DNS zone split-view configuration with BIND-DLZ

Personally I stopped working with pure file-based DNS servers long time ago.
Earlier this year I committed an article about BIND-DLZ to german freeX magazine on how to make DNS servers really dynamic. While I'm not permitted to publish my original article (or parts of it) online, I'm just following it up in some way, as I have another interesting story to tell.
Today I'm looing into DNS zone split-view on top of BIND-DLZ -- or in other words: How can my BIND server be authoritative on the public network interface, while be recursive only (thus, ignore any local authoritative zones) on the local network interface?

Some may say that this is easy, just run two instances of BIND on the same machine.

Well, yes, but this does apply to each and every situation.
Splitting the authoritative server and the cache has it's benefits (and I'd recommend to do this in most cases), especially when it comes to service availability and - to some degree - protection against DDoS. Though it has some drawbacks, amongst them the need for more resources, either in the form of separate boxes, virtual servers or even running the daemon twice on the same box.
In some scenarious, you're just bound to the fact, that you're limited in resources and simply can't run two daemons. Or you need to take extra steps to have your operating system support running two daemons concurrently. Most do not allow that out-of-the-box, so you'll find yourself struggling with copying init-scripts and directory structures. This may even break on the next distro upgrade, so maybe it's better to not even bother ;-)

Si, even if it may not be best practice, I want to look into a all-in-one solution for the scenario below:

I have a server, that is connected to the internet on one network interface. The machine acts as authoritative nameserver for a few domains, thus effectively hosting some DNS zones and being the master for them.
As I'm not too happy with static DNS zone files, I configured the host to serve out it's authoritative zone data dynamically through the BIND-DLZ (dynamically loadable zones) extension.
Let's call this the 'external' view, so it's actually what we see from the public internet.

Now, the very same server runs some services locally, like an MTA (mail transfer agent) for example.
The DNS server also listens to the loopback interface 127.0.0.1, which was added as first name server in /etc/resolv.conf. Why should we use an external name server if we have one locally already?

Everyone familiar with this kind of topic may notice, that this could actually lead to some issues in a real production environment. Let's asume that a new DNS zone for 'microsoft.com' is being added to our authoritative name server.

Since we didn't take any special precaution, the following will happen when we send an email message to microsoft.com through our local MTA: First the MTA will look in /etc/resolv.conf for any valid name server. Since we listed 127.0.0.1 as first name server, that's the one being queried first.

Remember that we added 'microsoft.com' as authoritative (primary) zone to our local name server?

Well, for sure, we wouldn't want to host 'microsoft.com' on our own system. That can only be wrong. You may as well substitute any other domain here, which will propably never be hosted with you (yahoo.com, google.com, etc). But as soon as you run some sort of hosting business, you may well end up in situation, that some dumbass will eventually bypass your nice input filters ;-)

So we end up with our own local name server replying to us with a obviously wrong SOA for the zone 'microsoft.com'. Even worse, we get wrong MX records as well, so our MTA will deliver the messages to a totally wrong destination. Sure not what we want!

To solve this when running with traditional BIND zone files, you'd simply extended your named.conf to have multiple views. So you'd add your authoritative zones to an outside view (requests coming through the public/outside interface), while adding an inside

or local view (for requests coming through the local network or loopack interface), where no zones are to be added.

So the question arises if it is possible to incorporate this kind of setup with BIND-DLZ zones setup?

After some testing, the answer to this is clearly yes. Though I must admit I have only tested this with DLZ's file-system driver, I think it should actually work with other DLZ drivers as well.

Let's look at a very basic named.conf (I omitted most of the options for better readability) with DLZ:

```
options {
 directory "/etc/namedb/working";
 pid-file "/var/run/named/pid";
 dump-file "/var/dump/named_dump.db";
 statistics-file "/var/stats/named.stats";
 listen-on { any; };
};

zone "." { type hint; file "/etc/namedb/named.root"; };

dlz "file system zone" {
 database "filesystem ./dns-root/ .dns .xfr 0 ~";
};
```

This would just enable a authoritative and recursing name-server, which has it's dynamic DLZ zones within the 'dns-root' directory, but with that exact behaviour as mentioned before.

