PostgreSQL Failover

How we used Consul to bridge the gap

The Goal

- Ability to failover Postgres within a data center
- Ability to failover Postgres between data centers
- Avoid false failover
- Avoid "Split Brain" situations at all costs
- Repoint Applications to new Postgres Master

Difficulty With PG HA

- Lack of quorum based solutions for master election
- Lack of functional master/master setup
- Recovery from split brain is difficult at best

The Approach

- Application Focused Failover
  - Update applications to go to the new master
- Database Focused Failover
  - Focus on promotion of new database server
  - Shoot The Other Node In The Head (STONITH)

Migrate Applications

- Manual failover (update applications)
- VIP failover (point to a VIP and migrate that)
    - Keepalived (RHEL cluster service manager)
    - Heartbeat/Pacemaker
- DNS
    - Host File, Centralized DNS, Consul

What is Consul

- Distributed Highly Available System
- Agents Report Information to Consul Servers
- Consul Uses "gossip" Protocol for Communication
- A Dynamic DNS Service for Hosts

What Does Consul Provide

- Service Discovery
- Health Checking
- Key/Value Store
- Multi-Datacenter
- Dynamic Lookups Based on Above Information
  - Service Based Lookup
  - Query Based Lookup

How We Use Consul

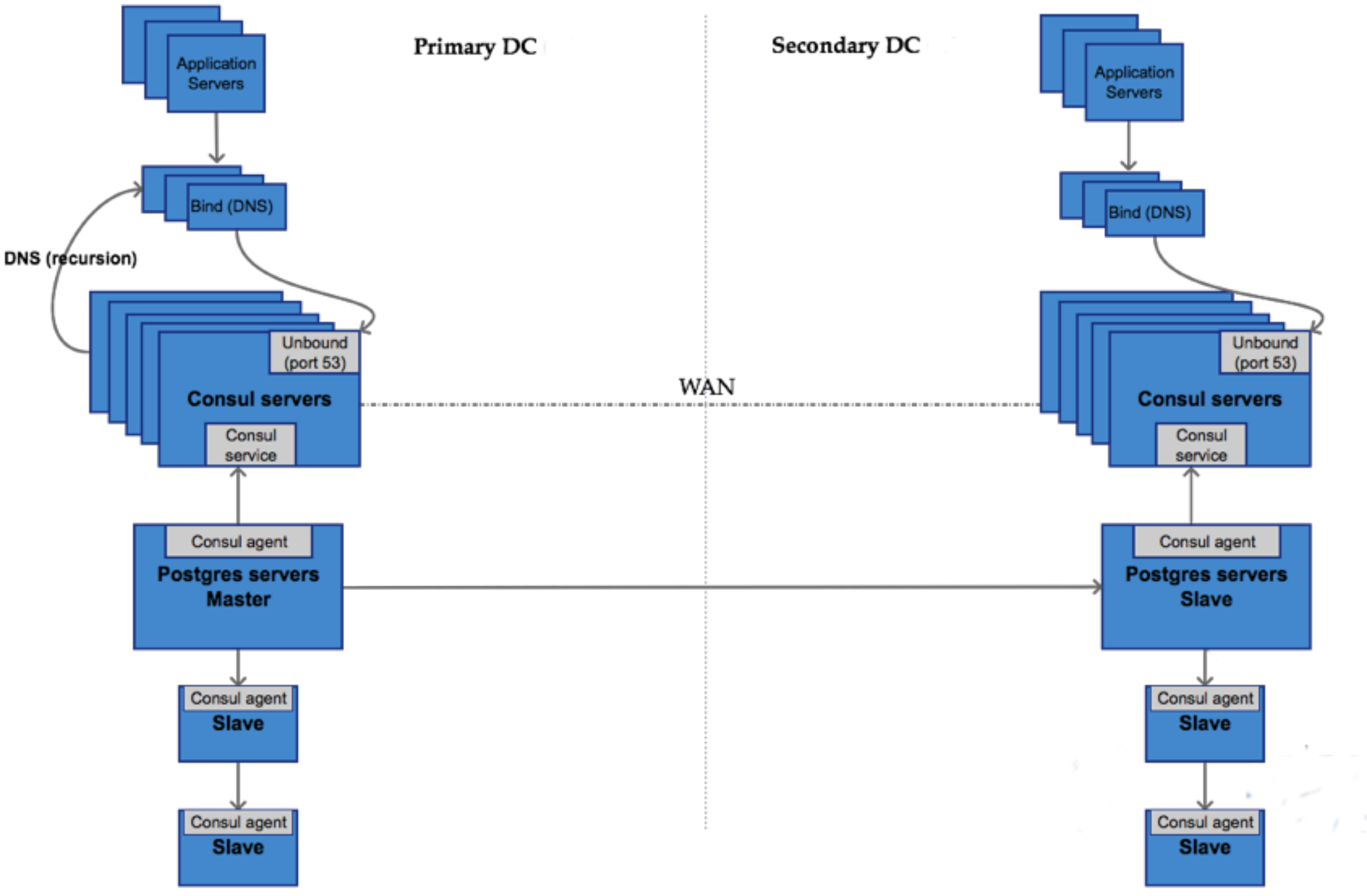
- Nodes register as master or slave of a given cluster
- Master registration requires a file to exist AND uses Consul leader election
- Applications point at a dynamic Consul query that gives results back via DNS

