Postgresql High Availability.

Christophe Pettus
PostgreSQL Experts, Inc.

FOSDEM PGDay 2016

What do we want?

High Availability!

When do we want it?

FATAL: the database system is starting up

What is "high availability"?

Technologies and processes intended to minimize time the database system is not able to execute the application's workload.

This means…

- … high uptime.
- … fast recovery or provisioning of replacement server.
- … rerouting the application (if required) to the new server.
- In short, minimal time in which there "is no database."

We'll focus on…

- … solutions that use streaming replication to maintain a failover system.
  - A couple of exceptions.
- Logical replication (slony, bucardo, londiste, pg_logical) has many virtues, but is not primarily a high availability tool.

What do we want?

- Automatic promotion
- Reprovisions failed servers
- Single endpoint
- Load balancing
- Environment agnostic
- Any number of secondaries
- Connection pooling
- Open source

So, what does this?


Nothing.

No current solution does it all.

- Everything has trade-offs.
- You get to decide based on:
  - Your environment.
  - Your requirements.
  - Your patience with scripting.

We'll talk about…

- Shared storage
- Bare streaming replication
- HAProxy
- pgpool2
- pg_shard
- Heroku
- Amazon RDS
- Patroni
- Stolon

You forgot "x"!

- Yes, I did. Oh, well!
- This set is representative of what's out there right now.
- Others are really not "high availability" solutions, but more for sysadmin convenience.
  - Not that there's anything wrong with that.

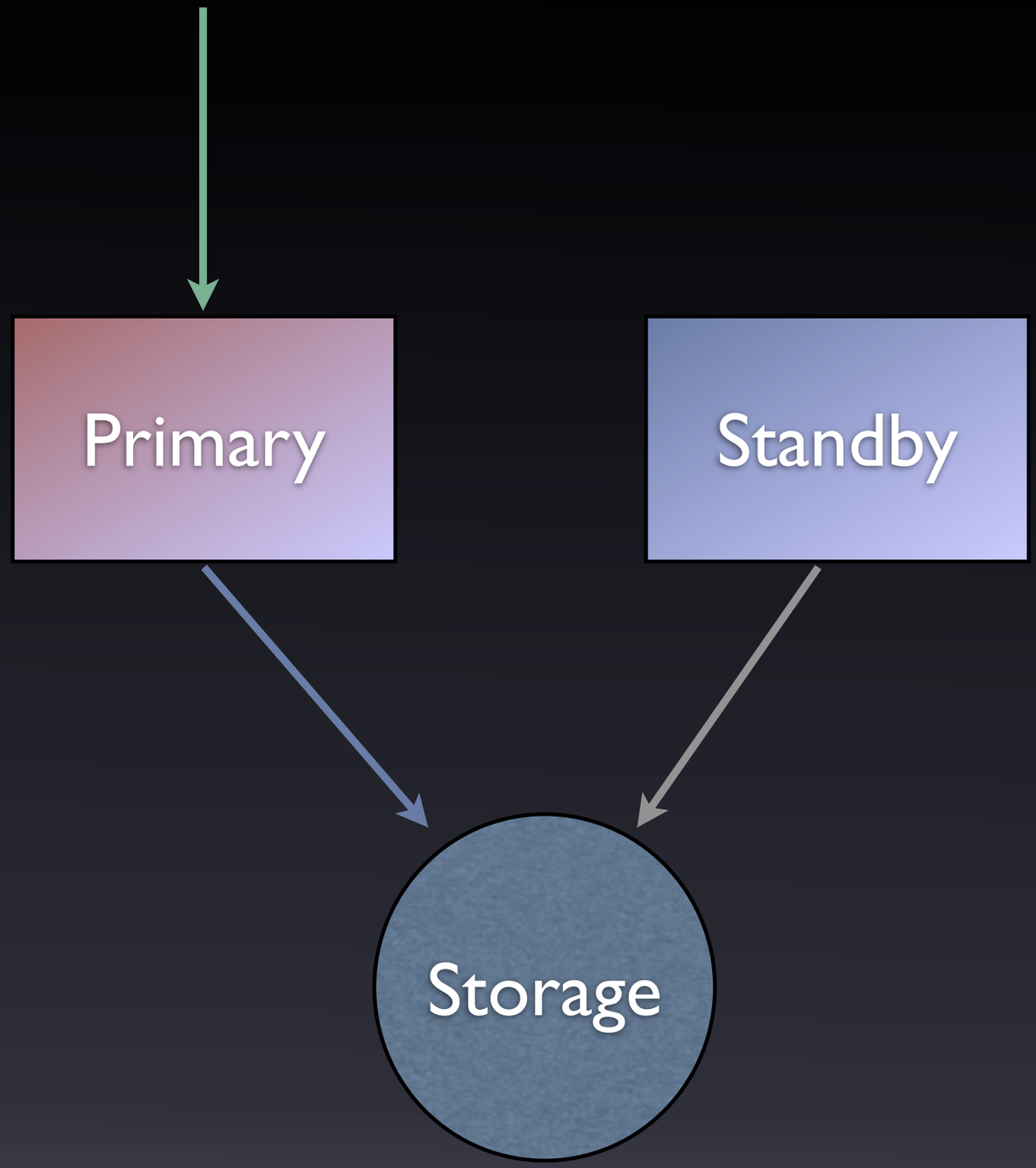
You can write Facebook in PHP.

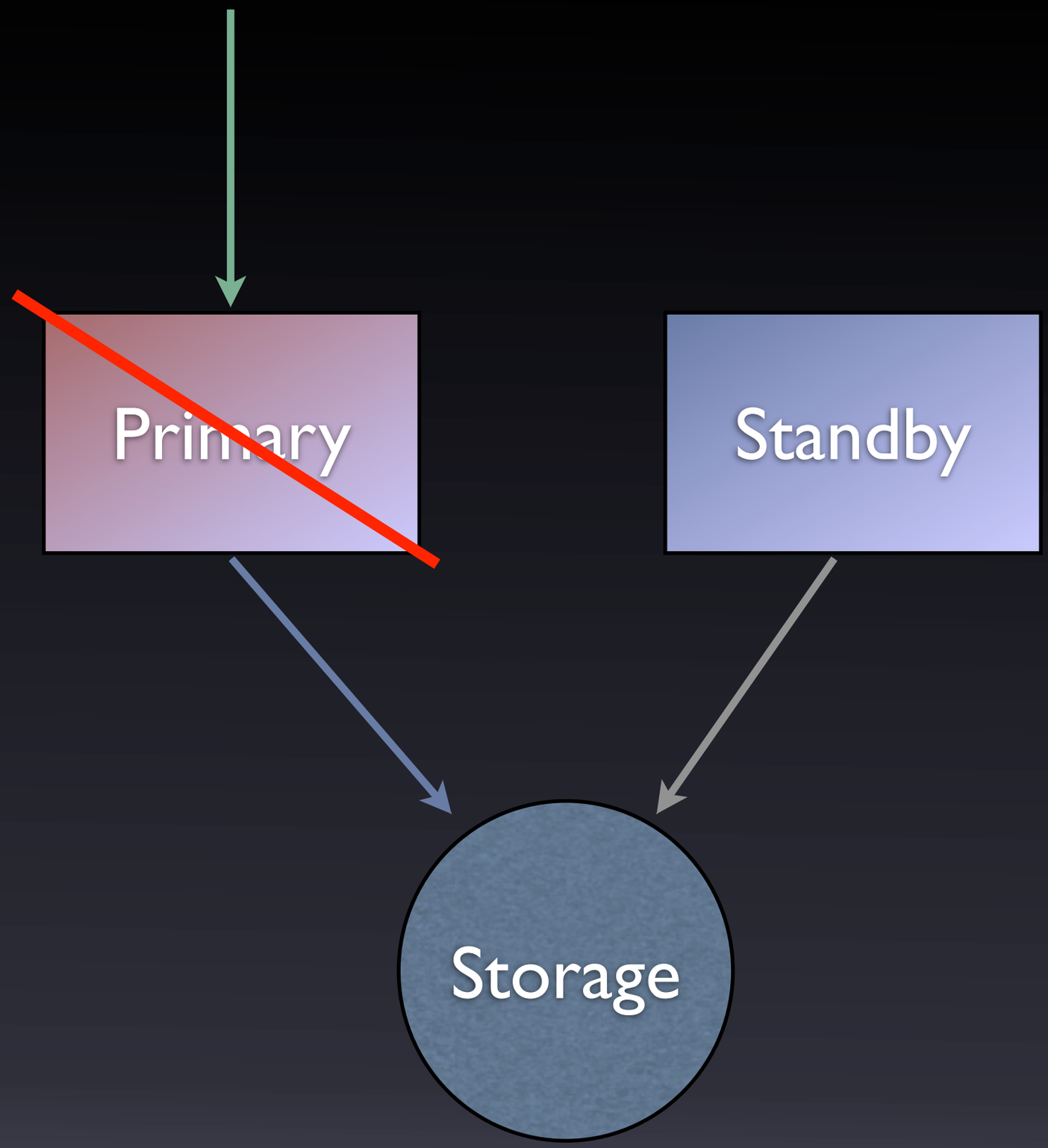
- Many of these solutions can be scripted to have more complex and advanced functionality.
- Focus here is on out-of-the-box functionality.

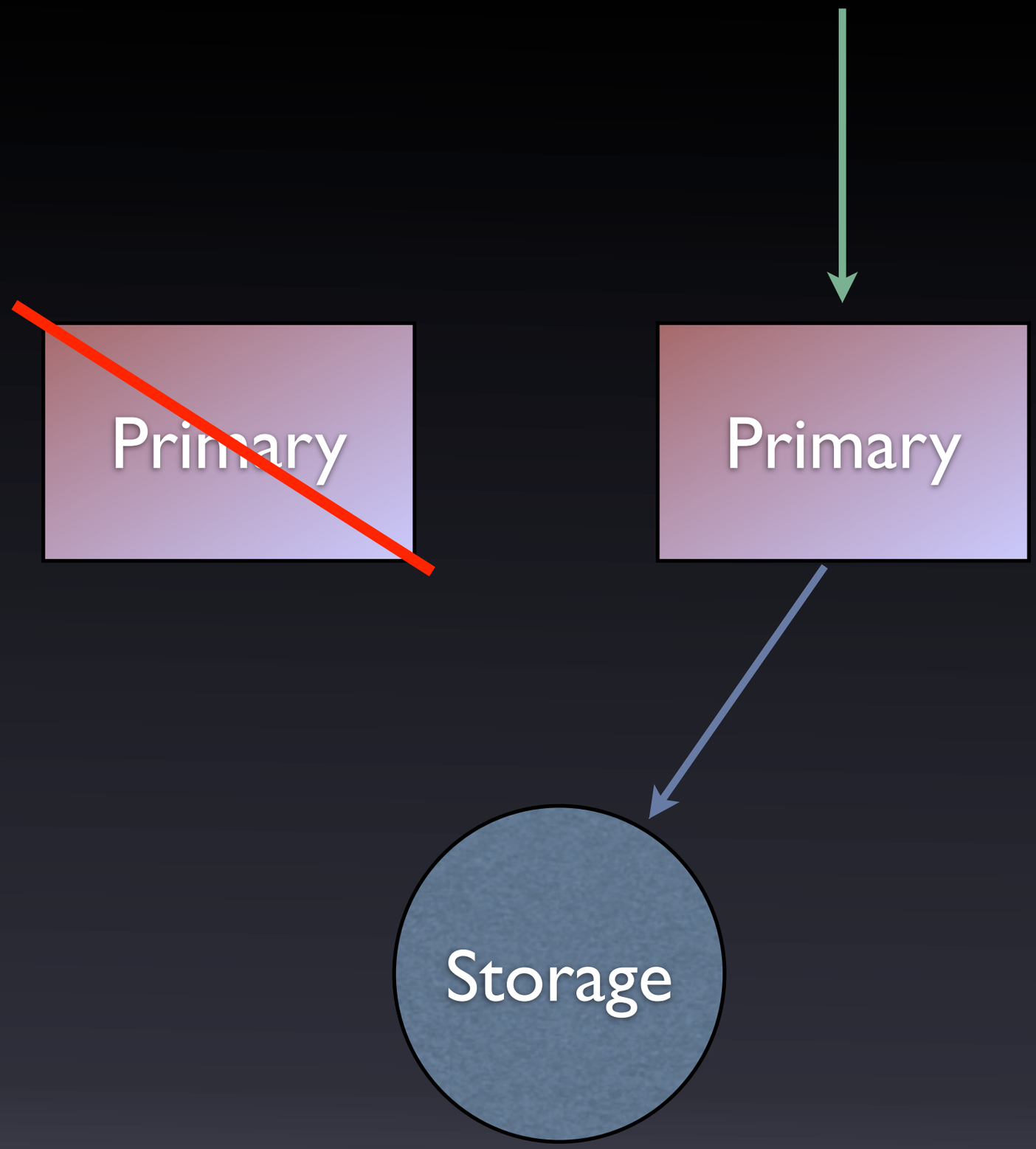
The Options.

