dnsdistと
NSDとUnboundで
BINDのふりをさせる話

東　大亮

2016-12-01

Internet Week 2016 DNSOPS.JP BoF

私について

- UnboundやNSDを魔改造して遊んでいる人です
- 最近knot DNSに手を出してますが話が通じなくて困っています
- 個人としての活動・発表です

BINDからの卒業が唱えられて久しいですが

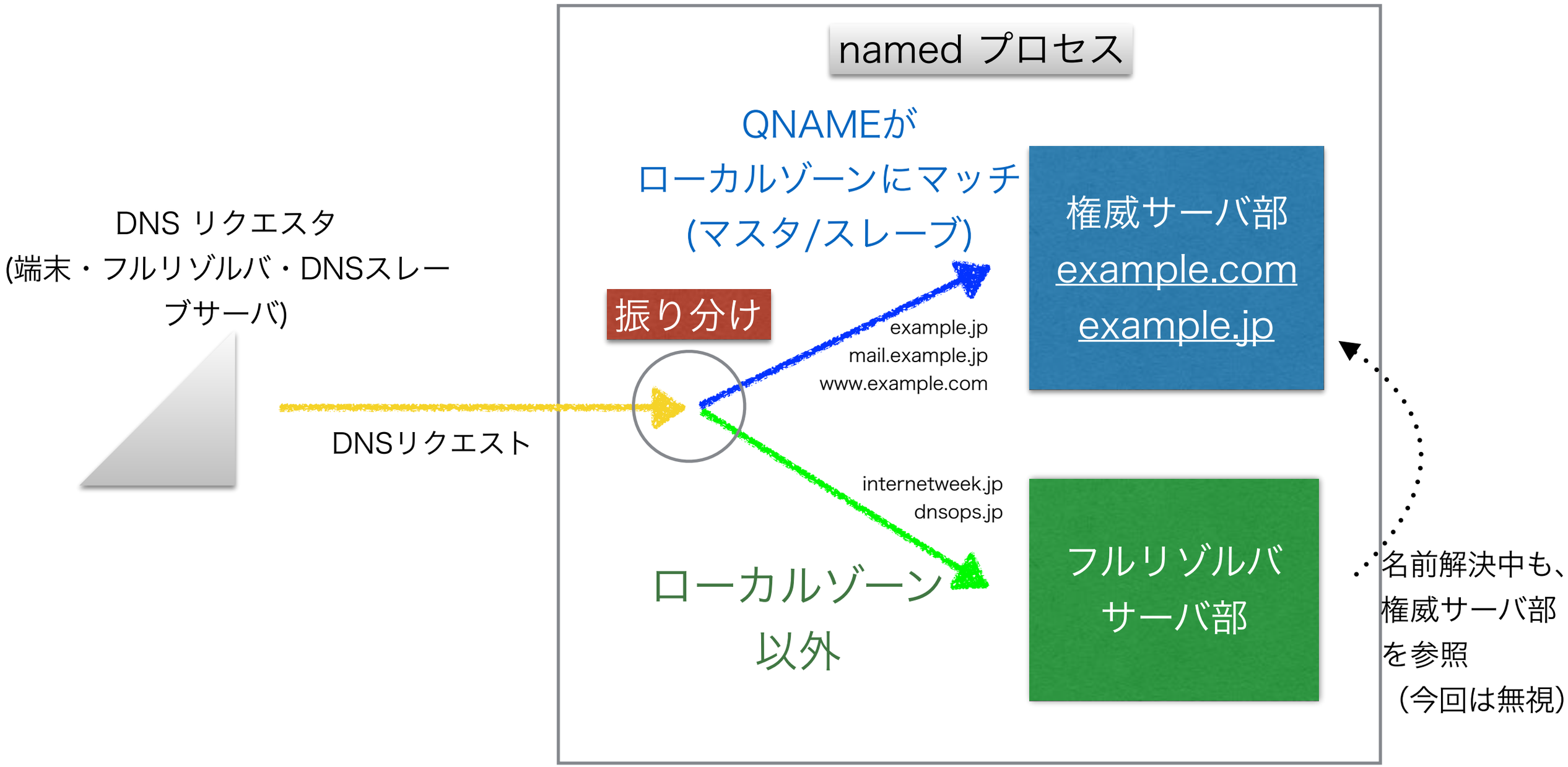
- 結局どうすりゃいいんだよ！！！
- BINDが動いている環境（ネットワーク・ホスト・利用者の設定）をあまり変えずに、NSDとUnboundにリプレイスすることを考えてみる

