遡りを禁止



メールアドレスの補完(cont'd)

search sub1.x.co.jp sub2.x.co.jp x.co.jp

- LOCALDOMAIN 環境変数によるユーザ設定
  - 最大6ドメイン (MAXDNSRCH)
- 検索の順序

  nic.ad.jp

  nic.ad.jp.sub.x.co.jp

  nic.ad.jp.x.co.jp

  nic.ad.jp.co.jp
  - RFC1535(I)より前は nic.ad.jp を最後に検索

Wildcard MX is harmful

- exact RR が存在しない場合にマッチ
- 存在しないアドレスにもメールが飛ぶ
  - 送信時に存在しないアドレスであることが不明
- 存在しないアドレスに補完される
  - ResolverOptions に HasWildcardMX を定義
    - sendmail.cf
- 配信先に対応するMX RRが引けない
  - 配信先ホスト名の最後に必ず . を補う

→ どうしても必要な場合にのみ利用する

古いglueレコードが消えない

- 4.8.3以前?
- server A: primary of x.co.jp
- server B: primary of sub.x.co.jp
  - お互いにセカンダリになっている
- x.co.jp の NS (server C)のアドレスを変更
- server C の古い glue レコードが消えない
  - server A で消しても
  - server B のからの zone transfer で甦る

  → セカンダリコピーからも消す

サーバが報告するエラー (cont.)

- bad referral
  - NS があるのに SOA がない
- NS points to a CNAME
- MX points to a CNAME
- dangling CNAME pointer
  - CNAME の先が何も指していない
- Lame server on 'x.co.jp'
  - Authorized のはずなのに、Unauthoritative answer が返ってきた

サーバが報告するエラー (cont'd)

- Response from unexpected source
  - 違うインターフェースアドレスからの応答?
  - アタック?
- zone "xxx" (class 1) SOA serial# (nn) is < ours (mm)
  - SOA serial が減った!

RFC1912(I): Common DNS Operational and Configuration Errors

デバッグ用ツール (cont.)

RFC1713: Tools for DNS debugging

- Host (bind 8 に添付)
  - ftp://ftp.nikhef.nl/pub/network/host_YYMMDD.tar.Z
- Dnswalk (bind 8 に添付)
  - ftp://ftp.pop.psu.edu/pub/src/dnswalk
- Lamers
  - ftp://terminator.cc.umich.edu/dns/lame-delegations/

デバッグ用ツール (cont'd)

- Doc (Domain Obscenity Control)
  - ftp://ftp.uu.net/networking/ip/dns/doc.2.0.tar.Z
- DDT (Domain Debug Tools)
  - ftp://ns.dns.pt/pub/dns/ddt-2.0.1.tar.gz
- Checker
  - ftp://catarina.usc.edu/pub/checker
- Dig (bind 8 に添付)

設定変更の作業手順

- ドメイン名の変更
  - メールアドレスの二重運用
- ドメインのIPアドレスの変更
- ネームサーバの変更(別のホストに)
- ネームサーバのIPアドレスの変更
  - RR に新旧を定義
- メールサーバの変更(別のホストに)
  - sendmail.cf で新しい方に転送
- メールサーバのIPアドレスの変更

ネームサーバのIPアドレス変更 (準備, cont.)

- 新アドレスが利用されていないことを確認
- セカンダリ･サーバの管理者に通知
  - named.boot でコピー元アドレスに新旧アドレスを記述
- 変更予定の A RR の TTL を短く(5分)
- SOA の Refresh と Retry を短く
- 新アドレスで逆引きを設定

ネームサーバのIPアドレス変更 (準備, cont'd)

- A RR の変更
  - 当該ゾーン、上位ゾーン
  - 新旧両方を登録する方法もある
- しばらくの間両方のアドレスにアクセスが来る

ネームサーバのIPアドレス変更 (完了後, cont.)

- 新アドレスの利用開始
- 旧アドレスはタイムアウトまで利用しない
- A RR の TTL, Refresh, Retry の値を戻す
  - Serial を上げるのを忘れずに

ネームサーバのIPアドレス変更 (完了後, cont'd)

- セカンダリ･サーバの管理者に完了の通知
- 旧アドレスの消去
  - glue record に残っていないか確認
  - 他ゾーンのセカンダリになっているとき
    - 変更依頼
  - 古いglueレコードが消えない問題

DNSの今後

- Dynamic Update
  - レコード単位のデータ更新
- Incremental Zone Transfer (IXFR)
  - トラフィックの削減と更新速度の向上
- Security Extention
  - SIG RR, NXT RR