BIND 9.11 Update

August 31, 2016

Presenter

Vicky Risk

Product Manager

BIND Core Team

sw eng, team lead

Evan H.

sr sw eng

Mark A.

sw eng

Mukund S.

sw eng

Witold K.

director of sw eng

Stephen M.

qa manager

Jeremy R.

qa engineer

Curtis B.

research fellow

Ray Bellis

New in 2015

sw eng, team lead

Evan H.

sr sw eng

Mark A.

sw eng

Mukund S.

sw eng

Witold K.

director of sweng

Stephen M.

qa manager

Jeremy R.

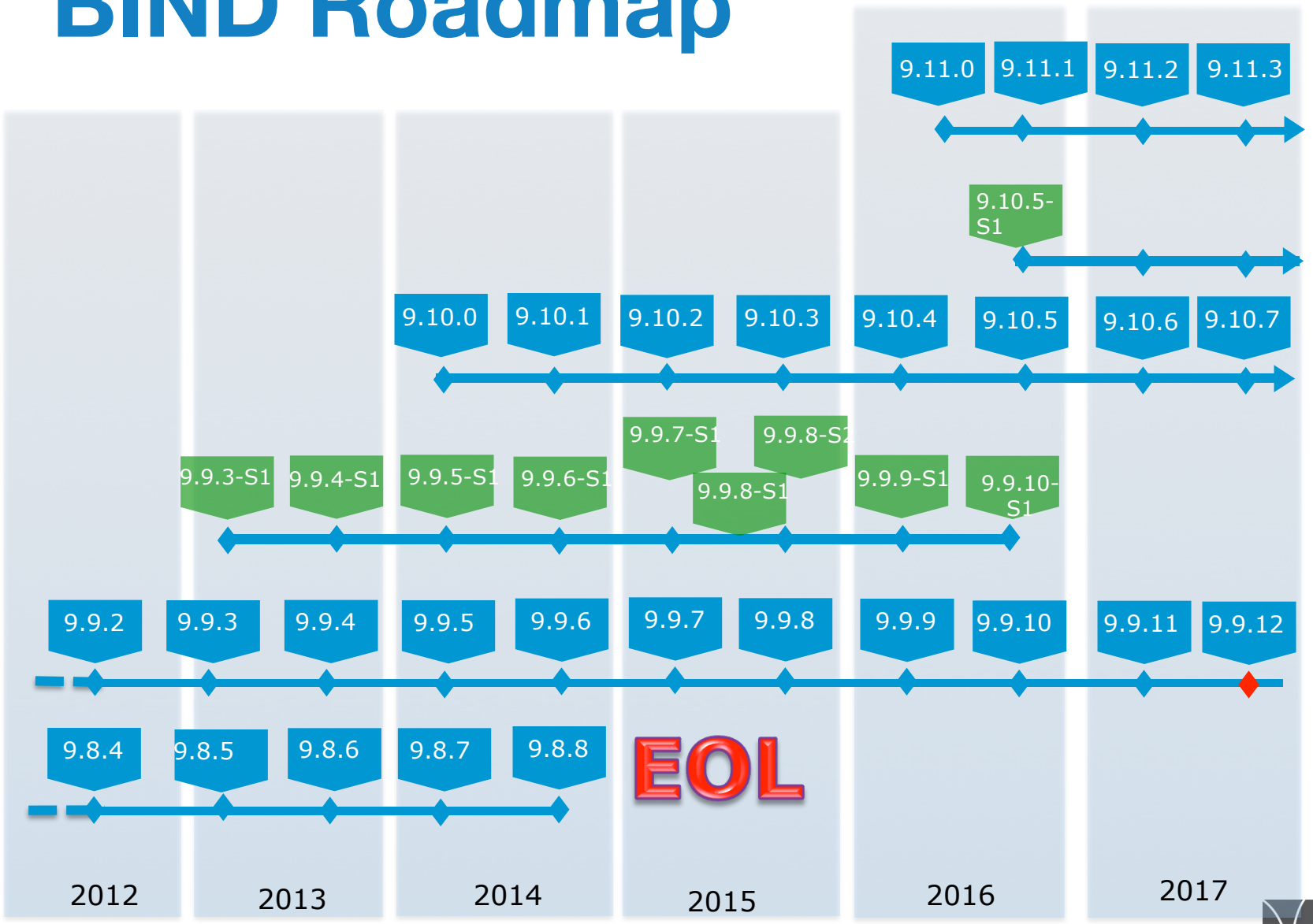
qa engineer

Curtis B.

research fellow

Ray Bellis

BIND Roadmap



Added in 9.11*

Bug fix	Feature change	Other (libraries, tests, docs)
263	106	39

S0	S1	S2	S3	S?
11	38	121	37	56

* some features and most fixes have appeared in prior maintenance versions

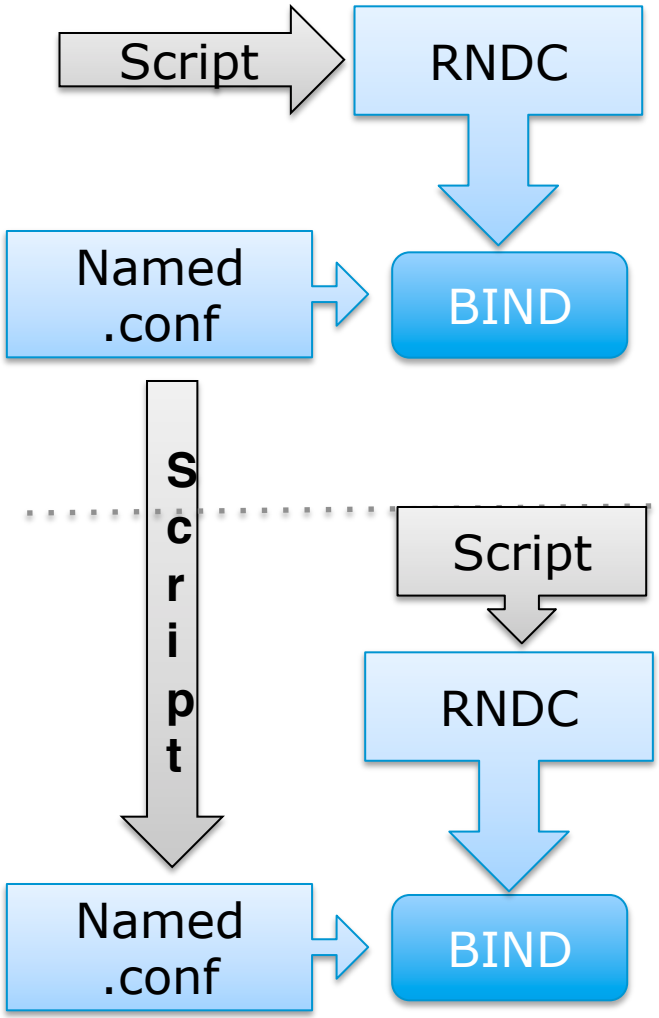
New in BIND 9.11

- **Zone Provisioning improvements**
  - Catalog zones
  - RNDC updates
  - NZF w/ LMDB
  - notify rate
  - DynDB
- **DNSSEC**
  - Negative trust anchor
  - keymgr utility
  - CDS, CDSKEY generation
- **dnstap**
- **DNS Cookies**
- **minimal ANY**
- **IPv6 bias**

Provisioning challenges

- Updating zone list across a large pool of slaves
- notify traffic overhead, particularly with a multi-tiered system
- RNDC designed for human interaction, being used by scripts
- Zones added via RNDC very slow to delete

User Scripts



IXFR/AXFR maintain zone CONTENTS only, for zones already configured on slaves

Left to the user to create a script or process for maintaining slave zone lists

Master

Slave

User scripts frequently use RNDC as an automation interface

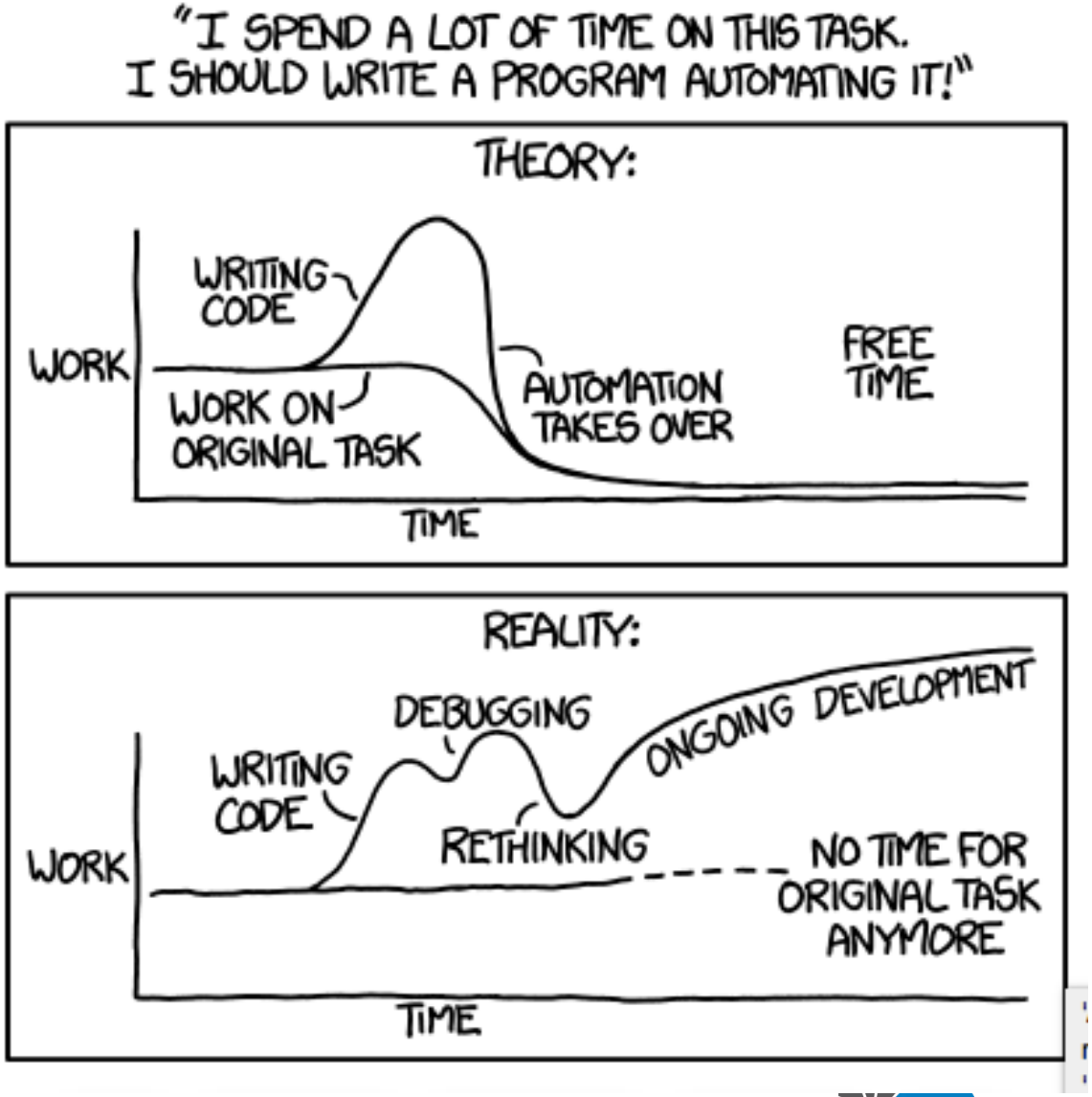
Or use 'include' files to maintain slave named.conf files