BINDとは何か

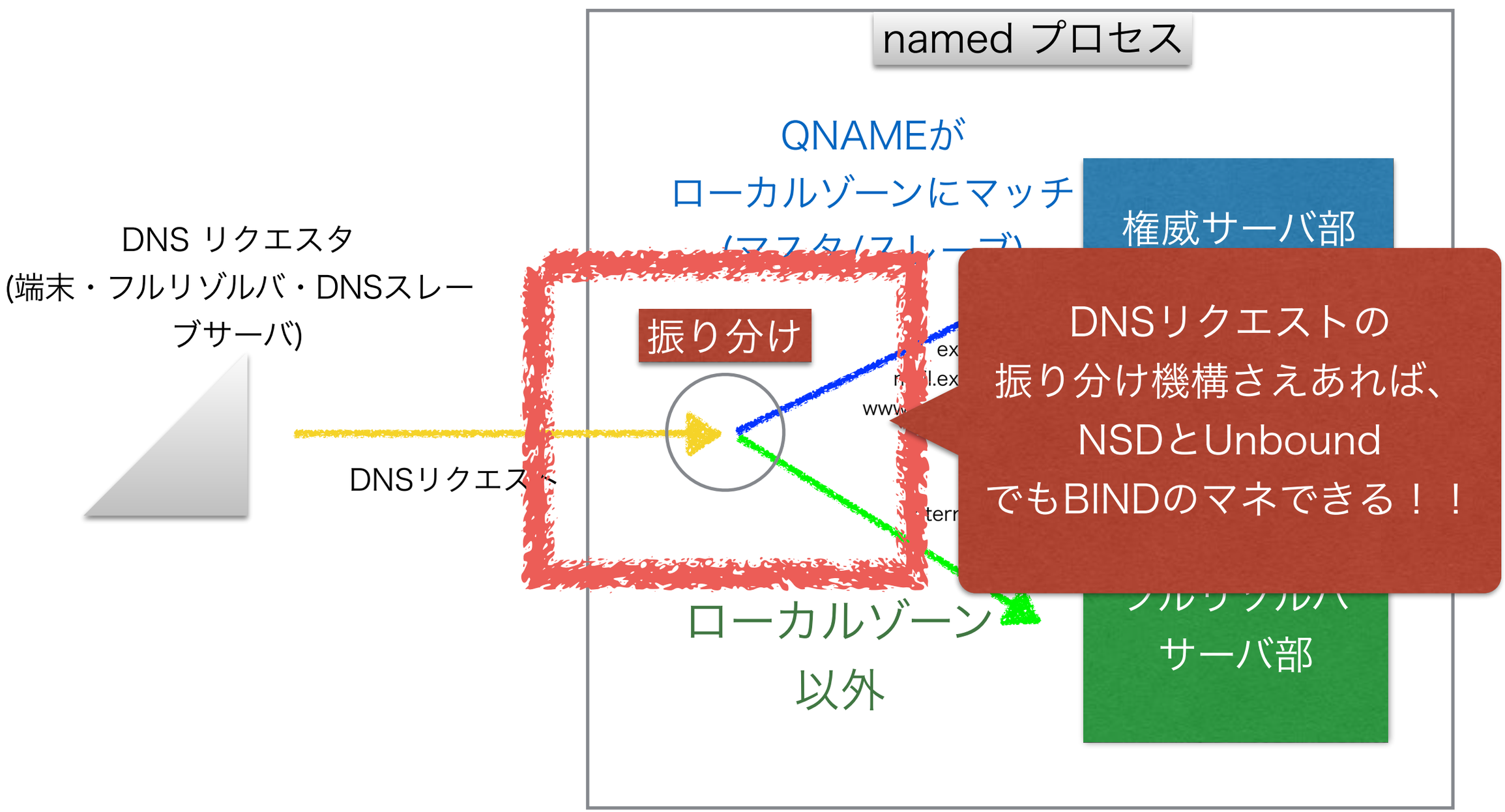
- 外見の動作としては、要するに、権威サーバと、キャッシュサーバ（フルリゾルバサービス）が合体したもの