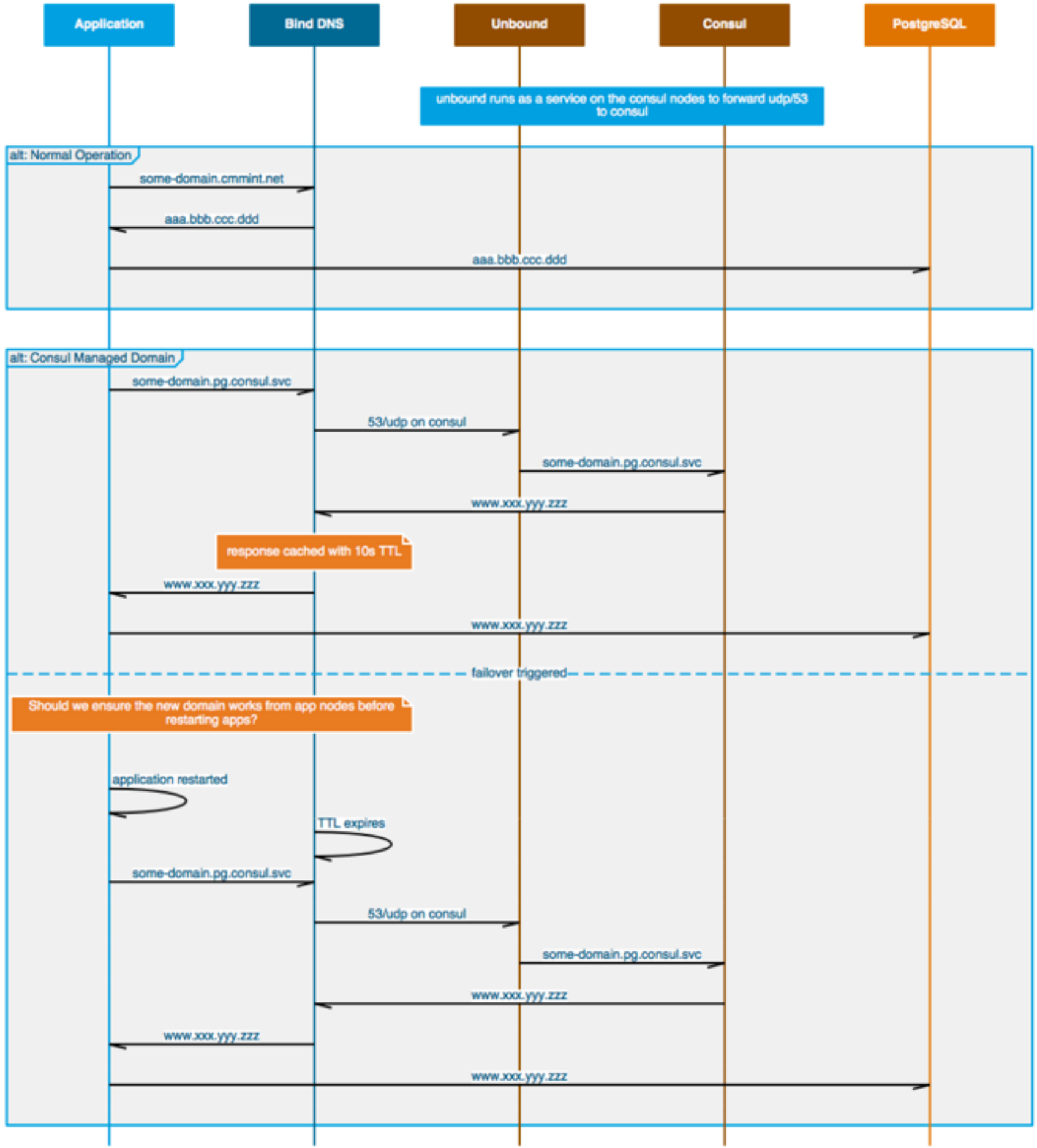
Service Based Lookup

- [tag.]<service>.service[.datacenter].<domain>
  - i.e.: master.pg.service.dc1.consul

Example of Consul Query

- pg-cluster1-master.query.production.consul
  - One result
- pg-cluster1-slave.query.production.consul
  - Multiple results in round robin result set


Primary DC
Secondary DC
Application Servers
Bind (DNS)
DNS (recursion)
Unbound (port 53)
Consul servers
Consul service
WAN
Consul agent
Postgres servers Master
Postgres servers Slave
Slave