Scripts

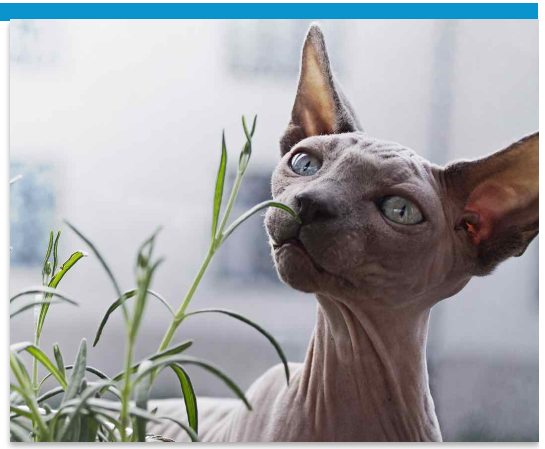
User Scripts
→MAINTENANCE

User Scripts
→BREAKAGE

User Scripts
→CONSULTING OPPTY



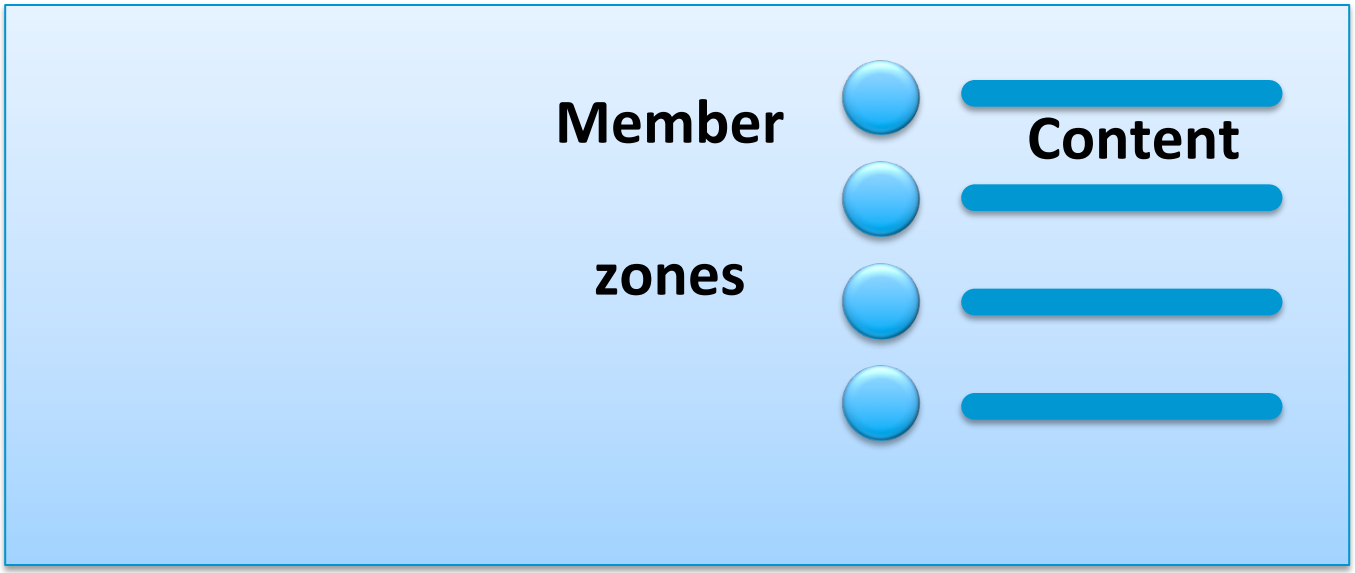
Catalog Zone



- a new zone on the master
  - in a special new format
  - contains a list of zones (the CatZ)
- updates to this zone are propagated to slaves, via IXFR/AXFR, adding and deleting zones