- RFC1034 4.3.2 に書かれている Name Serverのアルゴリズムを比較的忠実に実装

BINDとは何か


DNS リクエスタ
(端末・フルリゾルバ・DNSスレーブサーバ)
DNSリクエスト
named プロセス
QNAMEが
ローカルゾーンにマッチ
(マスタ/スレーブ)
振り分け
example.jp
mail.example.jp
www.example.com
internetweek.jp
dnsops.jp
ローカルゾーン
以外
権威サーバ部
example.com
example.jp
フルリゾルバ
サーバ部
名前解決中も、
権威サーバ部
を参照
（今回は無視）

BINDとは何か

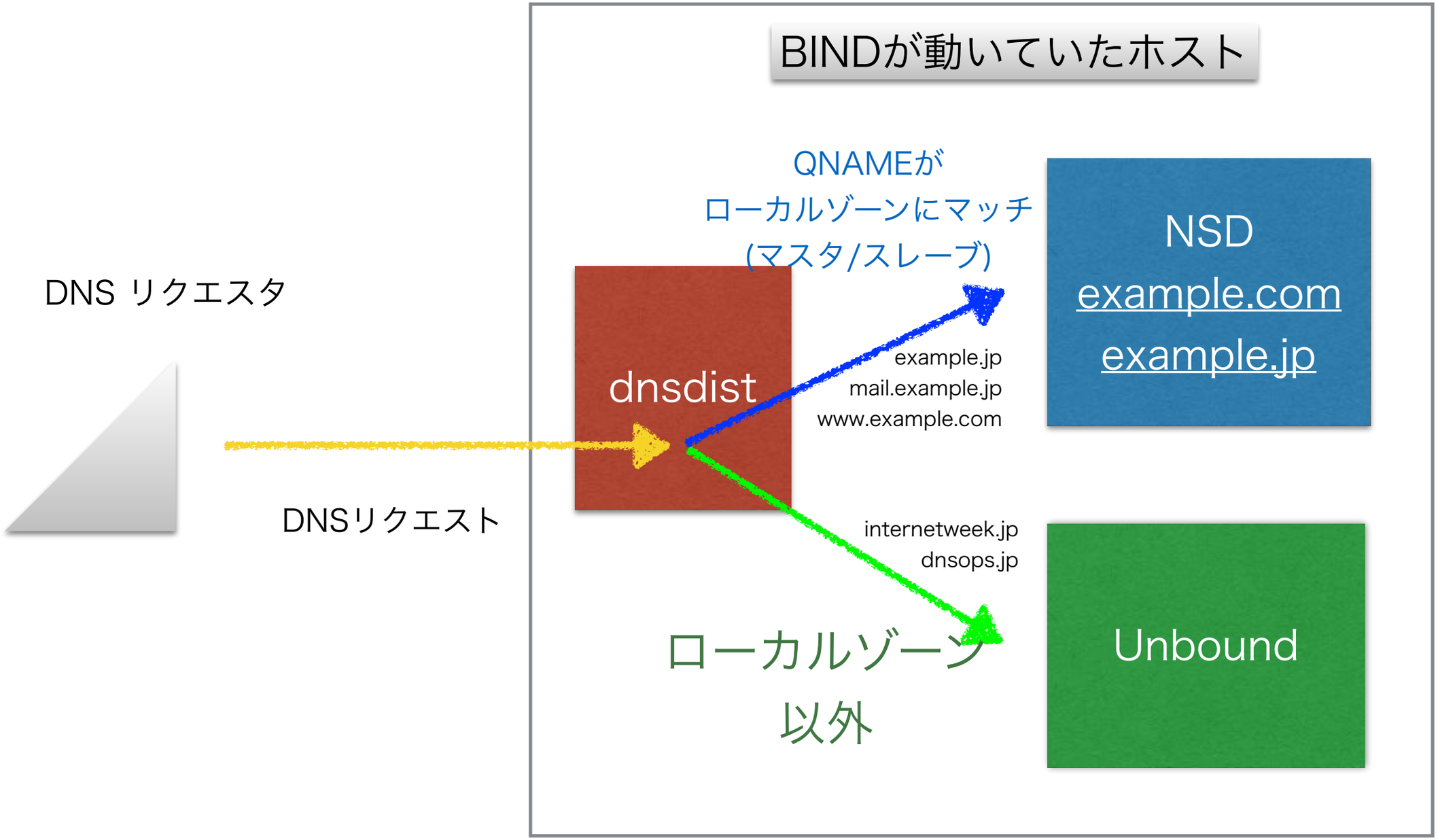

DNS リクエスタ
(端末・フルリゾルバ・DNSスレーブサーバ)
DNSリクエスト
namedプロセス
QNAMEが
ローカルゾーンにマッチ
振り分け
ローカルゾーン
以外
権威サーバ部
フルリゾルバ
サーバ部
DNSリクエストの
振り分け機構さえあれば、
NSDとUnbound
でもBINDのマネできる！！

dnsdist

- DNSに特化したソフトウェアロードバランサ
  - 蘭PowerDNS.COM B.V. が開発。GPLv2 — http://dnsdist.org
  - DNSリクエストを受信して、複数のバックエンドDNSサーバにフォワードする
  - DNSリクエストの内容(QNAME, ソースIPアドレス, QTYPE etc)に従い、どのバックエンドDNSサーバにフォワードするか細かく指定できる
  - 基本的にルールベースでDNSリクエストの扱いを指定するが、Lua言語のスクリプトで複雑な処理も可能

DNSリクエストの振り分け機構としてdnsdistを使ってみる

dnsdistでリクエストを NSDとUnboundに振り分け



named.confから振り分けルールを作る例

named.conf

```
options {
        allow-recursion { 192.168.0.0/16; };
};

zone "example.com" {
        type master;
        file "example.com.zone";
};
zone "example.jp" {
        type slave;