Shared Storage

- Database volume is shared at the disk block or file system level.
  - DRBD, NFS, SAN, etc., etc.
- A standby machine (configured, not active) is waiting to come up on primary failure.
- Applications rerouted via VIP or manually.







It does:

- Single endpoint *
- Any number of secondaries
- Open source

It doesn't:

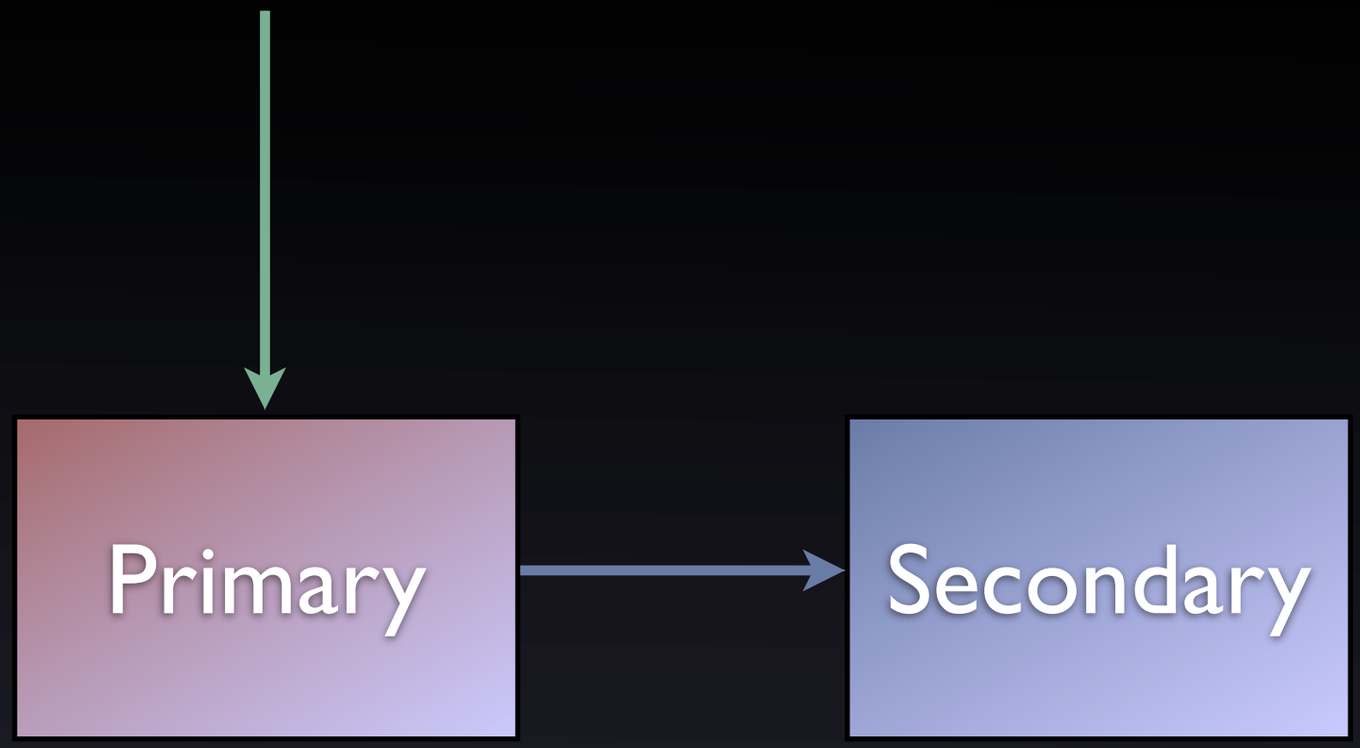
- Automatic promotion
- Reprovisions failed servers
- Load balancing
- Environment agnostic
- Connection pooling

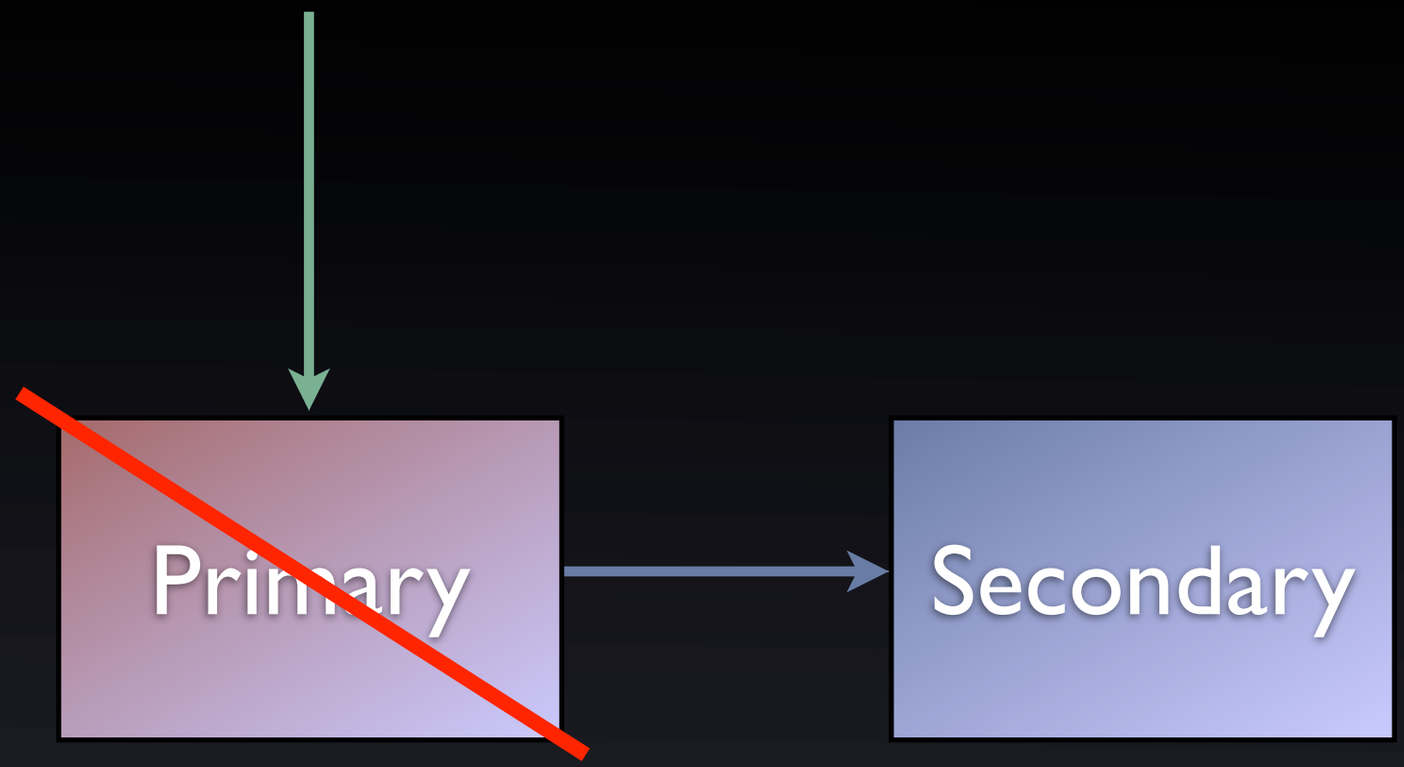
Notes.

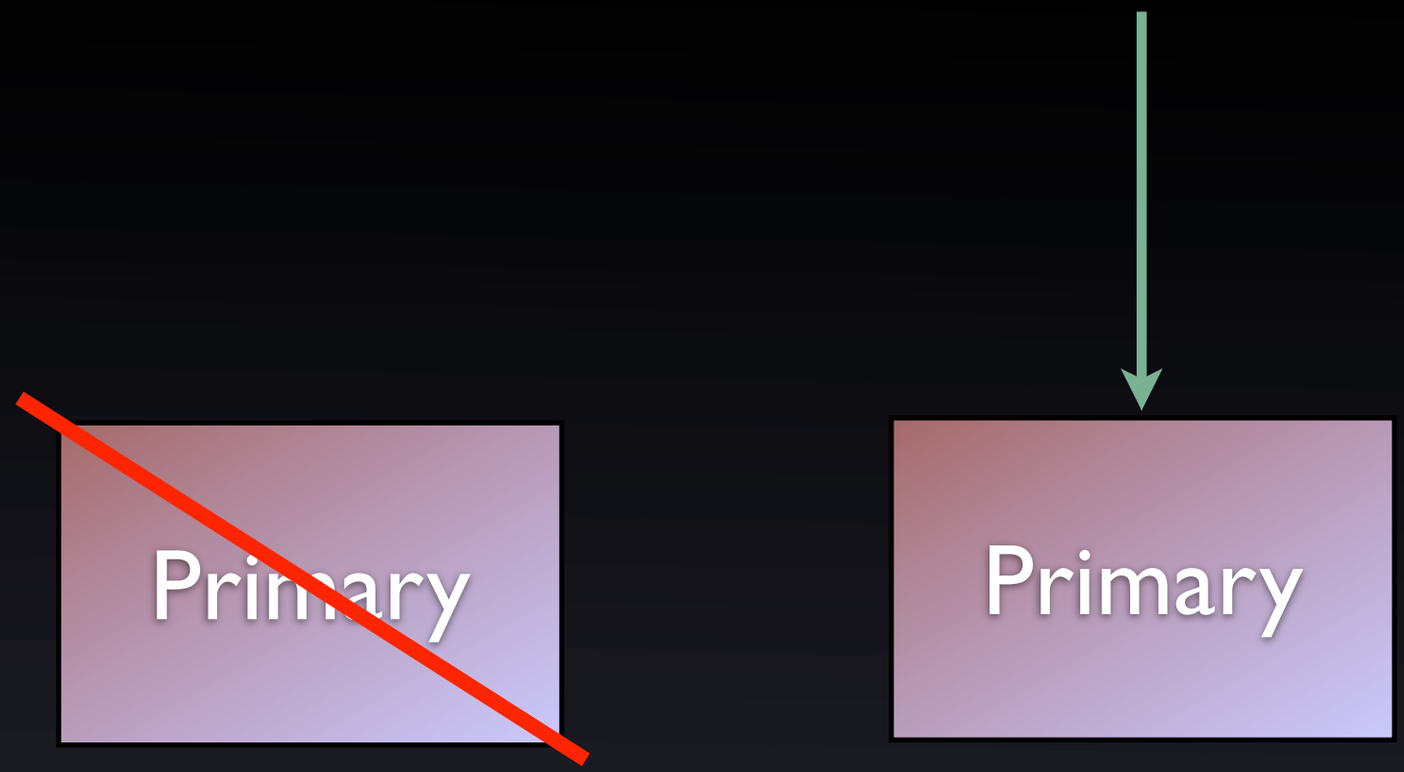
- Significant performance overhead.
- fsync-non-compliance danger.
- “No” risk of losing a committed transaction.
- Master failures can destroy shared storage, so that’s bad.