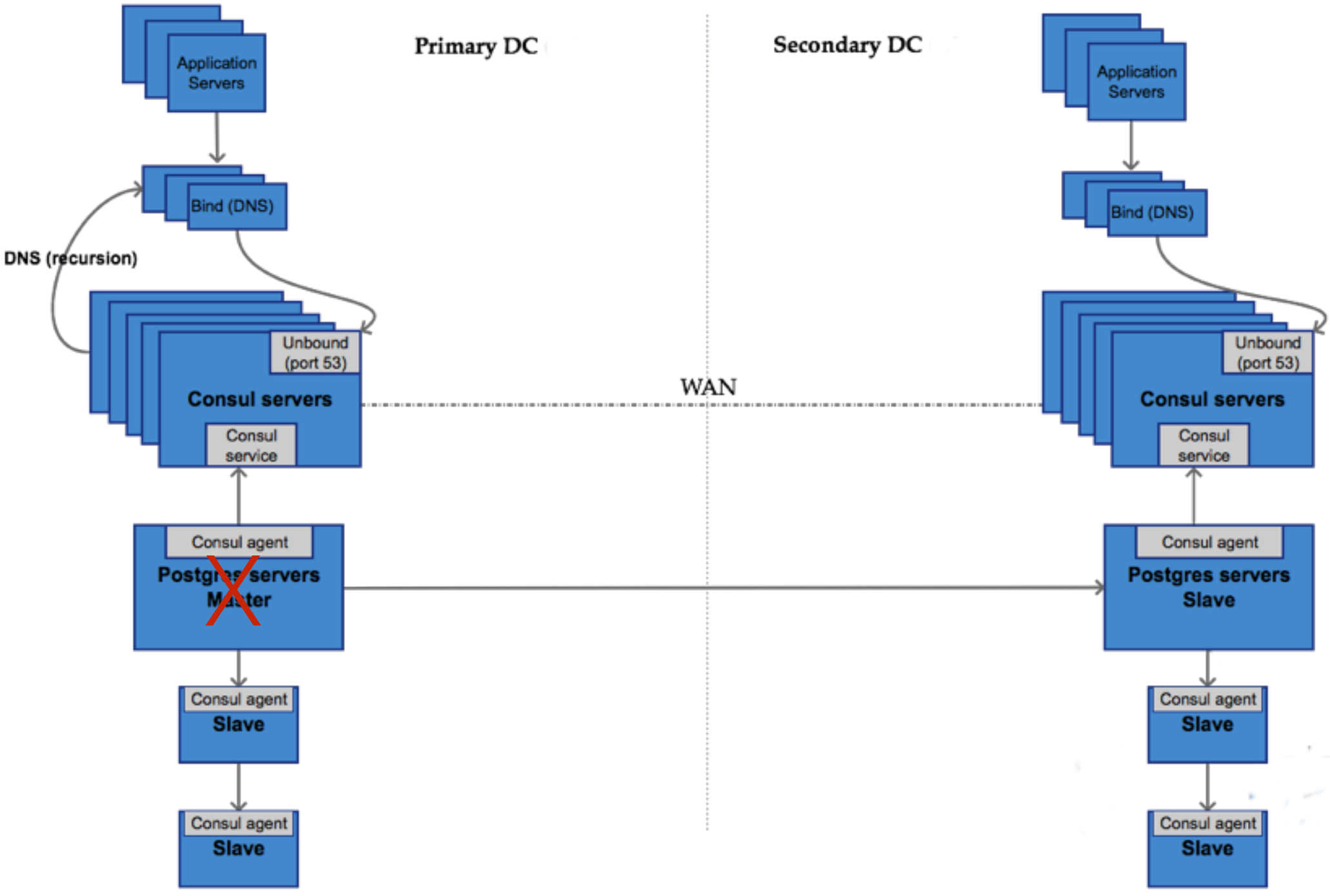

Application
Bind DNS
Unbound
Consul
PostgreSQL
unbound runs as a service on the consul nodes to forward udp/53 to consul
alt: Normal Operation
some-domain.cmmint.net
aaa.bbb.ccc.ddd
aaa.bbb.ccc.ddd
alt: Consul Managed Domain
some-domain.pg.consul.svc
53/udp on consul
some-domain.pg.consul.svc
www.xxx.yyy.zzz
response cached with 10s TTL
www.xxx.yyy.zzz
www.xxx.yyy.zzz
failover triggered
Should we ensure the new domain works from app nodes before restarting apps?
application restarted
TTL expires
some-domain.pg.consul.svc
53/udp on consul
some-domain.pg.consul.svc
www.xxx.yyy.zzz
www.xxx.yyy.zzz
www.xxx.yyy.zzz

Problems With Consul

- DNS lookups can potentially be cached by application
- Applications can pool connections which keep open sockets impacting failover
- Leader Election is restricted within datacenter