Add Zones to Master

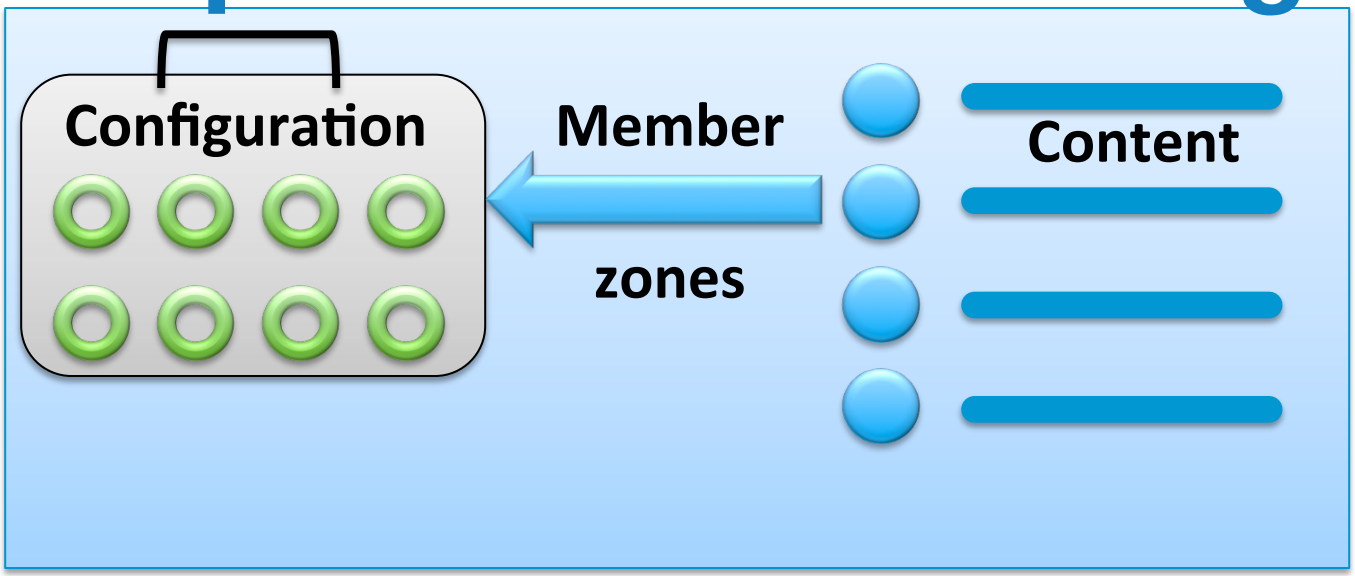
Master or Primary



Slave or Secondary

Create/Update Zone Catalog

Master or Primary



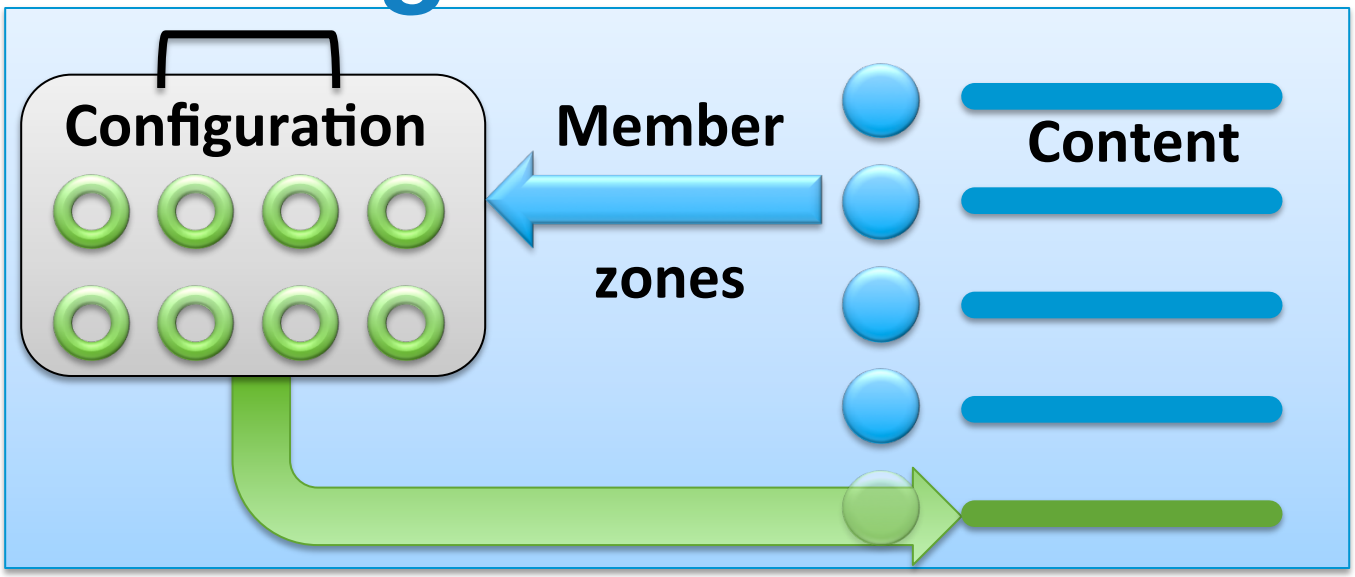
Slave or Secondary

python ./catz-add.py example2.com

adds the zone to both the master and the catalog zone at the same time

Catalog is a Zone

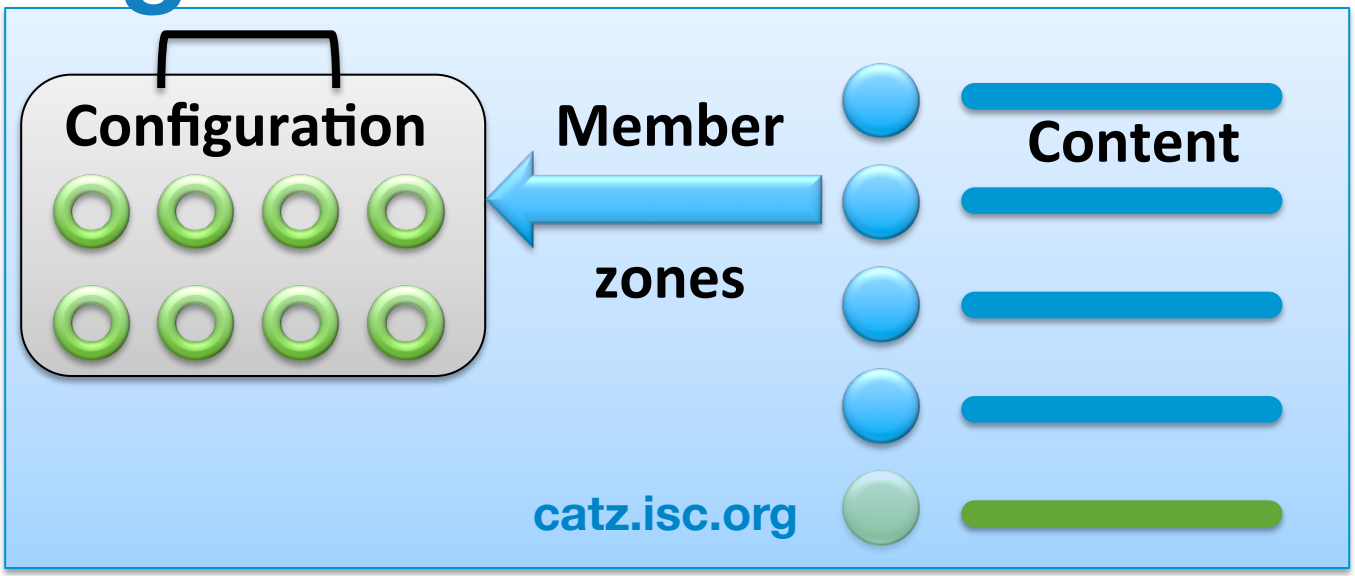
Master or Primary



Slave or Secondary

Configure CatZ on Slave

Master or Primary



Slave or Secondary

```
catalog-zones {
        zone “catz.isc.org”;
```

Configuration

Master

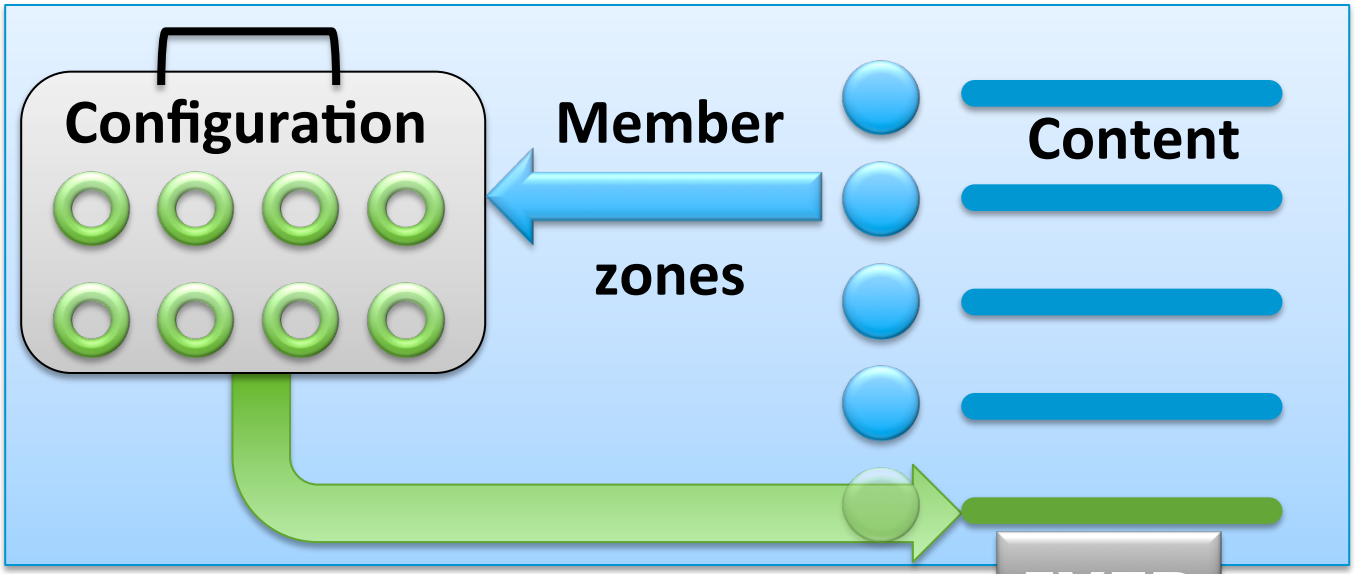
```
options {
        listen-on {
        10.53.0.1;
        };
        allow-new-zones yes;
};

zone “catz.isc.org” {
        type master;
        file “catz.isc.org.db”;
        allow-transfer {
        10.53.0.2;
        };
};
```

Slave

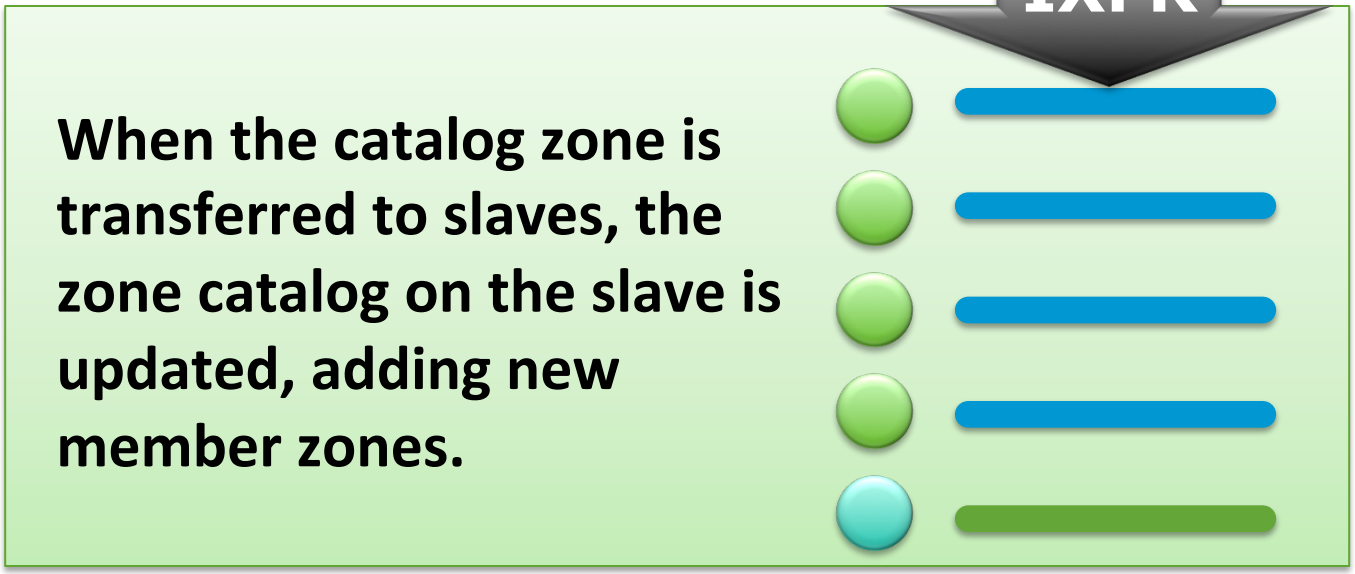
```
options {
        listen-on {
        10.53.0.2;
        };
        allow-new-zones yes;
        catalog-zones {
        zone “catz.isc.org”;
        };
};
zone “catz.isc.org” {
        type slave;
        masters {
        10.53.0.1 }
        };
};
```