Bare streaming replication.

- Primary server takes all write traffic.
- Secondary server might handle load balancing, or just run as a standby.
- On failure, manual promotion of secondary, manual rerouting of application (or VIP), manual...
- ... well, you get the idea.


Primary
Secondary


Primary
Secondary


Primary
Primary

It does:

- Single endpoint *
- Environment agnostic
- Any number of secondaries
- Open source

It doesn't.

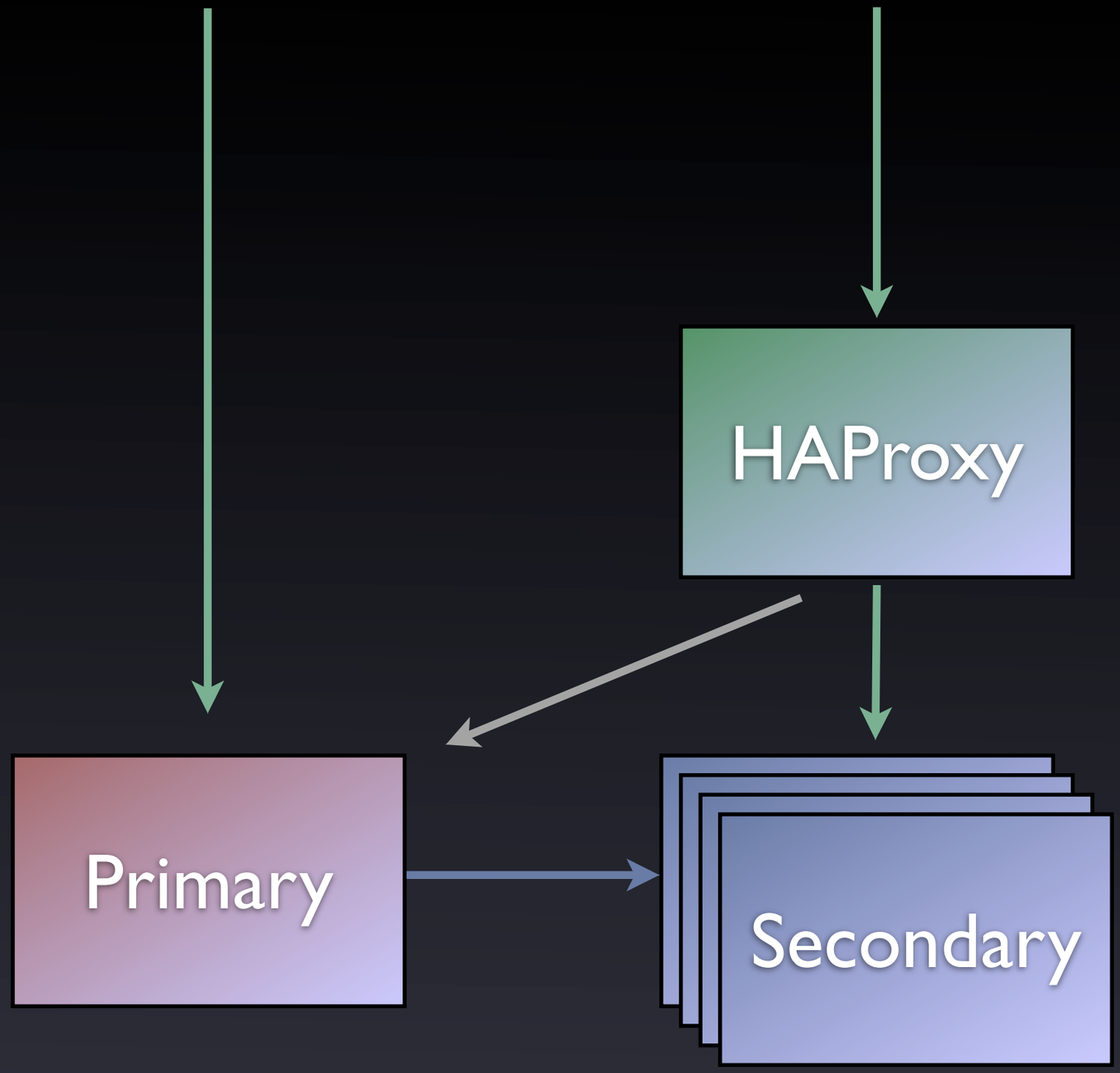
- Automatic promotion
- Reprovisions failed servers
- Load balancing
- Connection pooling

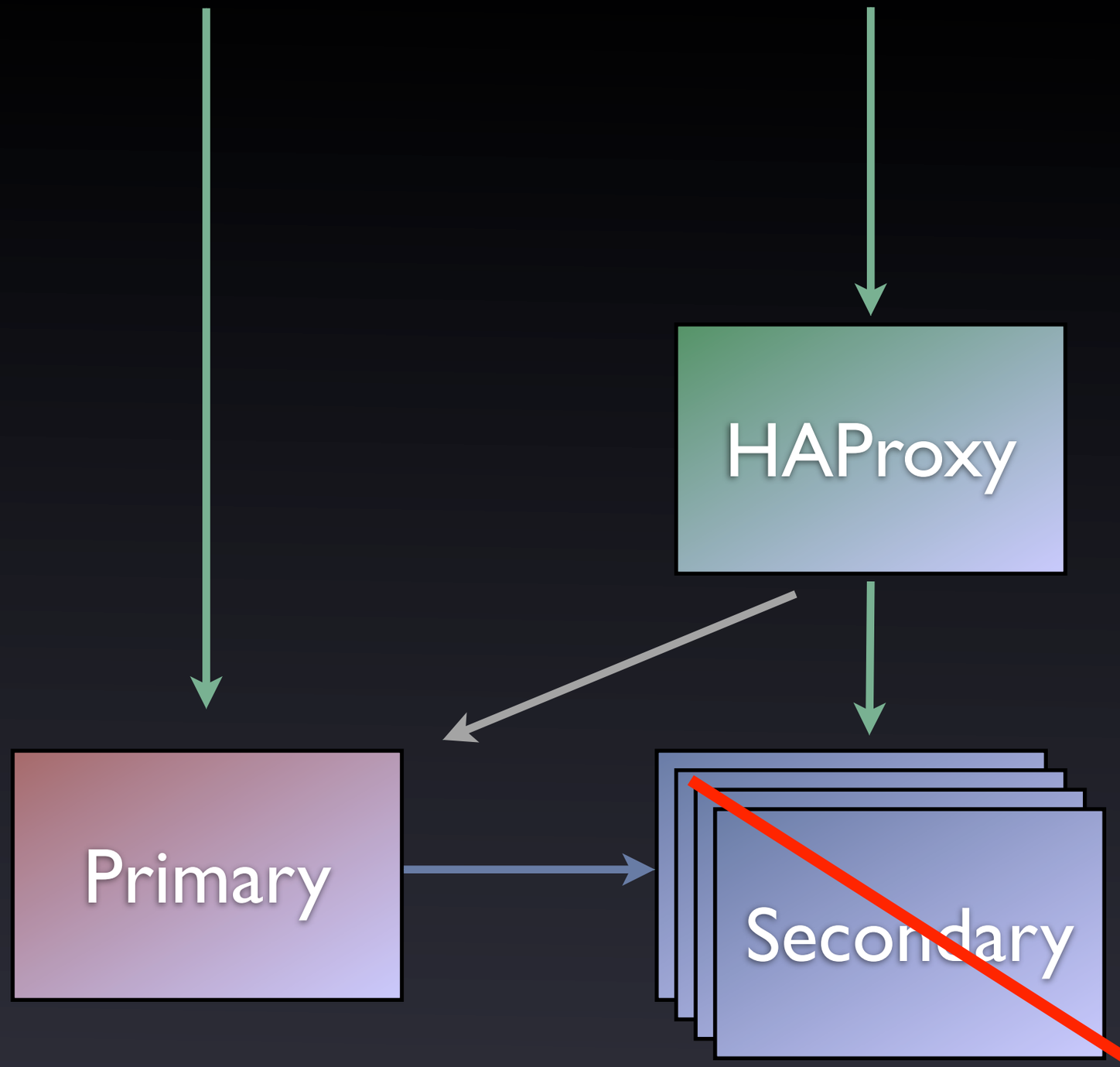
Notes.

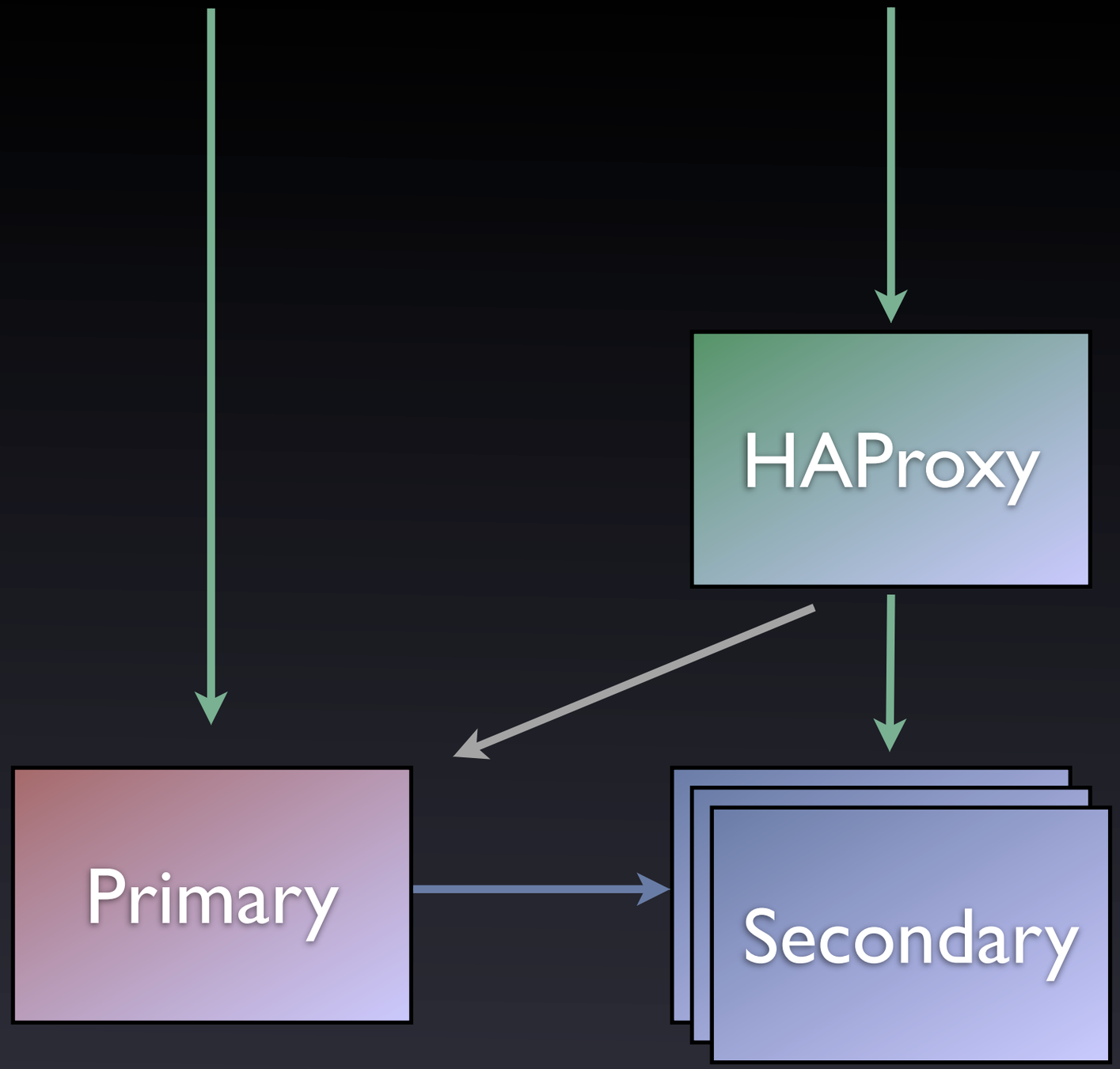
- Allows for pgbouncer as a pooling option.
- Tools exist to help with some tasks (handyrep, repmgr, etc.).
- Requires human intervention.
- Might be all a relatively simple setup requires.