        masters { 1.1.1.1; };
};
```

振分ルール：

① QNAMEが example.comとexample.jp (子孫のドメイン名含む)にマッチするものは権威(NSD)へ

② それ以外のQNAMEで、ソースIPが 192.168.0.0/16 にマッチするものは、リゾルバ(Unbound)へ

③ 上記いずれもマッチしないリクエストは、REFUSE (RCODE=5)で応答

dnsdistの設定例

```
newServer({address="127.0.0.1:40000", pool="auth"})
newServer({address="127.0.0.1:40001", pool="resolver"})

authdomains = newSuffixMatchNode()
authdomains:add(newDNSName("example.com."))
authdomains:add(newDNSName("example.jp."))

allow_recursion = newNMG()
allow_recursion:addMask("192.168.0.0/16")

addAction(SuffixMatchNodeRule(authdomains), PoolAction("auth"))
addAction(NetmaskGroupRule(allow_recursion), PoolAction("resolver"))
addAction(AllRule(), RCodeAction(5))

addACL("0.0.0.0/0")
addACL("::0/0")
addLocal("0.0.0.0:53")
addLocal("[::]:53")
```

振り分け先DNSサーバの定義
(authはNSD、resolverはUnbound)

ローカルゾーンの定義

resolverにフォワードを許可するソースIPアドレスを定義

振分ルールの定義

振分ルール：

① QNAMEが authdomains = (“example.com", “example.jp") (子孫のドメイン名含む)にマッチするものは権威(NSD = auth)へ

② それ以外のQNAMEで、ソースIPが allow_recursion =192.168.0.0/16 にマッチするものは、リゾルバ(Unbound = resolver)へ