Zones added to Slave

Master or Primary

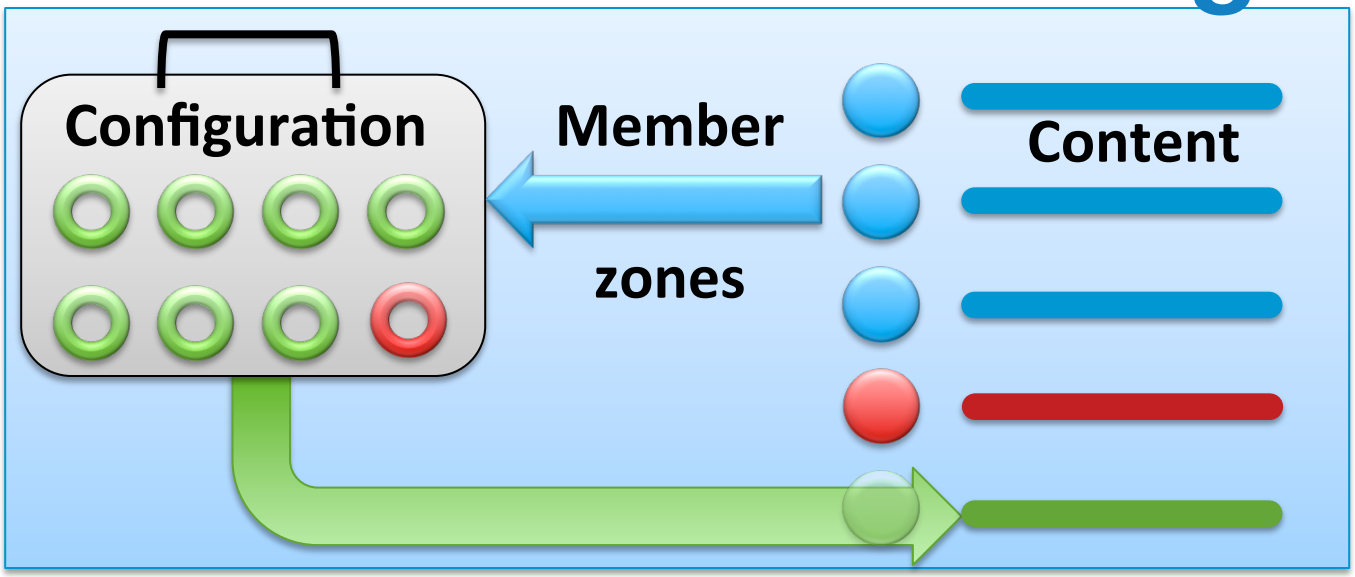


Slave or Secondary



Zone Deleted from Catalog

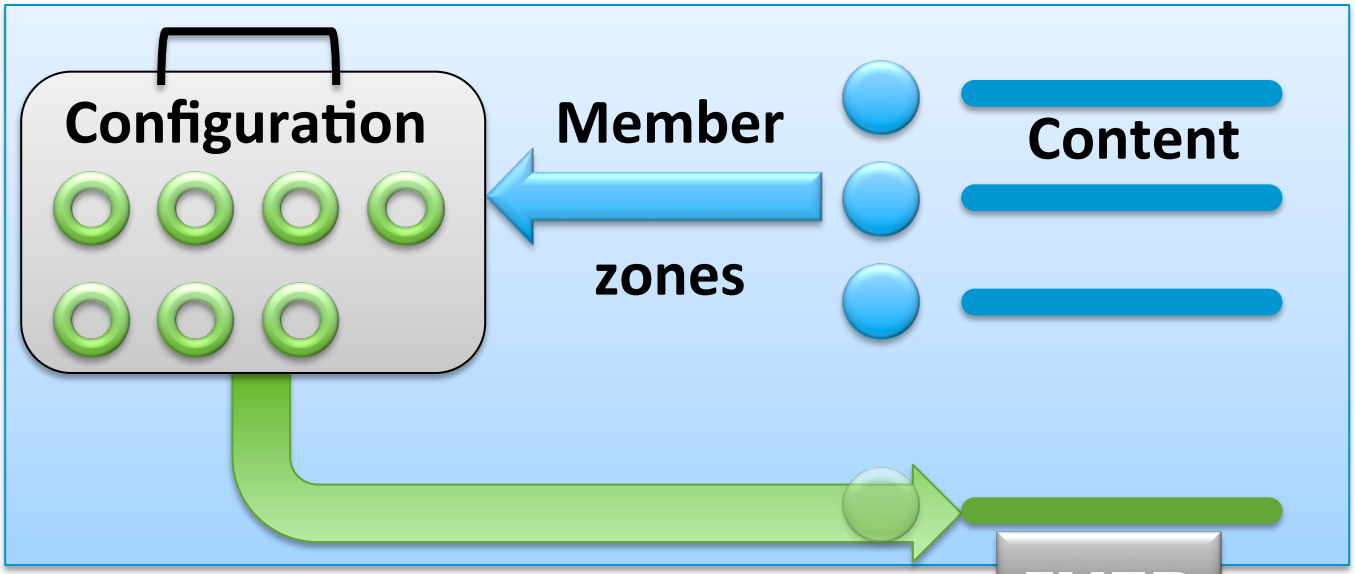
Master or Primary



Slave or Secondary

Zone Deleted from Slave

Master or Primary

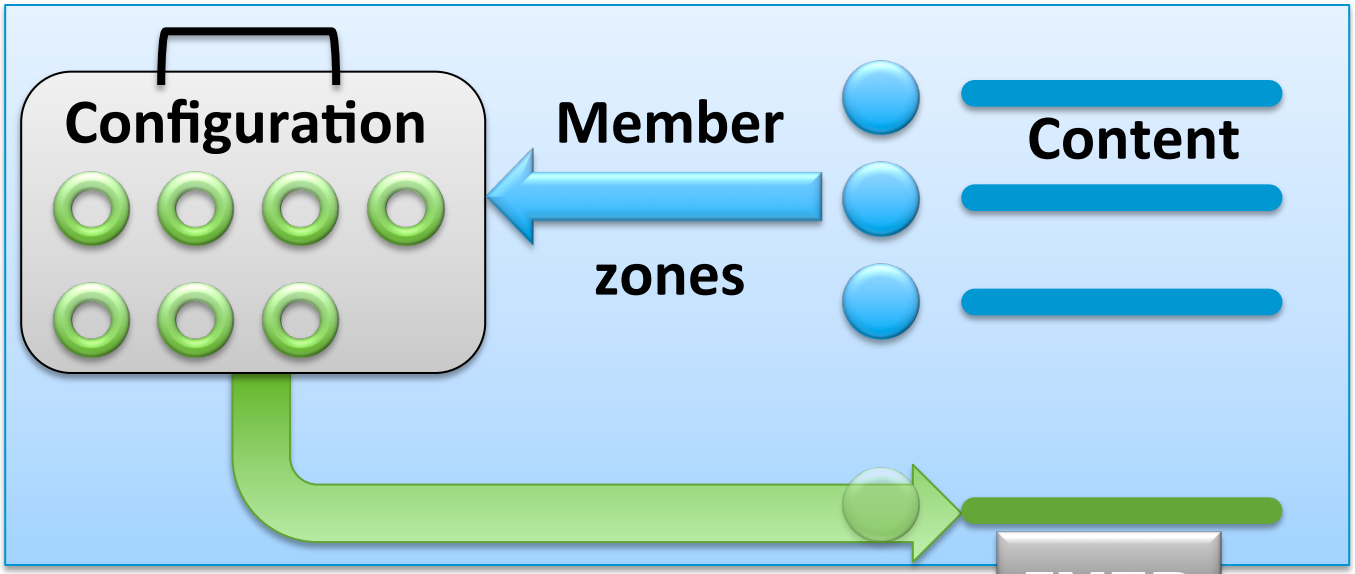


Slave or Secondary

When the catalog zone is transferred to slaves, the zone catalog on the slave is updated, removing member zones.

Zone Deleted from Slave

Master or Primary

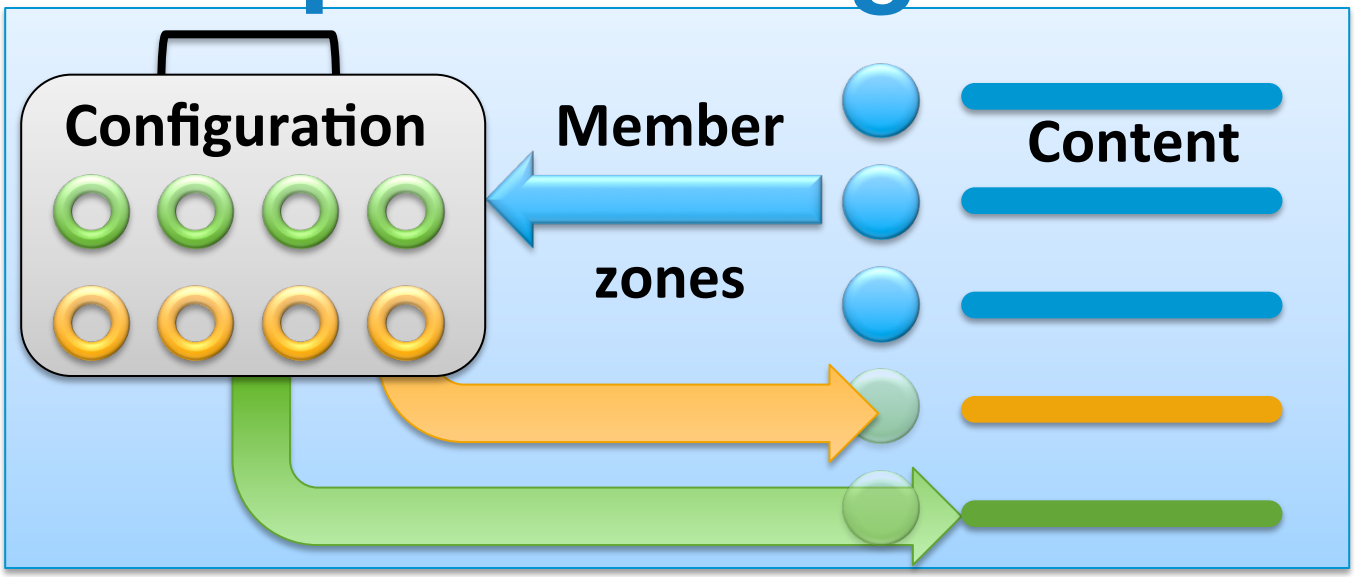


Slave or Secondary

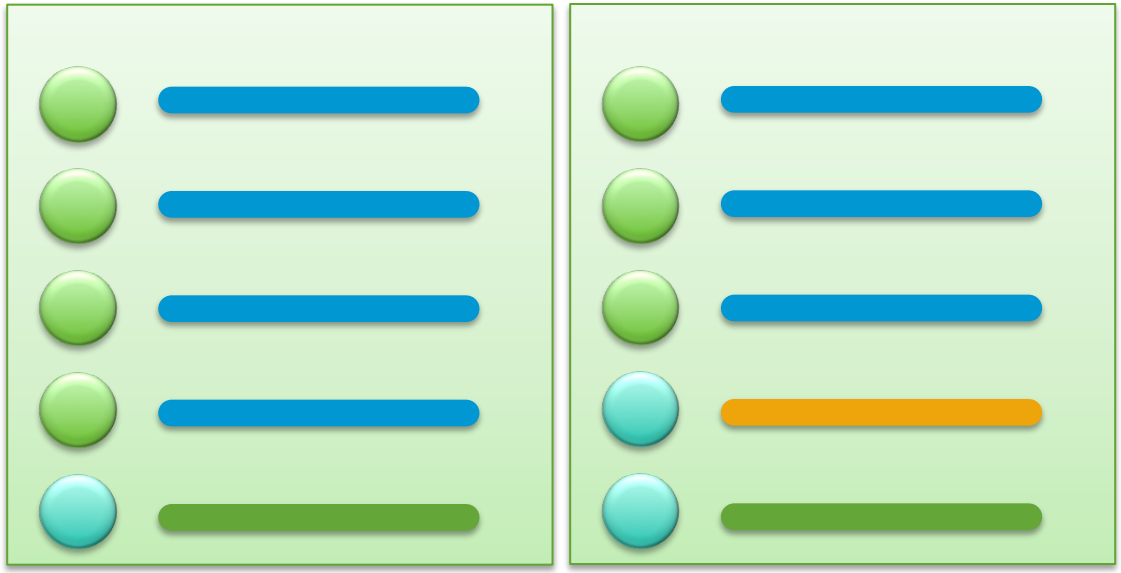
When the catalog zone is transferred to slaves, the zone catalog on the slave is updated, removing member zones.