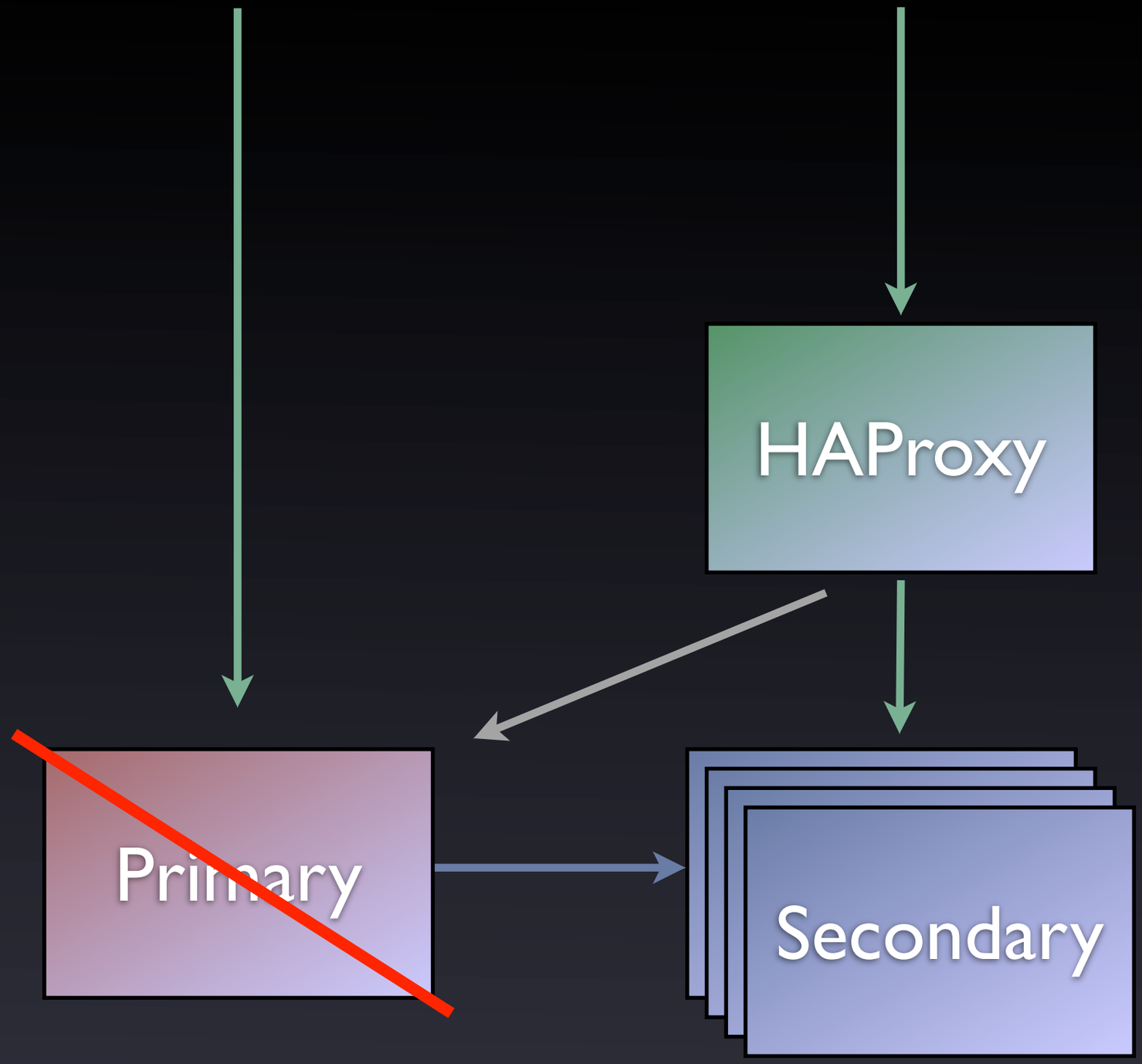
HAProxy

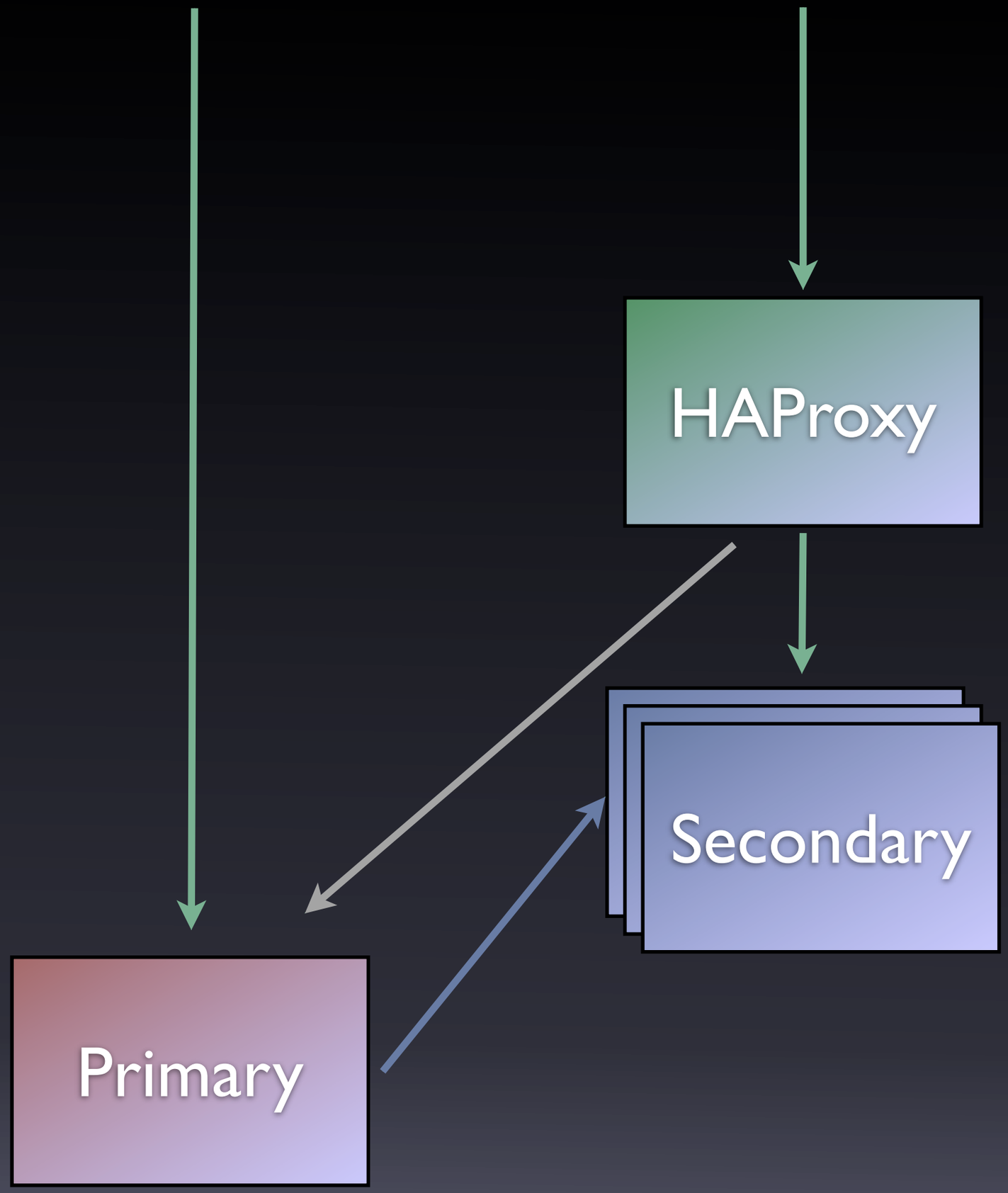
- HAProxy in front of a cluster of secondaries.
- If any secondary fails, HAProxy marks it down.
- Primary as backup server.
- If primary fails, manually promote a secondary to new role.


HAProxy
Primary
Secondary


HAProxy
Primary
Secondary


HAProxy
Primary
Secondary


HAProxy
Primary
Secondary



It does:

- Single endpoint *
- Load balancing *
- Environment agnostic
- Any number of secondaries
- Open source

It doesn't:

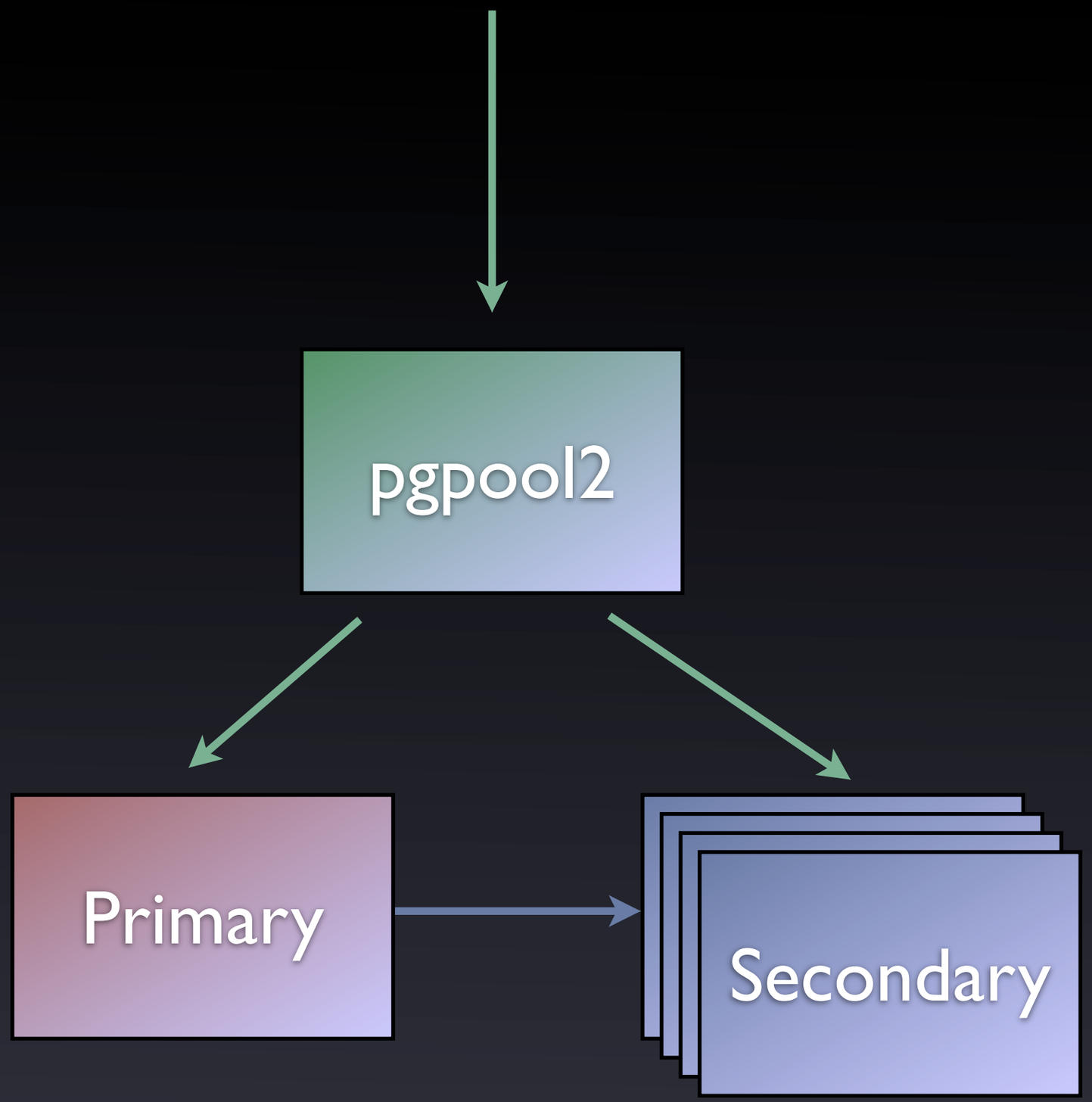
- Automatic promotion
- Reprovisions failed servers
- Connection pooling