So, we should now add two views, and external view and a internal view. How do we take them apart? Simple answer: by their source address. Since we run our local cache only through the localhost interface, any request coming from that source should be directed into the internal view.
On the other hand, all requests not coming from the localhost source shall be put into the external view. It may especially important that we then disable recursion on the external view.

```
options {
 directory "/etc/namedb/working";
 pid-file "/var/run/named/pid";
 dump-file "/var/dump/named_dump.db";
 statistics-file "/var/stats/named.stats";
 listen-on { any; };
};

view "external" {
      match-clients { !127.0.0.1; any; };

      recursion no;

 zone "." { type hint; file "/etc/namedb/named.root"; };
```

```
 dlz "file system zone" {
  database "filesystem ./dns-root/ .dns .xfr 0 ~";
 };
};


view "internal" {
 match-clients { 127.0.0.1; !any; };

 recursion yes;

 zone "." { type hint; file "/etc/namedb/named.root"; };
};
```

Now let's check if this holds true. First I query my authoritative domain 'example.org' through the hosts public IP on 192.0.2.11

```
$ dig @192.0.2.11 example.org

; <> DiG 9.7.3-P3 <> @192.0.2.11 example.org
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 30099
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 1, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;example.org.   IN A

;; AUTHORITY SECTION:
example.org.  3600 IN SOA mynameserver.example.org. hostmaster.example.org. 2004070101 10800 3600 604800 86400
```

Now I do the same for the domain 'microsoft.com':

```
$ dig @192.0.2.1 microsoft.com

; <> DiG 9.7.3-P3 <> @192.0.2.1 microsoft.com
; (1 server found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: REFUSED, id: 15463
;; flags: qr rd; QUERY: 1, ANSWER: 0, AUTHORITY: 0, ADDITIONAL: 0
;; WARNING: recursion requested but not available

;; QUESTION SECTION:
;microsoft.com.   IN A
```

As expected I get no reply (actually it's refused, since recursion is turned off).

Now I query again for example.org, but this time not on the external but on the loopback interface.

```
$ dig @localhost example.org

; <> DiG 9.6.-ESV-R3 <> @localhost example.org
; (2 servers found)
;; global options: +cmd
;; Got answer:
;; ->>HEADER<<- opcode: QUERY, status: NOERROR, id: 49769
;; flags: qr rd ra; QUERY: 1, ANSWER: 1, AUTHORITY: 2, ADDITIONAL: 3

;; QUESTION SECTION:
;example.org.   IN A

;; ANSWER SECTION:
example.org.  172800 IN A 192.0.43.10

;; AUTHORITY SECTION:
example.org.  172800 IN NS b.iana-servers.net.
example.org.  172800 IN NS a.iana-servers.net.

;; ADDITIONAL SECTION:
a.iana-servers.net. 1800 IN A 199.43.132.53
a.iana-servers.net. 1800 IN AAAA 2001:500:8c::53
b.iana-servers.net. 1800 IN A 199.43.133.53
```

Here I get the reply through dns recursion. Thus indeed my local zone for 'example.org' is completely ignored, despite we run the authoritative zone from the same instance.

So to summarize: Even when running my authoritative zones off BIND-DLZ, adding views to BIND can still divide one single BIND instance into authoritative and recursing-only instance without interfering with local services, even if a domain is erronously added the the authoritative view.

As a plus, I tested if it's possible to run different BIND-DLZ namespaces for external and internal views. The good news: This also works fine. The config, in the example of the DLZ file system driver, may look like this:

```
options {
 directory "/etc/namedb/working";
 pid-file "/var/run/named/pid";
 dump-file "/var/dump/named_dump.db";
 statistics-file "/var/stats/named.stats";
 listen-on { any; };
};
```

```
view "external" {
      match-clients { !127.0.0.1; any; };

      recursion no;

 zone "." { type hint; file "/etc/namedb/named.root"; };

 dlz "file system zone" {
  database "filesystem ./dns-root-ext/ .dns .xfr 0 ~";
 };
};


view "internal" {
 match-clients { 127.0.0.1; !any; };

 recursion yes;

 zone "." { type hint; file "/etc/namedb/named.root"; };

 dlz "file system zone" {
  database "filesystem ./dns-root-int/ .dns .xfr 0 ~";
 };
};
```

In this example, I just added two separate DNS root directories, each of which can hold different authoritative zone data. Thus you can really take full adtvange of the BIND split views by having the very same domains an external and an internal set of records, each separate of the other.

Quiet cool, isn't it? :-D