Multiple catalogs

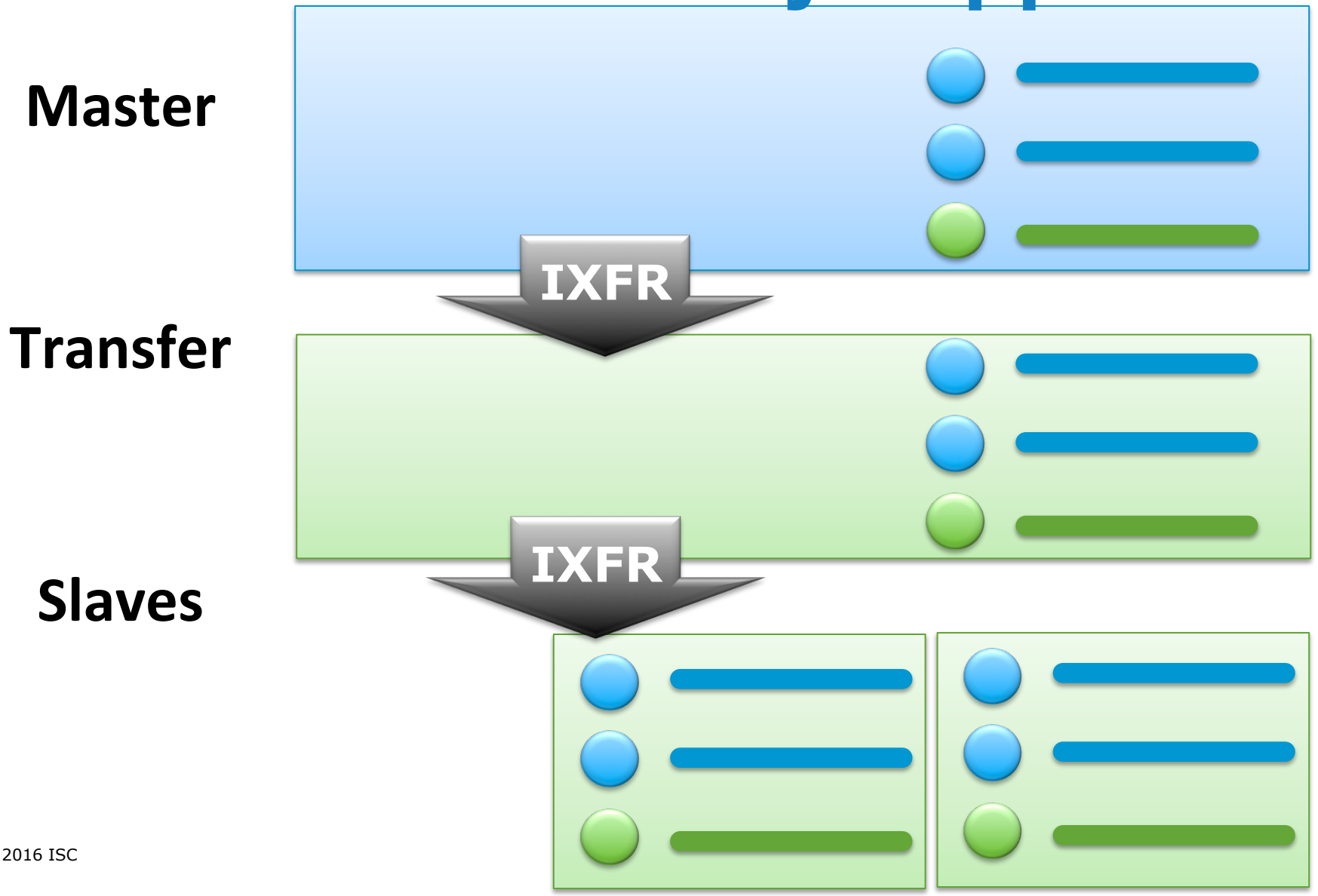
Master



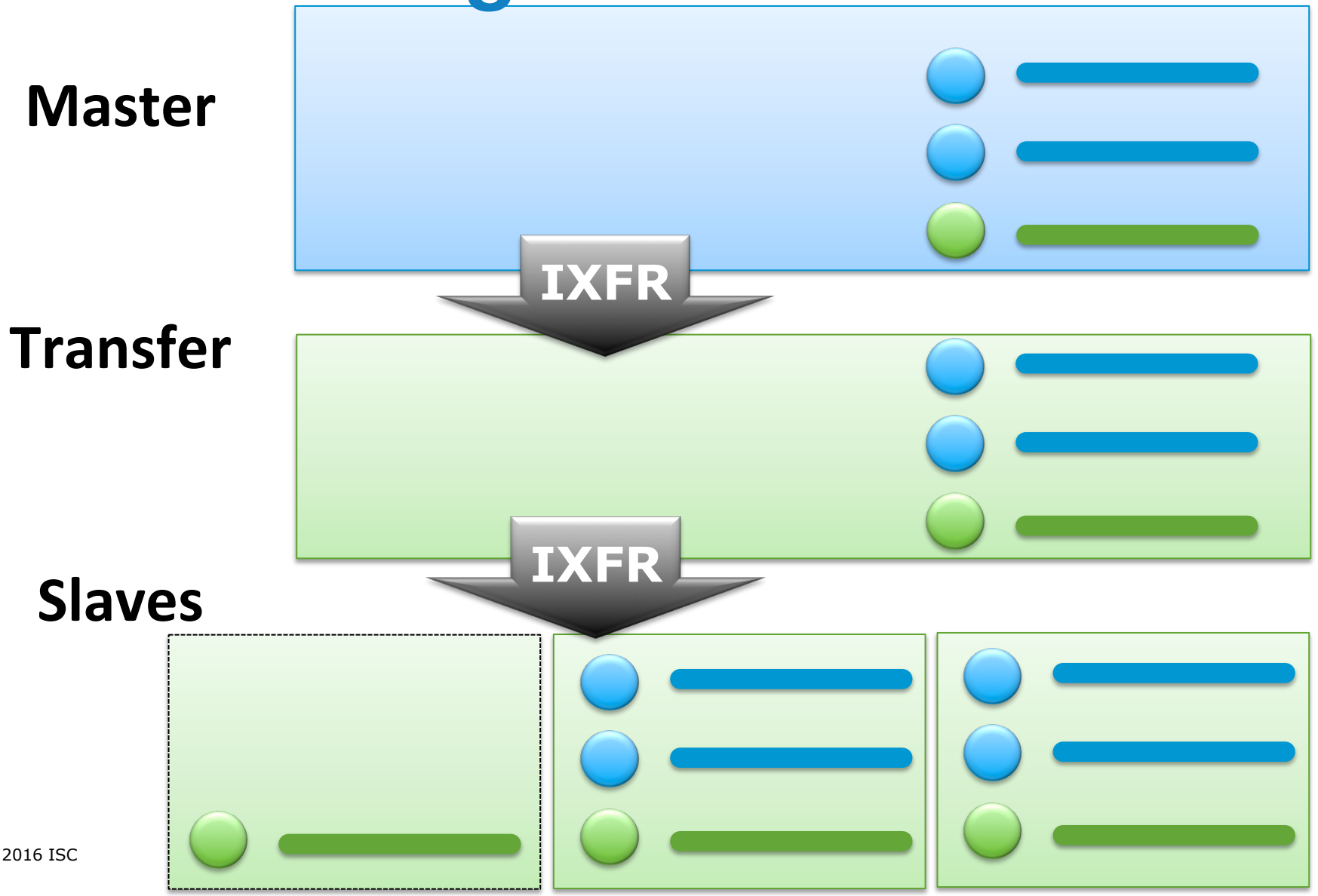
Slaves



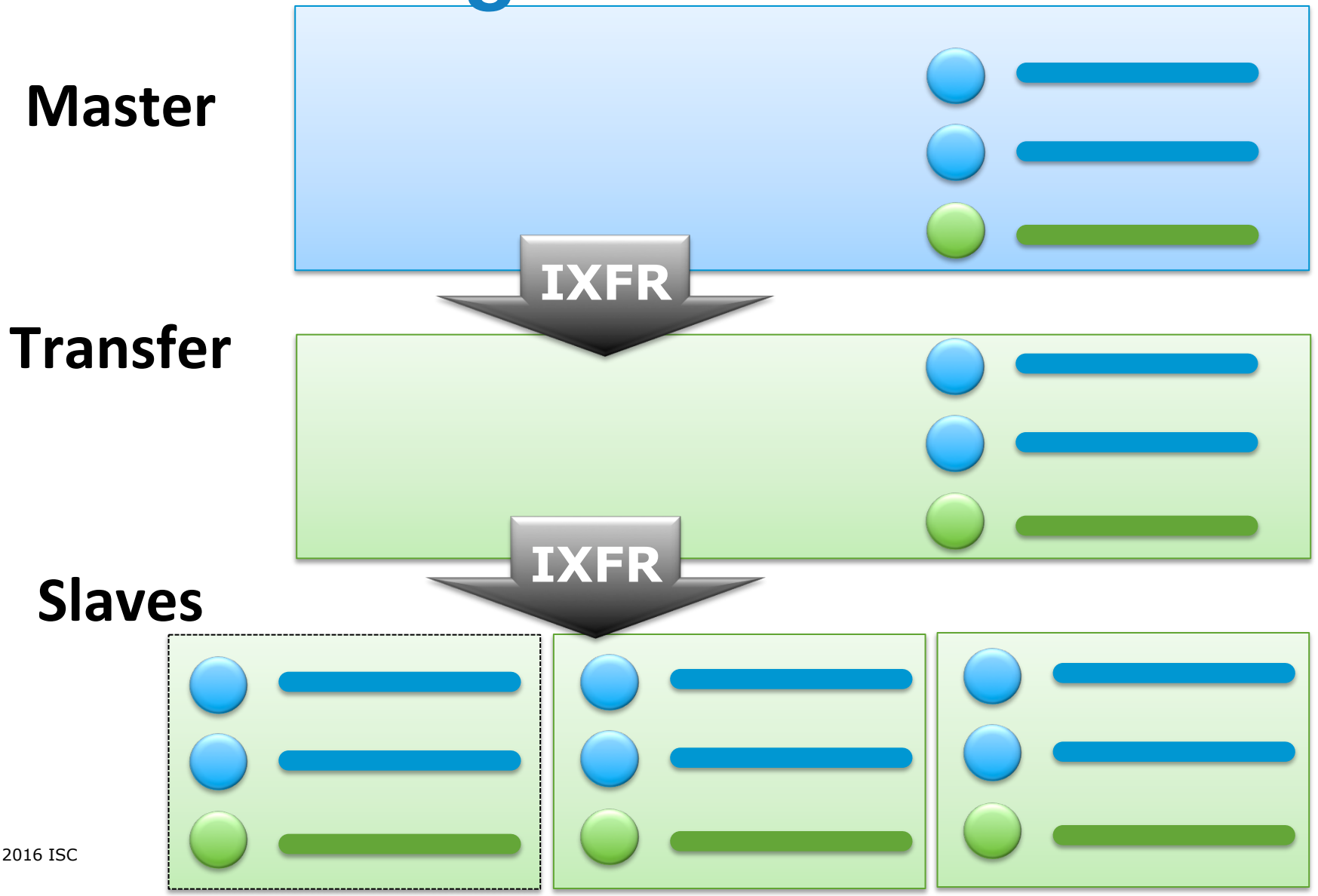
Transfer Hierarchy Supported



Adding another Slave



Adding another Slave



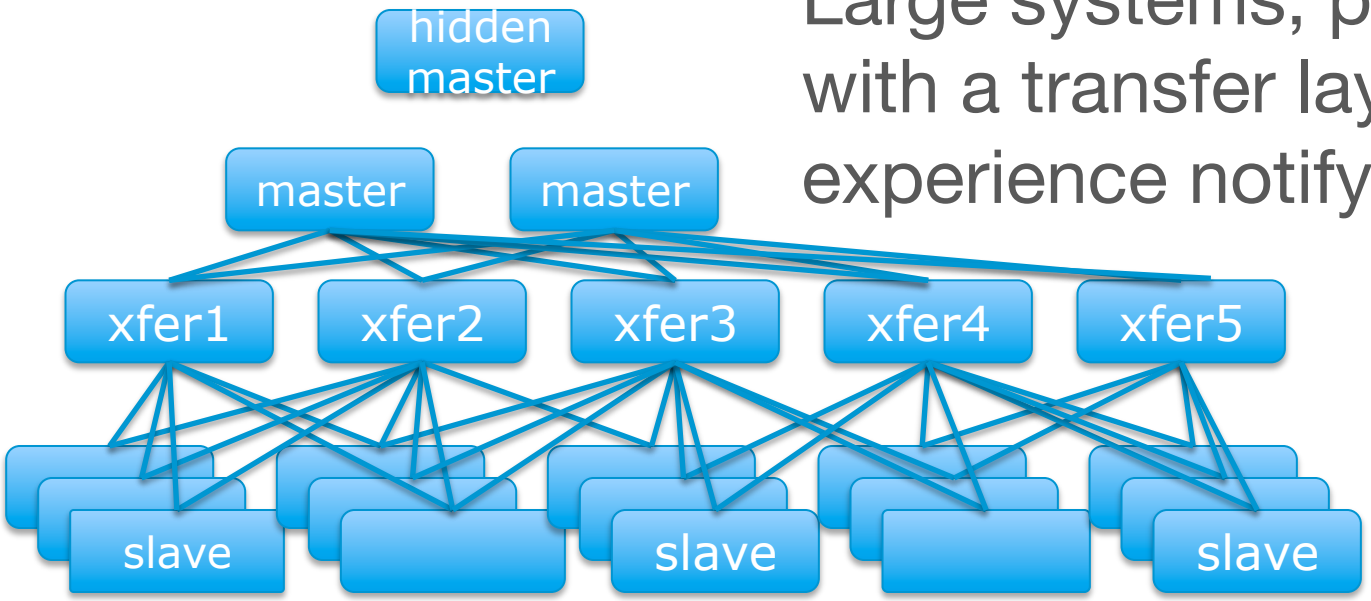
Zone options supported

- master
- allow-update
- allow-transfer
- keys
- allow-query

Provisioning challenges

- ✓ Updating zone list across a large pool of slaves
- notify traffic overhead, particularly with a multi-tiered system
- RNDC designed for human interaction, being used by scripts
- Zones added via RNDC very slow to delete

Notify Storms



Large systems, particularly with a transfer layer of servers, experience notify congestion

- Separate startup notify-rate queue
- LIFO rather than FIFO gets newly added zones updated faster

Automating with RNDC

New RNDC commands

- check if a zone exists
- show current zone configuration