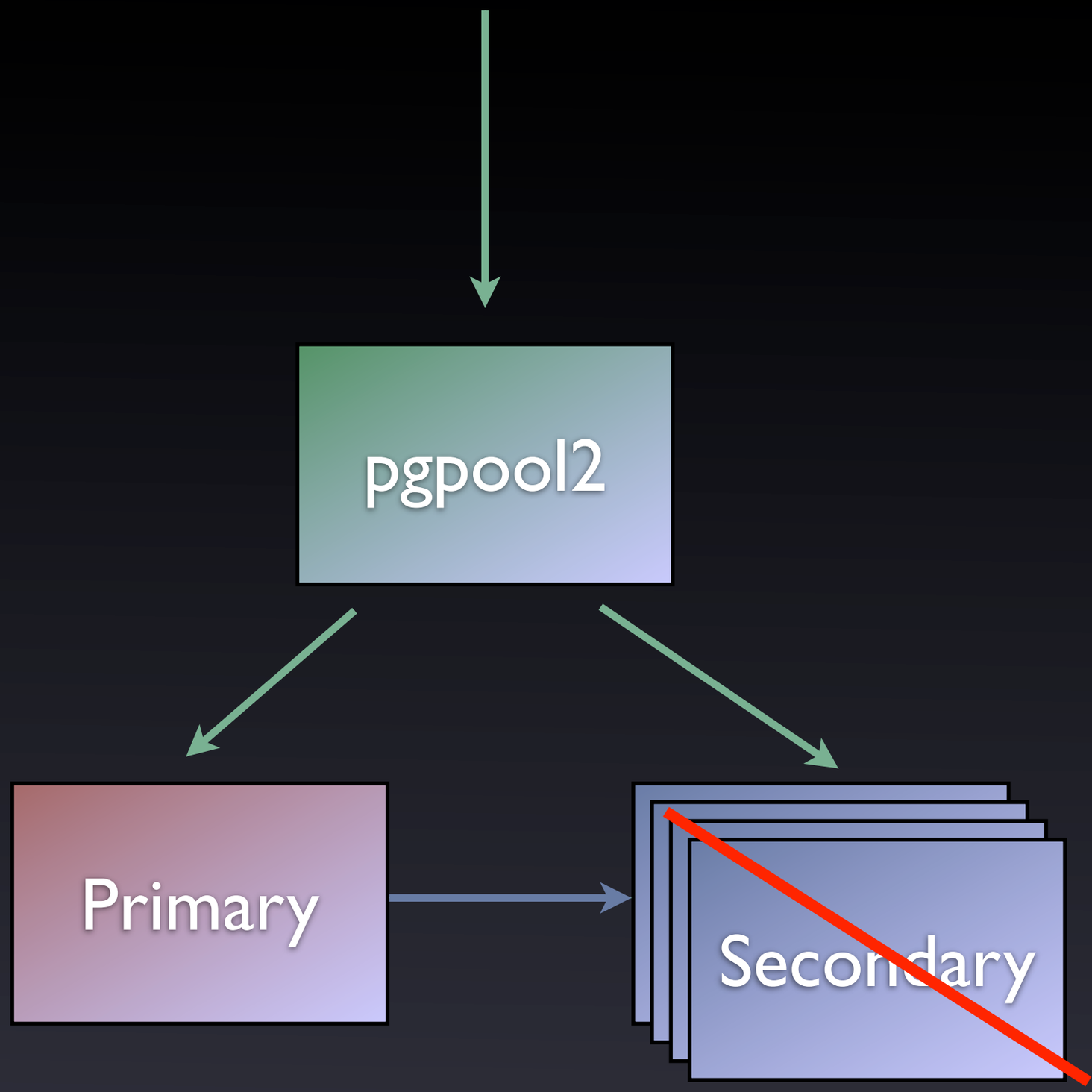
Notes.

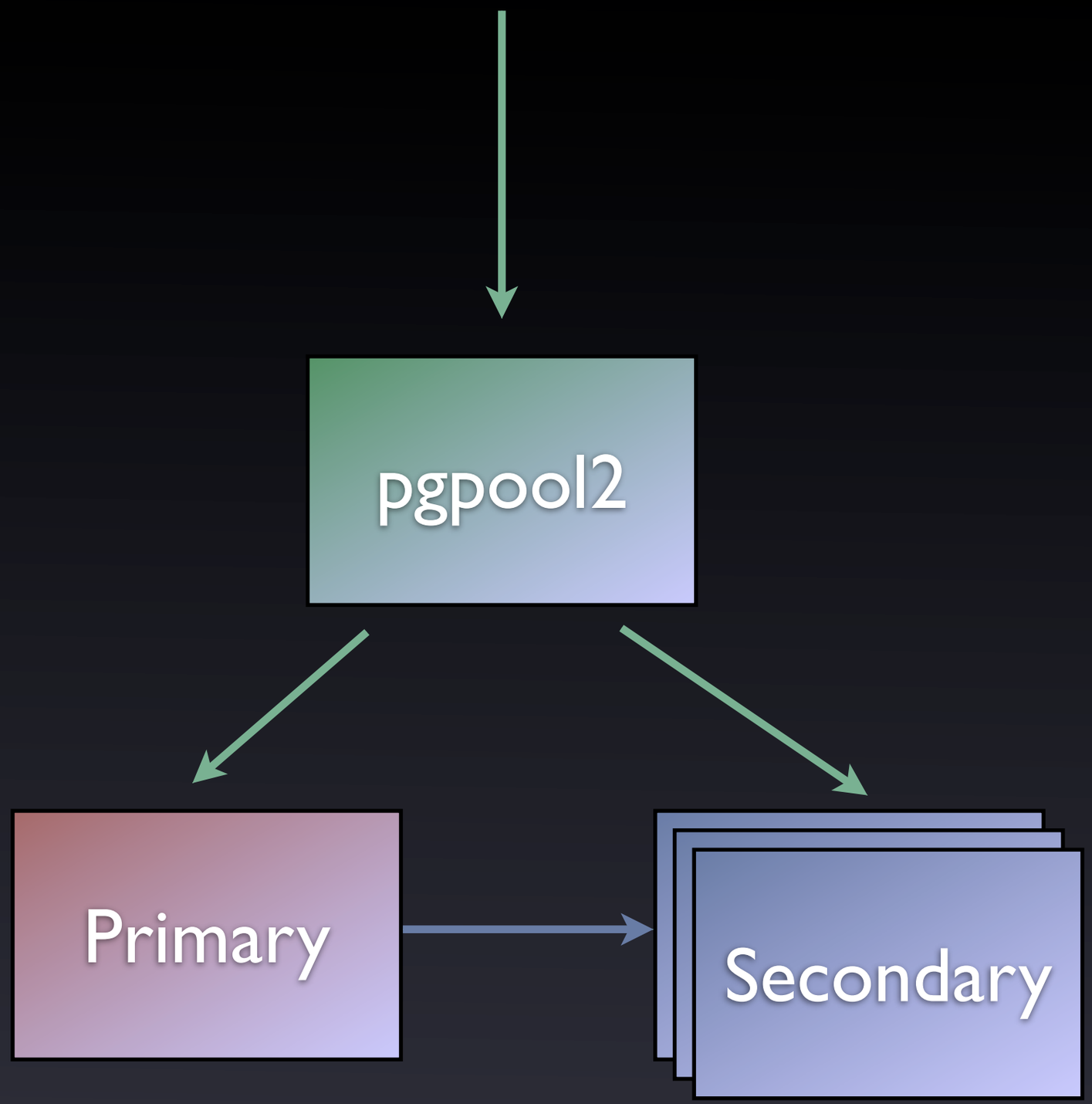
- Write traffic must be directed to primary.
- Lua scripting support might advance to allow for automatic promotion? Maybe?
- Mostly for balancing across secondaries.
- Requires HAProxy 1.6.