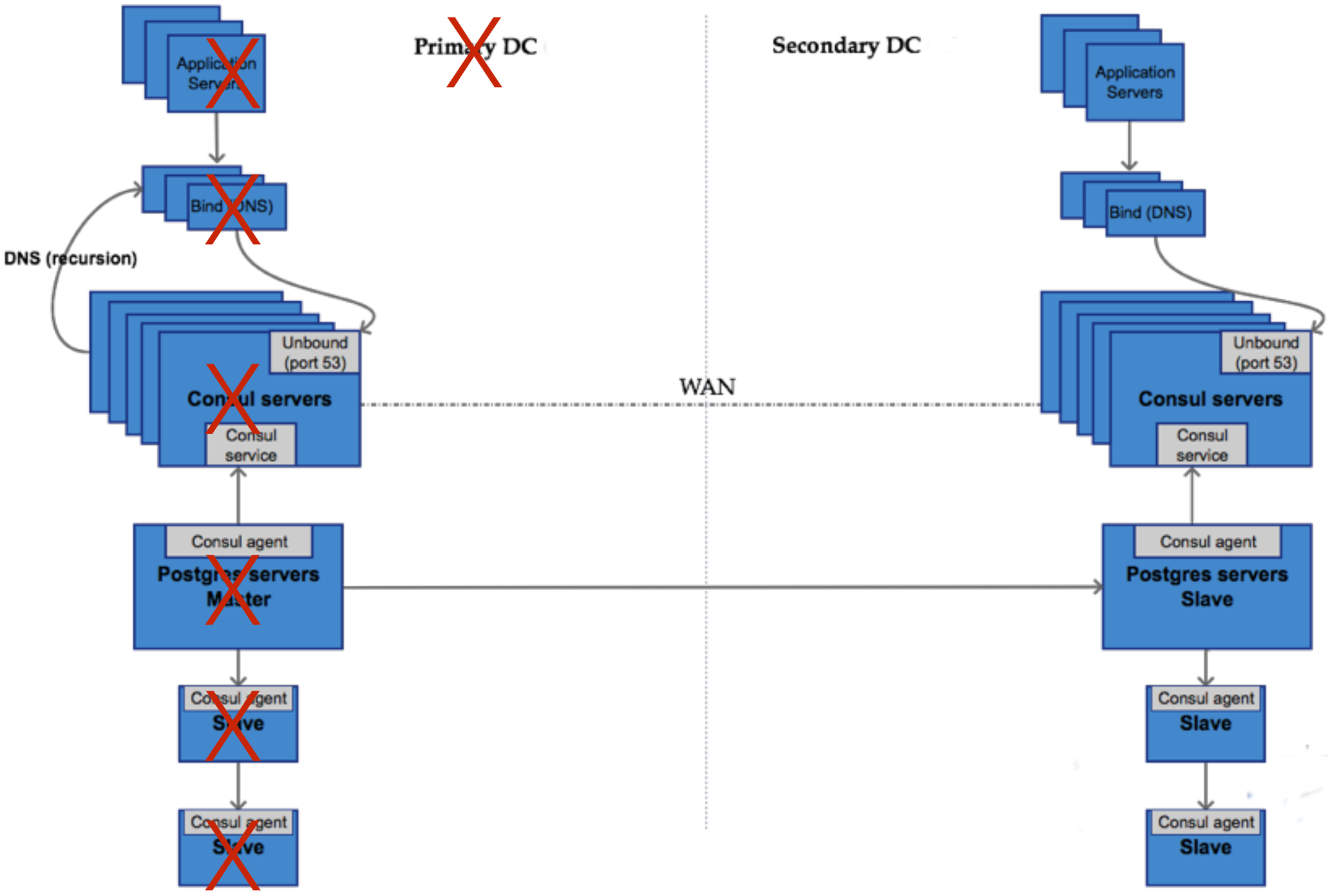
Planned Failover

- Demote master from leader election
- Change pg_hba.conf to not allow non postgres connections
- Kill all open non-postgres/replication connections
- Verify replication is caught up
- Promote slave with trigger file
- Register for leader election
- Re-point other data center primary slave and fix old master


Primary DC
Secondary DC
Application Servers
Bind (DNS)
DNS (recursion)
Unbound (port 53)
Consul servers
Consul service
WAN
Consul agent
Postgres servers Master
Consul agent
Slave
Consul agent
Slave
Application Servers
Bind (DNS)
Unbound (port 53)
Consul servers
Consul service
Consul agent
Postgres servers Slave
Consul agent
Slave
Consul agent
Slave

Unplanned Failover (within datacenter)

- Ensure that master leader election is timed out
- Promote slave with trigger file
- Register for leader election
- Re-point other data center primary slave and fix old master


Primary DC
Secondary DC
Application Servers
Bind (DNS)
DNS (recursion)
Unbound (port 53)
Consul servers
Consul service
WAN
Consul agent
Postgres servers Master
Slave
Application Servers
Bind (DNS)
Unbound (port 53)
Consul servers
Consul service
Consul agent
Postgres servers Slave
Consul agent
Slave
Consul agent
Slave

Managing Failover At Scale

- Micro-service architecture scales horizontally
- Many different database clusters
- Thousands of different application servers
- A scripted solution, but how do you coordinate????


YO DAWG, I HEARD YOU
MANGE YOUR PUPPET
WITH ANSIBLE