- modify zone configuration
- offer read-only access for unprivileged applications

RNDC Python Module

reuse RNDC a single connection for a whole series of commands

Faster Zone Removal

- zones added via RNDC stored in a ‘new zone file’ (NZF)
- deleting entries from NZF can be 5x slower than adding a zone
- compile with Lightning Memory-Mapped Database Manager (LMDB) if this is an issue for you

Provisioning challenges

- ✓ Updating zone list across a large pool of slaves
- ✓ notify traffic overhead, particularly with a multi-tiered system
- ✓ RNDC designed for human interaction, being used by scripts
- ✓ Zones added via RNDC very slow to delete

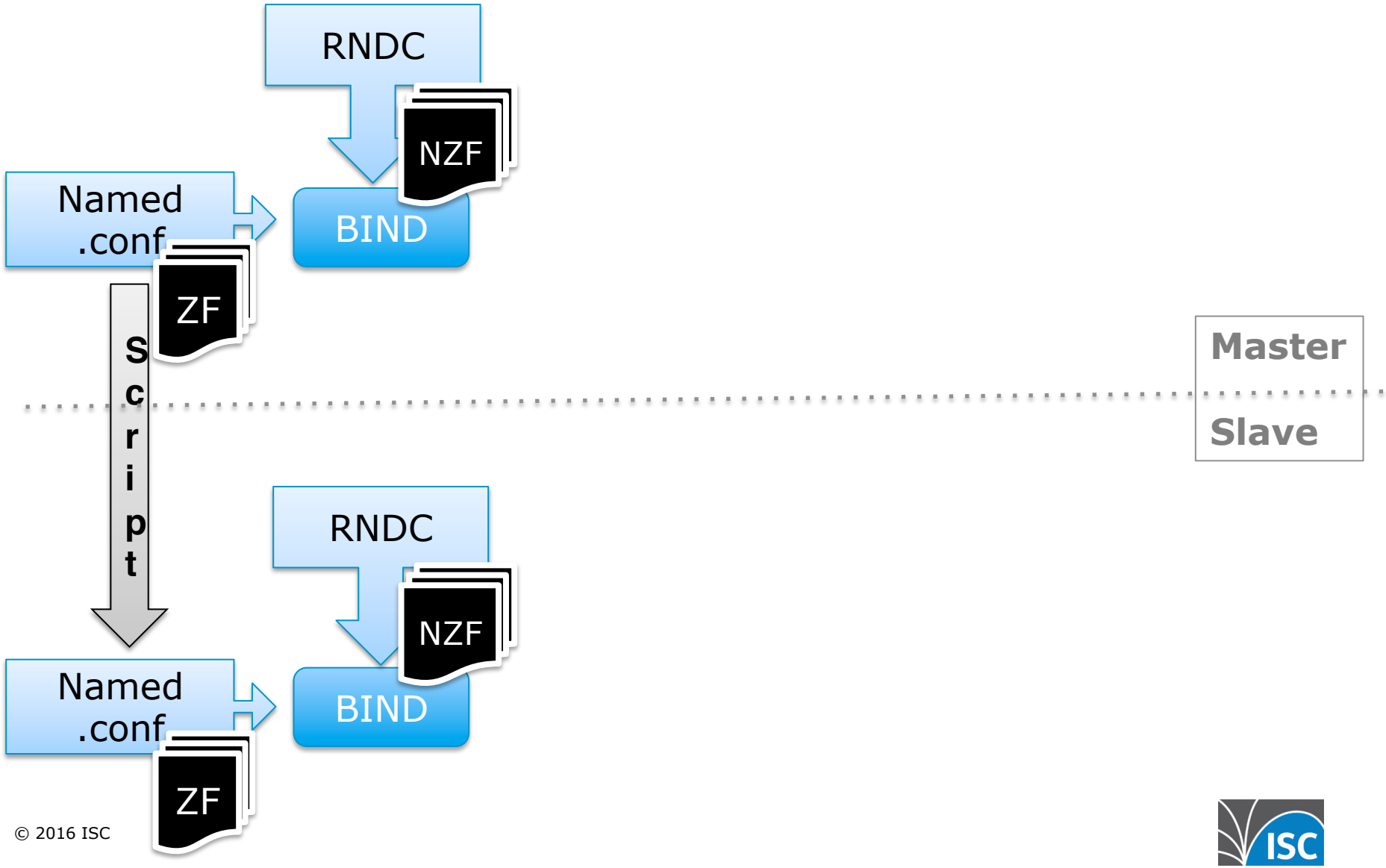
Alternative: Keep all zones in an external database

- PowerDNS does this
- OpenStack uses this model
- DLZ enables this – but zones are served *very slowly*
- Deployment choice to prefer DB tools, such as Multi-Master, to propagate information