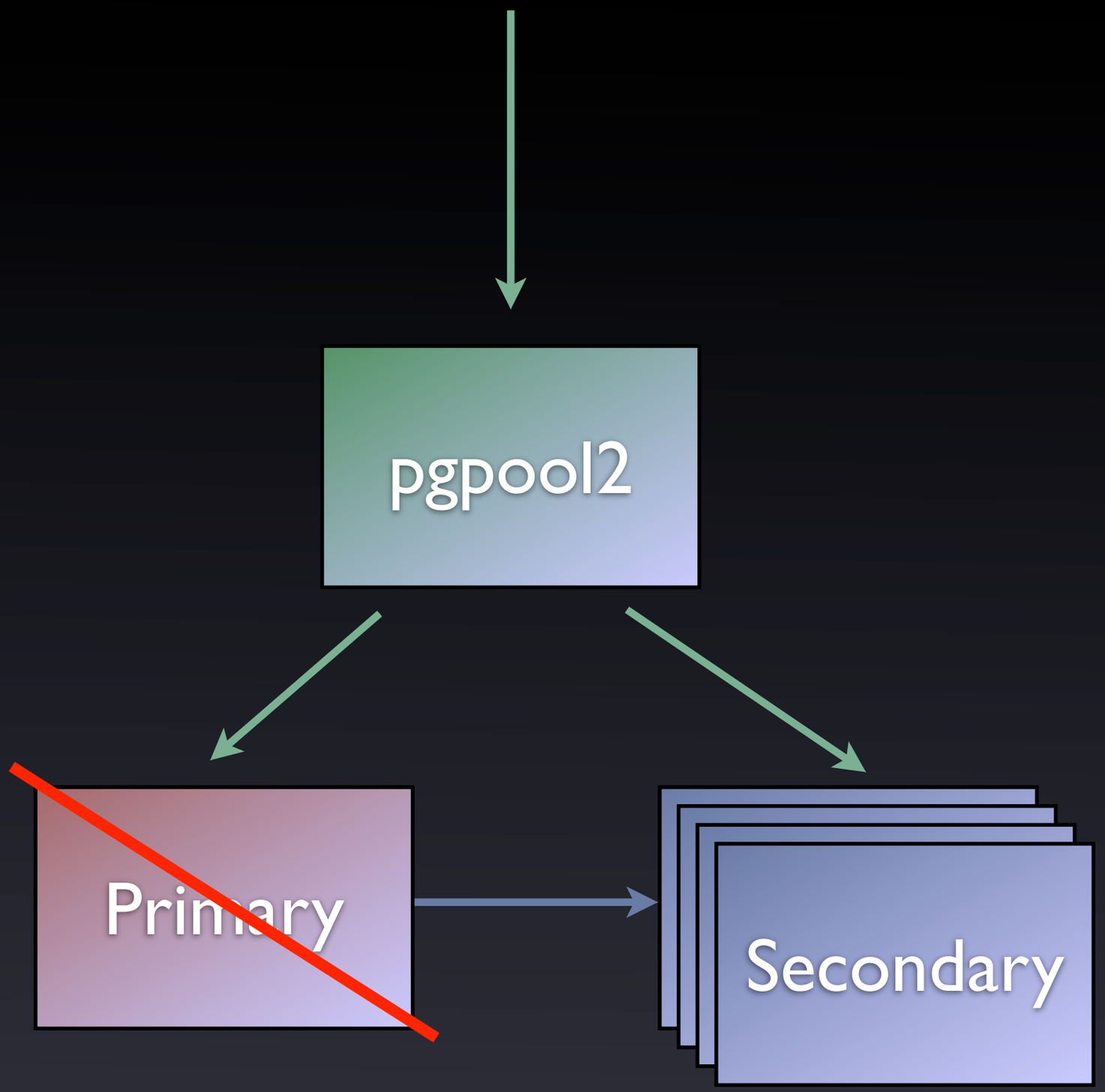
pgpool2

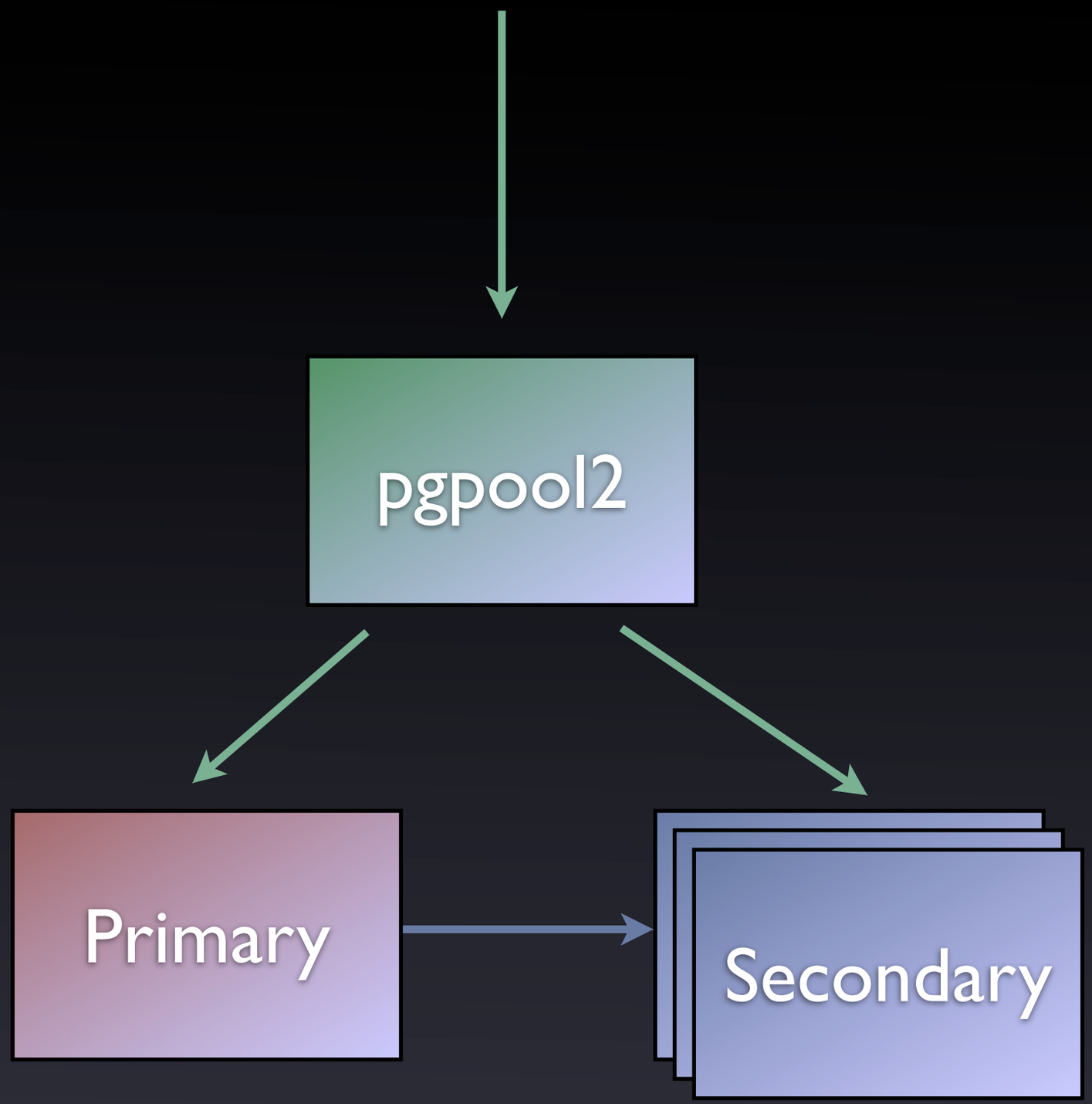
- The traditional solution to this problem.
- Front-end tool that accepts connections, and routes them.
- Can parse queries to assign to primaries or secondaries.











It does:

- Automatic promotion
- Single endpoint
- Load balancing
- Environment agnostic
- Any number of secondaries
- Open source

It doesn't:

- Reprovisions failed servers *
- Connection pooling

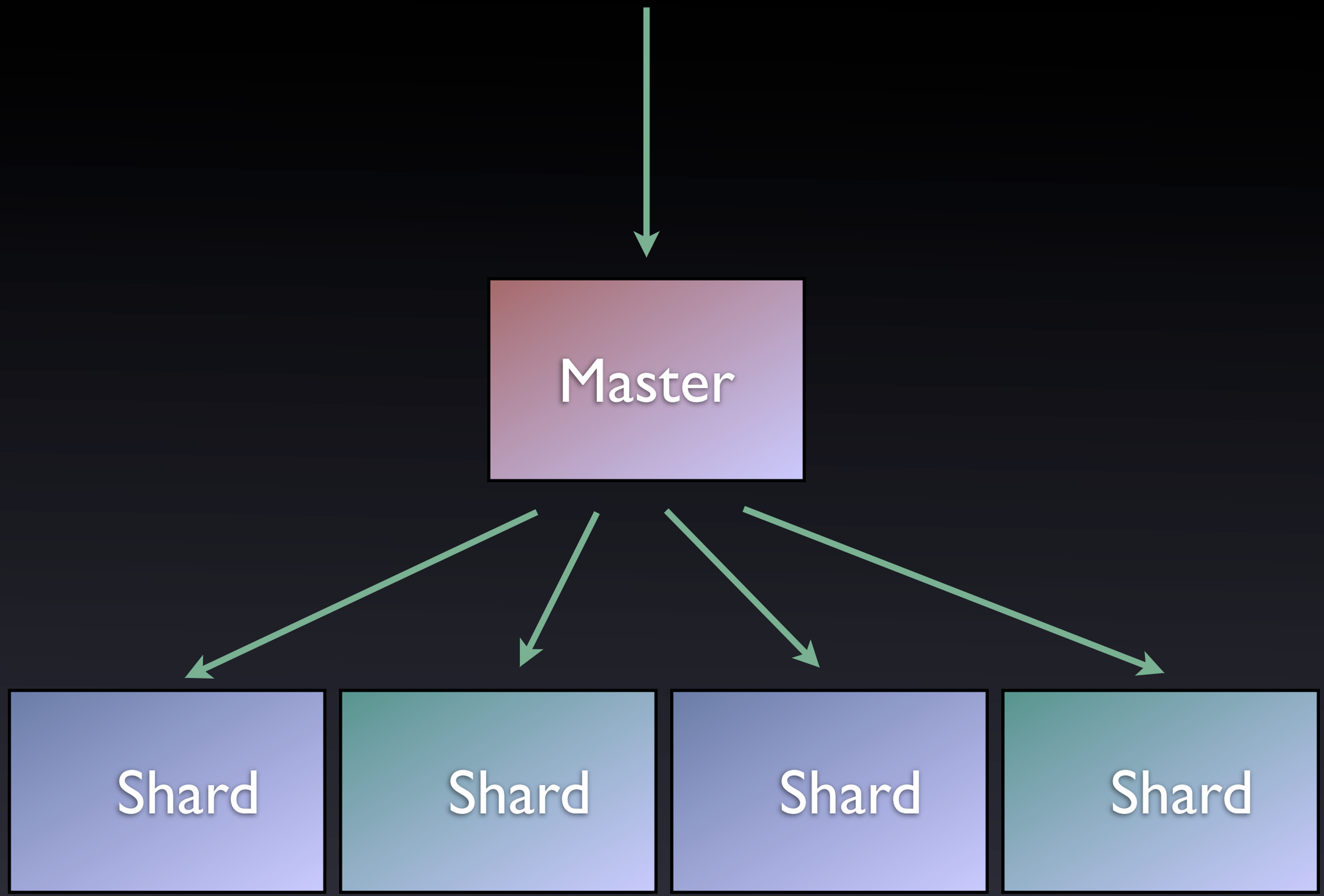
PGX
POSTGRESQL
EXPERTS, INC.

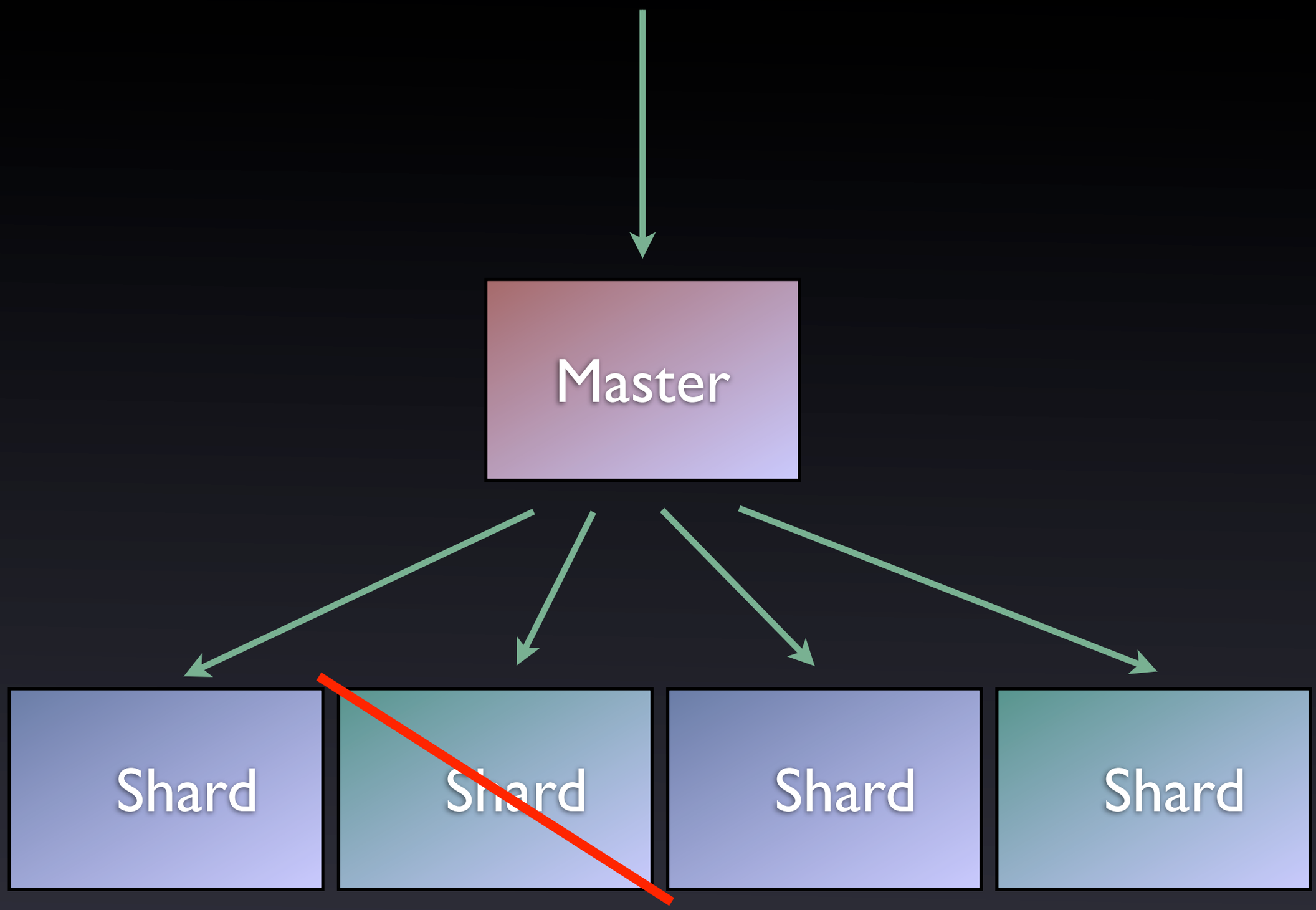
Notes.