③ 上記いずれもマッチしないリクエストは、REFUSE (RCODE=5)で応答

NSDとUnboundの設定

nsd.conf

```
server:
 ip-address: 127.0.0.1@40000

zone:  # master zone
 name:"example.com."
 zonefile: "example.com.zone"

zone:  # slave zone
 name:"example.jp."
 request-xfr: 1.1.1.1 NOKEY
```

127.0.0.0:**40000**でリクエストを受信する以外は普通のNSDと同等の設定

unbound.conf

```
server:
 interface: 127.0.0.1@40001
 access-control: 127.0.0.1 allow
```

127.0.0.0:**40001**でリクエストを受信する以外は普通のUnboundと同等の設定

dnsdistを使ってリクエスト処理する場合の注意点

- NSD/Unboundから見えるDNSリクエストのソースIPアドレスはすべて dnsdistのソースIP (127.0.0.1)に変換されてしまう

- ソースIPアドレスによるアクセス制限(allow-query, allow-recursion, allow-transfer)はdnsdistで行う必要がある

- リクエストログでソースIPアドレスを記録したい時も同様

named.confからdnsdist, NSD, Unboundの設定を生成するツール

- 作ってみました：bind2other
  - https://github.com/hdais/bind2other
  - Pythonスクリプト (Python 2.6)
  - PLY (Python Lex/Yacc)を使って、named.confを構文解析し、それとなるべく同等の動きをするdnsdist, NSD, Unboundの設定ファイルを作る
  - PLYは同梱しているので git cloneすれば大抵の環境で動くはず

bind2other: 使い方

```
$ ./bind2other.py named.conf
```

カレントディレクトリに dnsdist.conf, nsd.conf, unbound.conf ができるので、それでdnsdist/NSD/Unboundを起動

```
$ sudo nsd -c nsd.conf
$ sudo unbound -c unbound.conf
$ sudo dnsdist -C dnsdist.conf -d
```

```
dig @127.0.0.1 example.com    # ローカルゾーン
dig @127.0.0.1 www.google.com # ローカルゾーン以外
```

bind2other: 実装済みのBIND機能

optionsは以下を実装
- allow-query
- allow-recursion
- allow-transfer
- directory

```
options {
  allow-query { … };
  allow-recursion { … };
  allow-transfer { … };
  directory “…”;
};
```

zoneはmaster, slaveのみ、
zone毎のallow-transferをサポート

```
zone “master-zone-name” {
  type master;
  file “…”;
  allow-transfer { … };
};
```

```
zone “slave-zone-name” {
  type slave;
  masters { … };
  allow-transfer { … };
};
```

allow-query 等に使用できるacl文も使用可能。
acl のネストも可

```
acl acl_name1 { 10.0.0.1; };
acl acl_name2 { acl_name1; 10.0.0.2; };
```

bind2other: 禁断のview

view は以下を実装

match-clients (リクエストのソースIPアドレスでマッチ)

view固有の allow-query, allow-transfer, allow-recursion

viewのマスタ、スレーブゾーン

```
view view_name {
  match-clients { … };
  allow-query { … };
  allow-transfer { … };
  allow-recursion { … };
  master_zone;
  slave_zone;
};
```

bind2other: できないこと

- 今まで説明した機能以外は実装していません
  - ごく一部の機能しか実装していないので、皆さんがお使いの named.confを読ませても、エラーになる可能性が高い
  - rndcも未実装
  - ログもdnsdist, NSD, Unboundそれぞれの形式で出力
  - dnsdistは設定ファイルのreload機能がないため、bind2otherで設定ファイルを再生成したら、dnsdistを再起動してdnsdistを再読込させる必要がある

まとめ

- dnsdistと、NSD/Unbound (他のDNSサーバ実装でも可) を組み合わせれば、BINDの動作をある程度マネできる

- キャッシュと権威の同居も、viewも可能

- でも、そんなことはやめましょう

  - 結局BINDと同じ動きをするのはBINDしかありません

  - 脱BINDしたい方は、素直に権威とキャッシュの分離をがんばってやりましょう

おわり