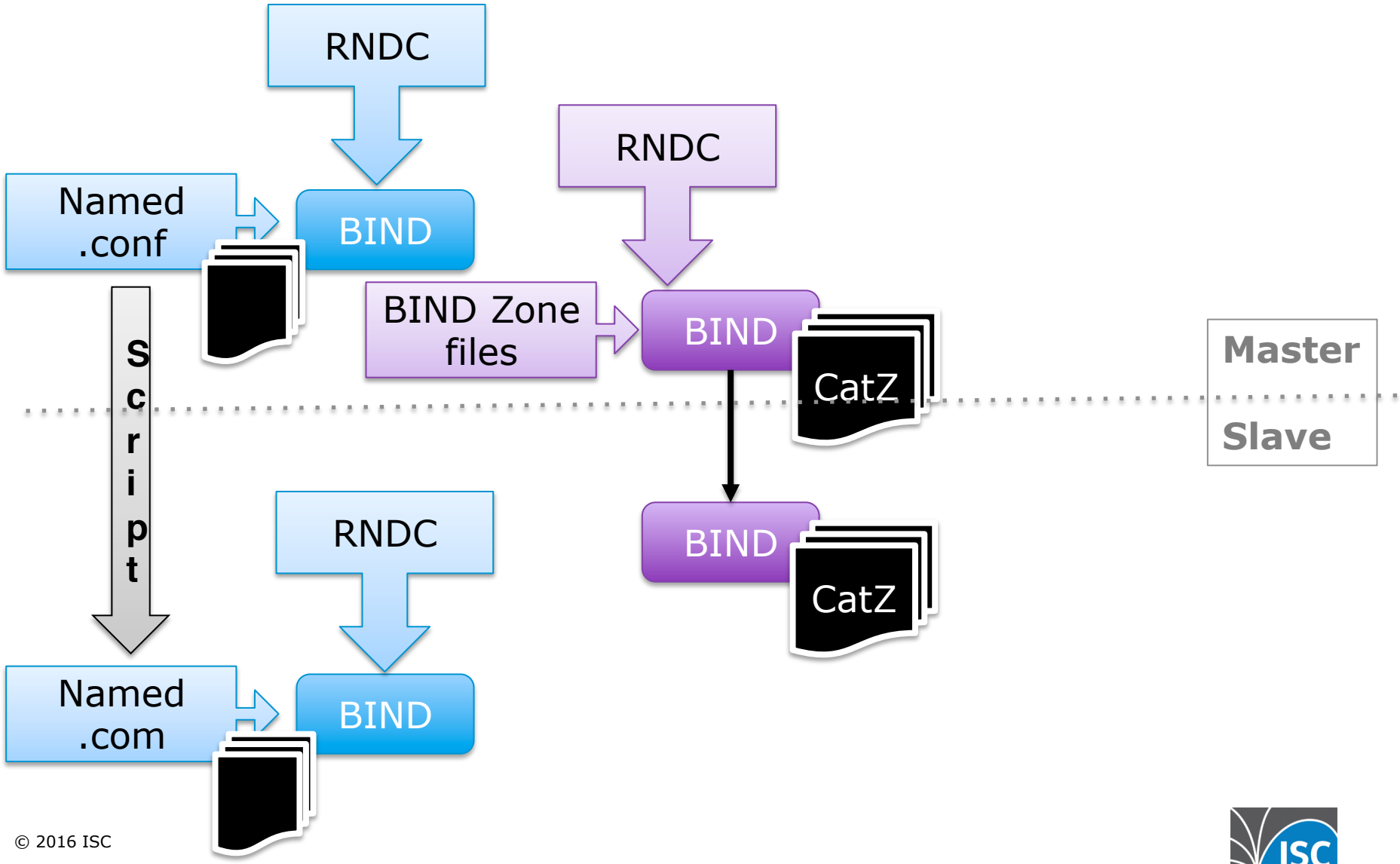
NEW - DynDB interface

- Load zone data into memory from external database
- Performance is ~ 95% of 'native' zone files!
- Works with DNSSEC
- Developed for RedHat's FreeIPA (LDAP)
  https://fedorahosted.org/bind-dyndb-ldap/
  We are hoping for contributions of other backends, such as LMDB or Cassandra

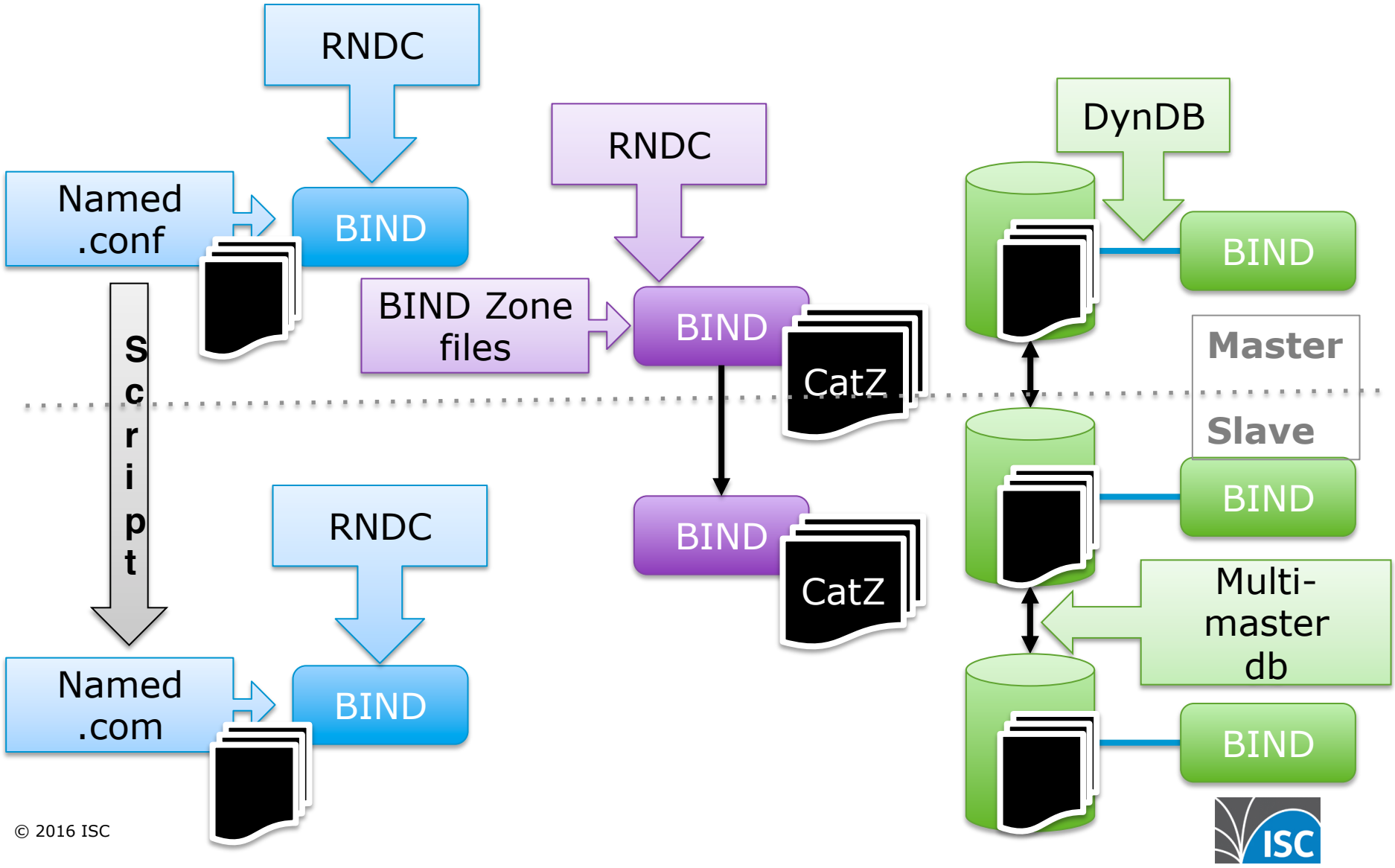
Provisioning Options



Provisioning Options



Provisioning Options



Provisioning Summary

- Traditional BIND zone files (load on restart) load faster with map format added in 9.10, now can be removed with RNDC
- New zone files (NZF) added dynamically during operation (via RNDC) - now in LMDB database
- Both of above, now propagated to slaves with Catalog Zones in 9.11
- Automated provisioning operations via RNDC
- external zone database now with no performance penalty using DynDB

New in BIND 9.11

- **Zone Provisioning improvements**
  - Catalog zones
  - RNDC updates
  - NZF w/ LMDB
  - notify rate
  - DynDB
- **DNSSEC**
  - keymgr utility
  - CDS, CDSKEY generation
  - Negative trust anchor
- **dnstap**
- **DNS Cookies**
- **minimal ANY**
- **IPv6 bias**

dnssec-keymgr

- python script intended to be scheduled in a cron job
- reads a policy definition file and creates or updates DNSSEC keys to ensure that a zone's keys match the policy for that zone.
- **New keys** are created when necessary
- If the policy changes, all **applicable keys are corrected**

Automates repetitive maintenance tasks

Policy Definition

- Policy Classes
  - different profiles for zones needing higher security
- Algorithm policies
  - e.g. default key size for a given algorithm
- Policy options
  - algorithm, TTL, ‘coverage’, key size, roll period, pre-publish, post-publish

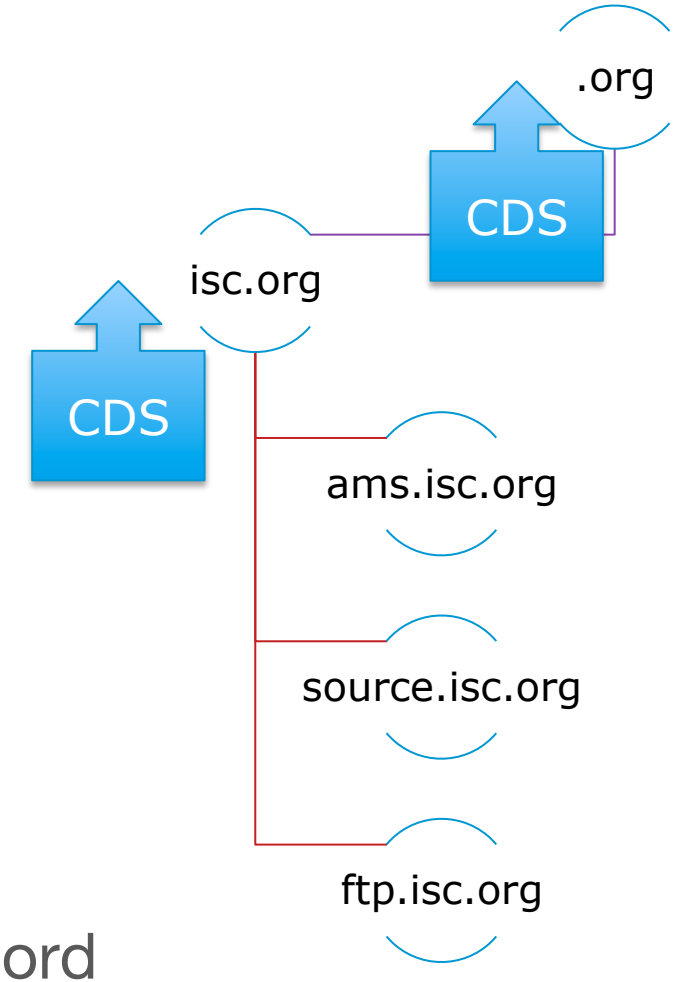
thanks to Sebastian Castro, .NZ for his help on this tool