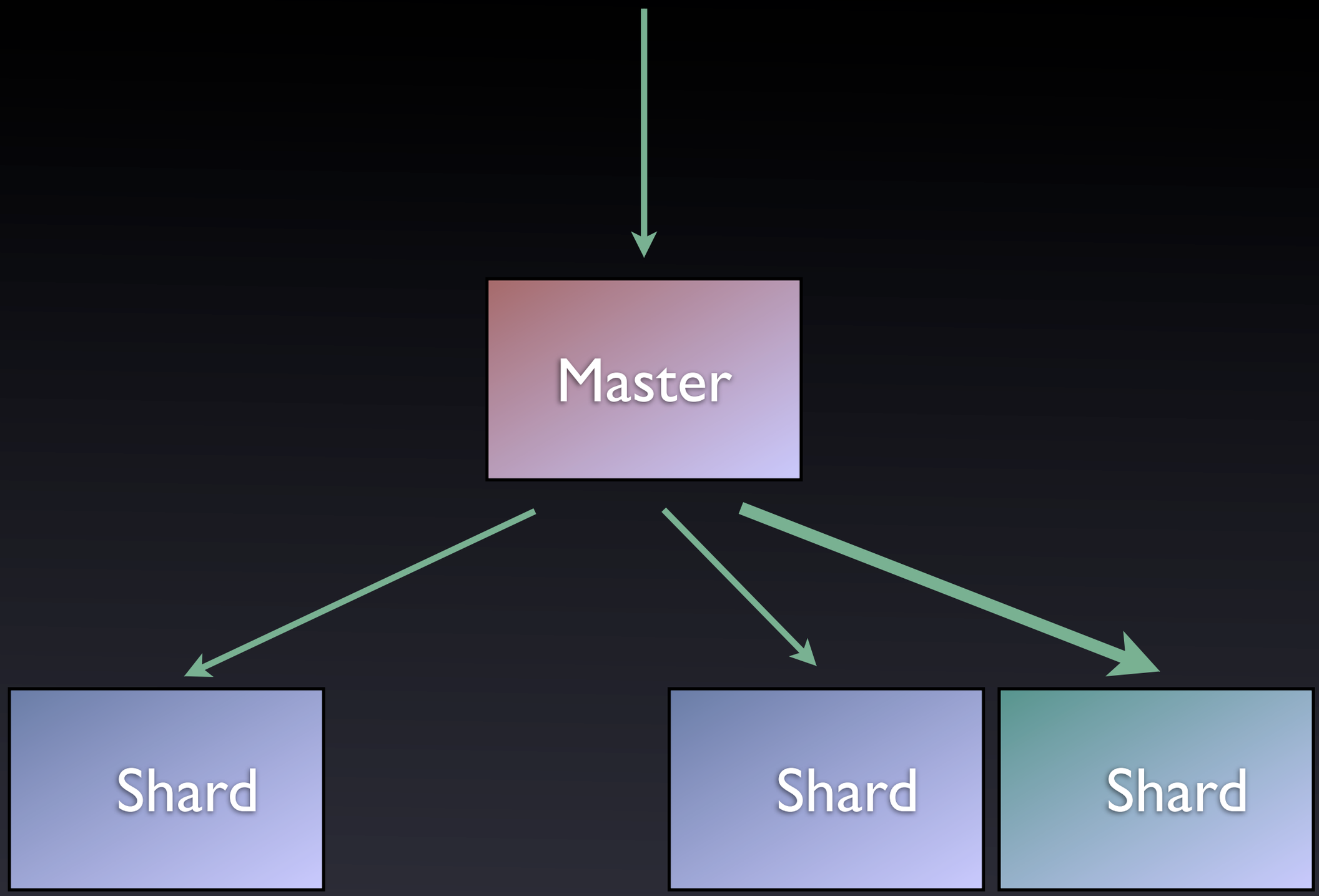
- Despite name, does not do "connection pooling" in the multiplexing sense.
- Does not have the best reputation for ease of installation or maintenance.
- Requires external scripting to do promotion and node management.

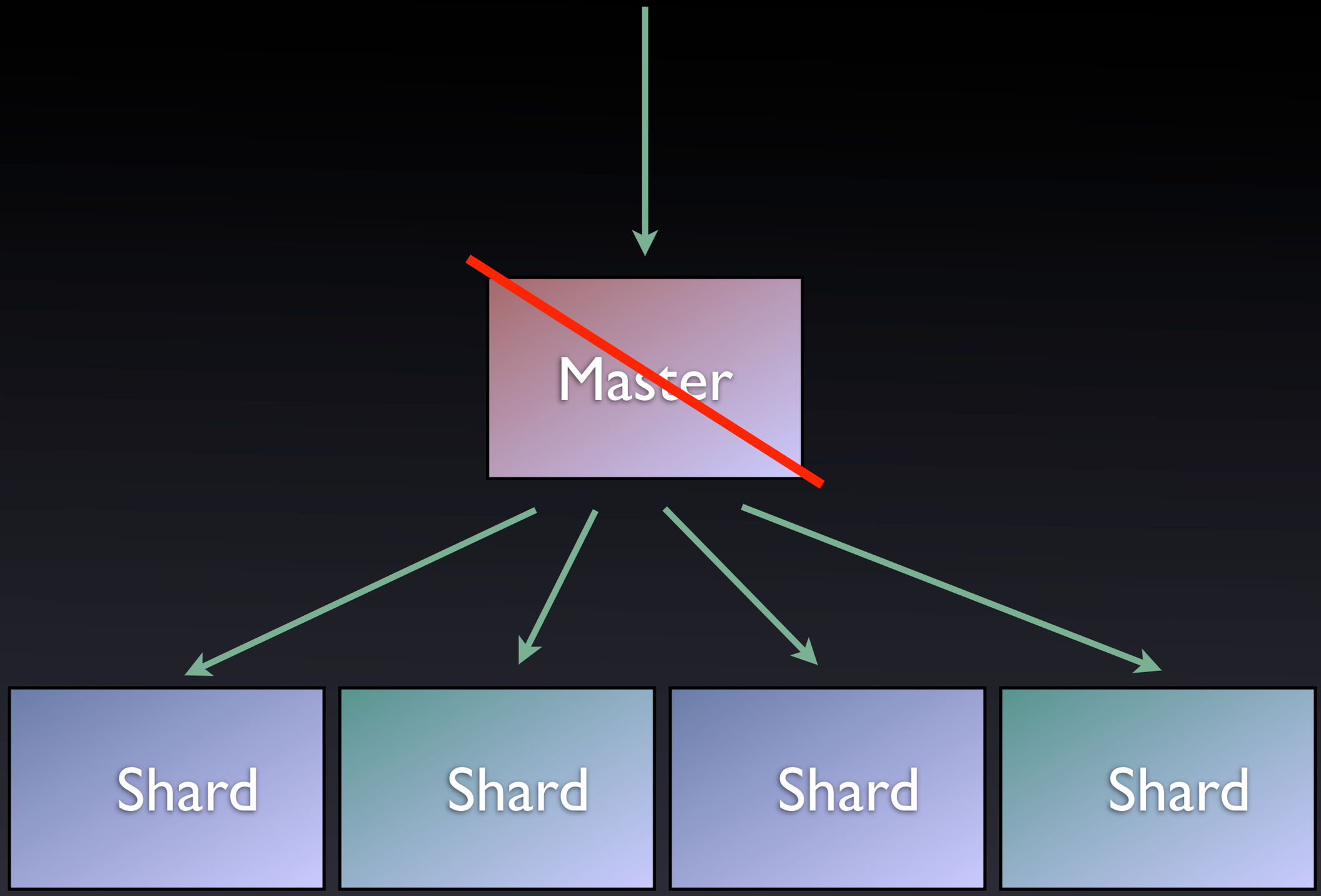
pg_shard

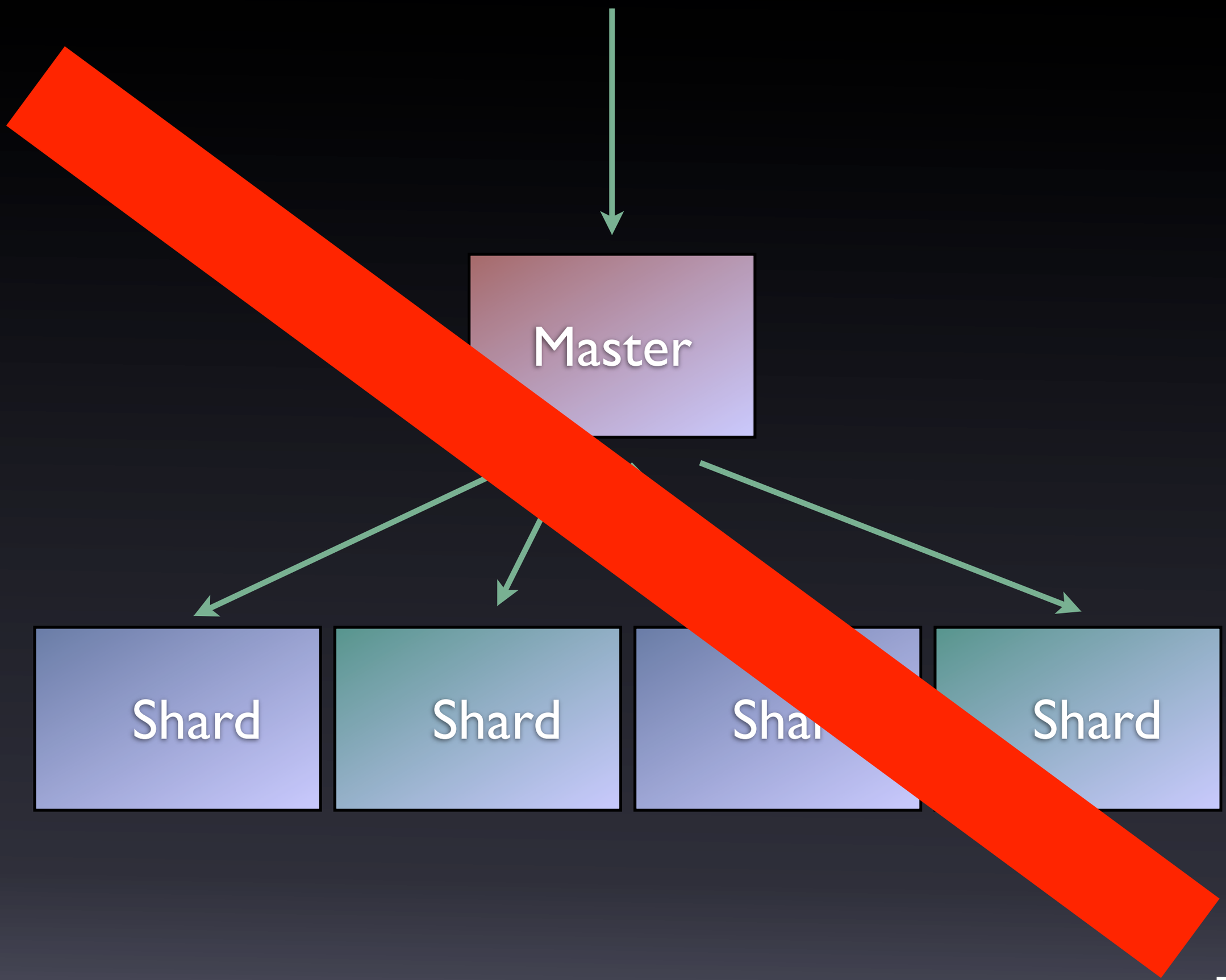
- Extension to PostgreSQL from Citus Data.
- A master node receives queries.
- A series of shard nodes holds portions of the data.
- HA provided by multiple shards holding the same set of data.











It does:

- Automatic promotion *
- Reprovisions failed servers *
- Single endpoint
- Load balancing
- Any number of secondaries *
- Open source

It doesn't:

- Environment agnostic
- Connection pooling

Notes.