Parent-child updating

- Unsolved problem in DNSSEC
- How to upload DNSSEC KSK data to parent zone *securely*
- Today, this is often manual, using a web portal

Parent Polls Child

- Child performs a KSK rollover
- **BIND automatically creates both CDS and CDSKEY records, signed with KSK**
  - Use DNSSEC to authenticate updates
- Parent polls for either CDS or CDSKEY
  - Some parents want to receive DNSKEYs and create the DS record
  - Other want to receive DS records



DNSSEC-blame

when a secured domain fails to validate, users blame the validating resolver

Negative Trust Anchor

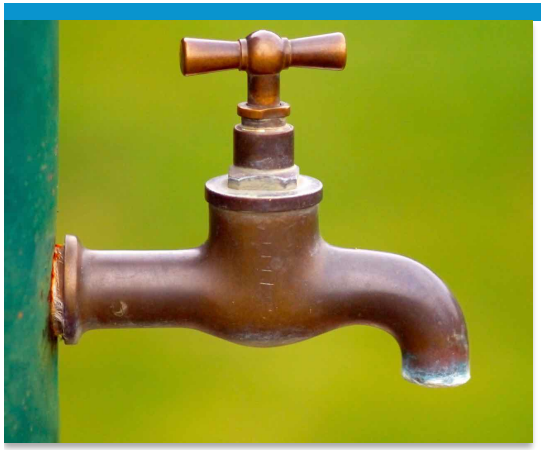
rndc nta [(-d | -f | -r | -l *duration*)] *domain* [*view*]

- Temporarily disables DNSSEC validation (1 hour, up to 1 week)
- stored in a file (*viewname*.nta) in order to persist across restarts
- nta-recheck. **named** will periodically test to see whether data below an NTA can now be validated

New in BIND 9.11

- **Zone Provisioning improvements**
  - Catalog zones
  - RNDC updates
  - NZF w/ LMDB
  - notify rate
  - DynDB
- **DNSSEC**
  - Negative trust anchor
  - keymgr utility
  - CDS, CDSKEY generation

- **dnstap**
- **DNS Cookies**
- **minimal ANY**
- **IPv6 bias**



dnstap

- flexible method for capturing and logging DNS traffic (query + response)
- more DNS intelligence than pcap
- lower overhead than BIND logging
- works across BIND, Knot and Unbound

Output: socket or file

socket

```
dnstap {auth; resolver query;} ;
dnstap-output unix "/var/run/bind/dnstap.sock”;
```

file

```
dnstap { all; } ;
dnstap-output file "/var/tmp/example.dnstap”;
```

when dnstap output is being written to a file ...

- **rndc dnstap -roll** causes dnstap output files to be rolled like log files
  - Currently (in 9.11.0), you must roll the logs as needed
- Note that dnstap is designed to drop logs rather than block operation
- New **dnstap-read** utility makes log files human-readable.

dnstap References

Thanks to Robert Edmonds, Farsight Security, Inc.

- BIND dnstap webinar posted at https://www.isc.org/mission/webinars/
- http://dnstap.info
- https://kb.isc.org/article/AA-01342/0/Using-DNSTAP-with-BIND-9.11.html

Problem: Source IP Spoofing

- BIND resolver checks the Source port, the Question, and now, also the Cookie
- Valid cookie tells us that source IP is not spoofed, so less likely to be abuse traffic

➢ *Not all abuse involves spoofed addresses (infected clients)*

Cookie Controls

- **require-server-cookie**
  - require valid cookie before sending full answer (resolver and authoritative)
- **send-cookie**
- **no-cookie-udp-size**
  - limits the size of response that will be sent without a cookie
- **cookie-secret**
  - enables cluster to share cookies

Cookies pros and cons

PROs

- Minimal overhead
- Avoid amplification
- Minimize round trips
- Cookies can be shared amongst server pools
- Easy to deploy, opportunisitic
- May eventually eliminate need for source-port randomization (!!!)
- IETF-Standardized

CONs

- Like other EDNS options, can trigger EDNS incompatibilities
- Not a ‘magic bullet’, just part of the arsenal

Cookies pros and cons

PROs

- Minimal overhead
- Avoid amplification
- Minimize round trips
- Cookies can be shared amongst servers
- Fast to deploy, opportunisitic
- May eventually eliminate need for source-port randomization (!!!)
- IETF-Standardized

CONs

- Like other EDNS options, can trigger EDNS incompatibilities
- Not a 'magic bullet', just part of the arsenal

DEFAULT=ON

DNS Cookies

no cookie, invalid cookie
minimal response

valid cookie
no rate limiting

Avoid Amplifying Responses

Request **Response**

* thanks to Tony Finch

IPv6 Bias

- Glue (in 9.9.9, 9.10.4+)
  - Prefer A for IPv4 connections
  - Prefer AAAA for IPv6 connections
- SRTT adjustment (in 9.11)
  - default value is 50 MS
  - gives IPv6 address 50 MS advantage in selection

IPv6 Bias

- Glue (in 9.9.9, 9.10.4+)
  - Prefer A for IPv4 connections
  - Prefer AAAA for IPv6 connections
- SRTT adjustment (in 9.11)
  - default value is 50 MS
  - gives IPv6 address 50 MS advantage in selection

DEFAULT=ON

Other

- many new dig options, multiple dig (mdig)
- squelch duplicate processes (help the novice!)
- relaxed PKCS#11 interface to enable pci card hsms
- TLSA record sent with MX record
- IPv6 mtu change to avoid fragmentation
- server-side support for pipelined TCP queries
- default value for the number of UDP listeners = detected processors minus one
- quantum signing size control (sig-signing-signatures *number)*

New RRTYPES

- **AVC** - Application Visibility and Control (Cisco)
- **CDS** Contains the set of DS records that should be published by the parent zone.
- **CDSKEY** Identifies which DNSKEY records should be published as DS records in the parent zone.
- **CSYNC** Child-to-Parent Synchronization in DNS as described in RFC 7477.
- **NINFO** - Zone status information
- **OPENPGPKEY**
- **RKEY -** Resource record key
- **SINK** - Kitchen Sink record
- **SMIME** - S/MIME Security Certificate (in 9.10.4)
- **TA, TALINK** – Trust Anchor, Trust Anchor link

Invisible Features

Since 9.10.0, we have added:

- continuous performance testing
- regular, on-going fuzz testing
- new, more complicated build test combinations

Performance

Performance generally DECREASES as you add features

expect a decrease in qps from 9.10 for authoritative (for few XXL zones)

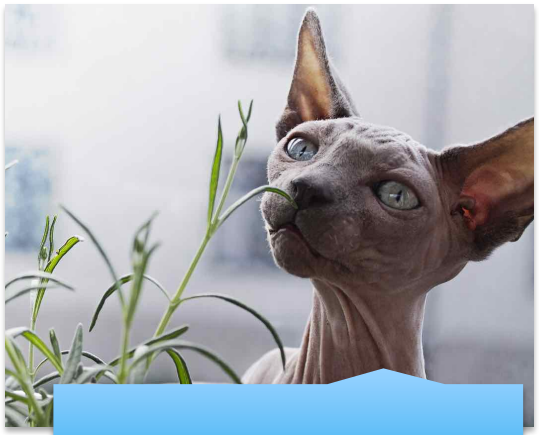
about **the same as 9.10** for large #s of small zones

Resolver operators will see an **increase** in qps (vs 9.10)

References

1. RFC 7873, Domain Name System (DNS) Cookies https://tools.ietf.org/html/rfc7873
2. RFC 7344 Automating DNSSEC Delegation Trust Maintenance https://tools.ietf.org/html/rfc7344
3. Catalog zones (draft in progress) https://datatracker.ietf.org/doc/draft-muks-dnsop-dns-catalog-zones/
4. S/MIME https://datatracker.ietf.org/doc/draft-ietf-dane-smime/
5. Using DANE to Associate OpenPGP public keys https://datatracker.ietf.org/doc/rfc7929/
6. www.dnstap.info
7. Using dnstap with BIND 9.11 https://kb.isc.org/article/AA-01342
8. A-short-introduction-to-Catalog-Zones https://kb.isc.org/article/AA-01401
9. DNS-Cookies-in-BIND-9 https://kb.isc.org/article/AA-01387

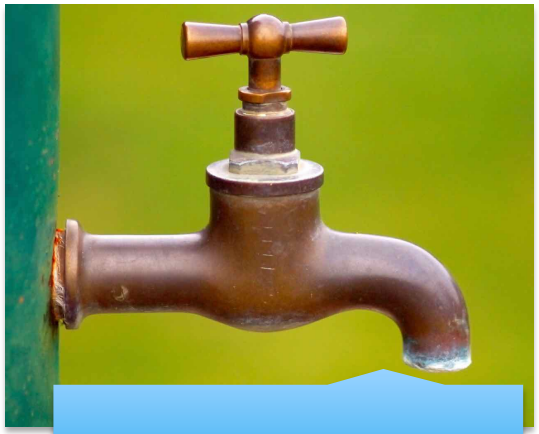
Summary: New in BIND 9.11



CATalog Zones

RNDC features

Cookies!



dnstap

DNSSEC updates

Minimal ANY