- Master is a single point of failure.
  - … so it needs its own HA solution.
- Not transparent to clients.
- Significant restrictions on types of queries and the schema.
- Not just an HA solution: Also does distributed querying.

Heroku

- Commercial PostgreSQL-as-a-service offering.
- Specific for applications running on Heroku's compute service.
- Essentially a managed community PostgreSQL instance.
- You do not have superuser on the database.


Big Old Cloud of Apps
Database


Big Old Cloud of Apps
Database


Big Old Cloud of Apps
Database

It does:

- Automatic promotion
- Reprovisions failed servers
- Single endpoint
- Load balancing *
- Any number of secondaries
- Connection pooling *

It doesn't:

- Environment agnostic
- Open source

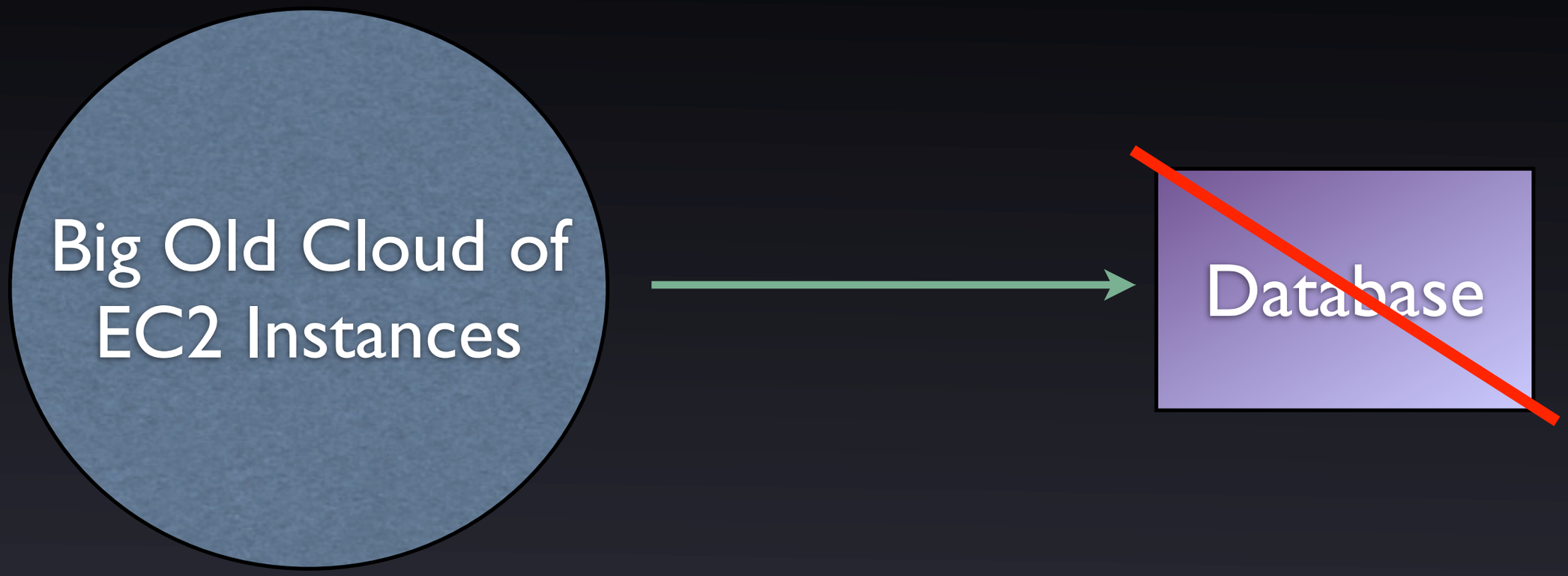
Notes.

- Heroku manages the database instance for you.
- Accounts can create their own secondaries, for both load balancing and failover purposes.
- Complex relationship between HA features and account plans.

Amazon RDS

- Amazon's PostgreSQL-as-a-service offering.
- A package of:
  - A managed EC2 instance.
  - A managed PostgreSQL instance.
  - A "shadow" failover machine (using proprietary replication technology).


Big Old Cloud of EC2 Instances
Database


Big Old Cloud of
EC2 Instances
Database


Big Old Cloud of EC2 Instances
Database

It does:

- Automatic promotion
- Reprovisions failed servers
- Single endpoint

It doesn't:

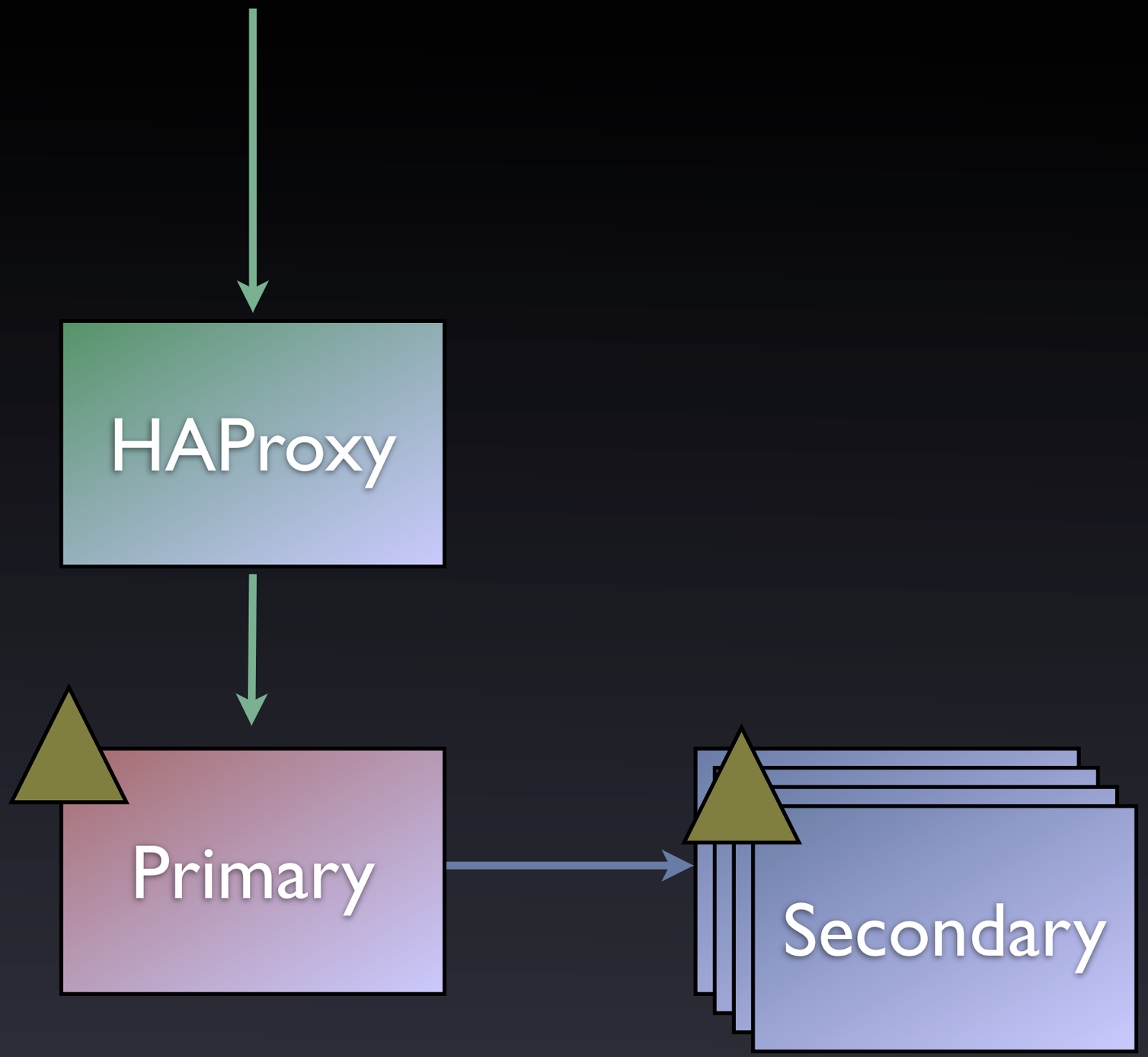
- Load balancing
- Environment agnostic
- Any number of secondaries
- Connection pooling
- Open source

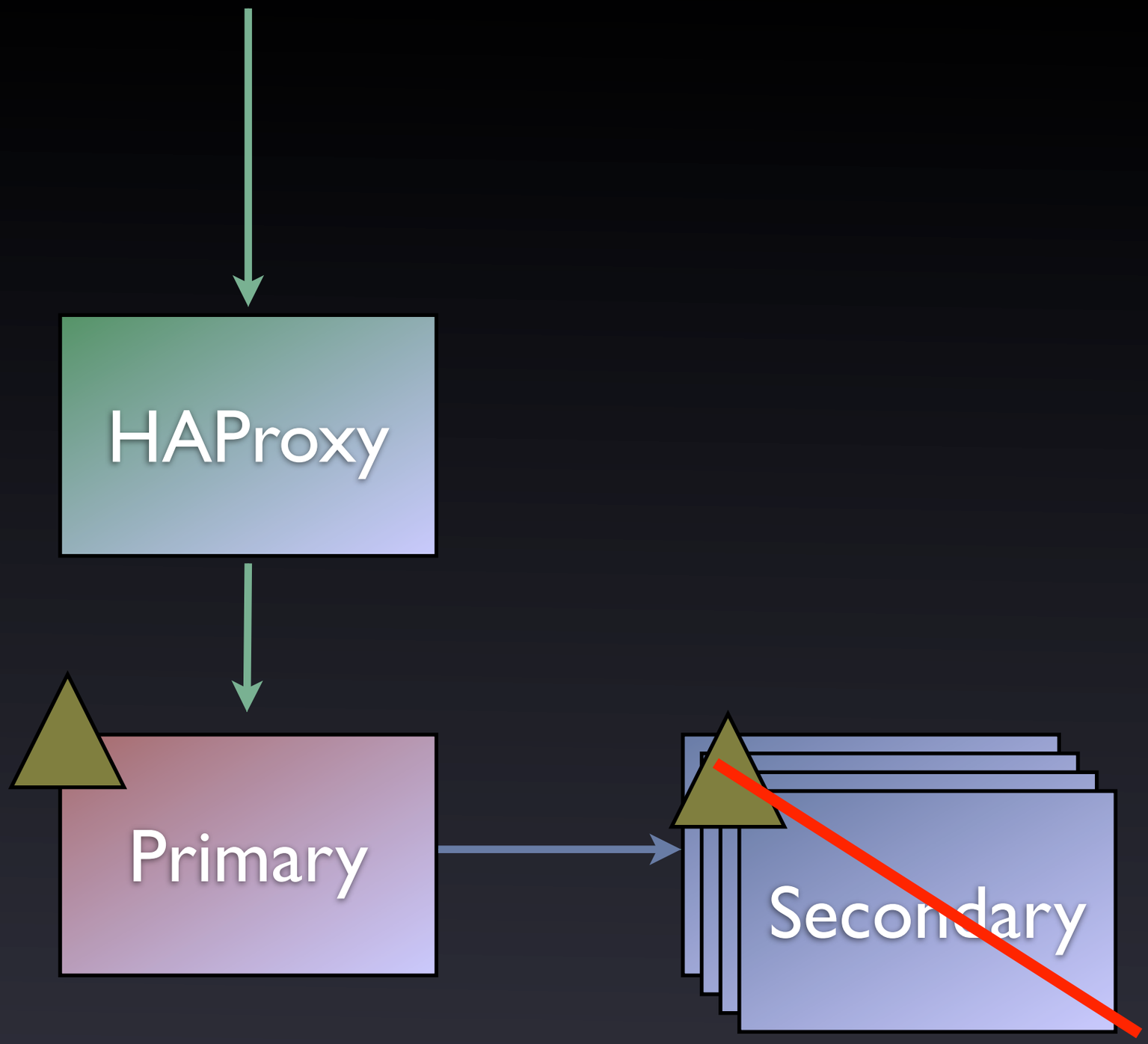
Notes.

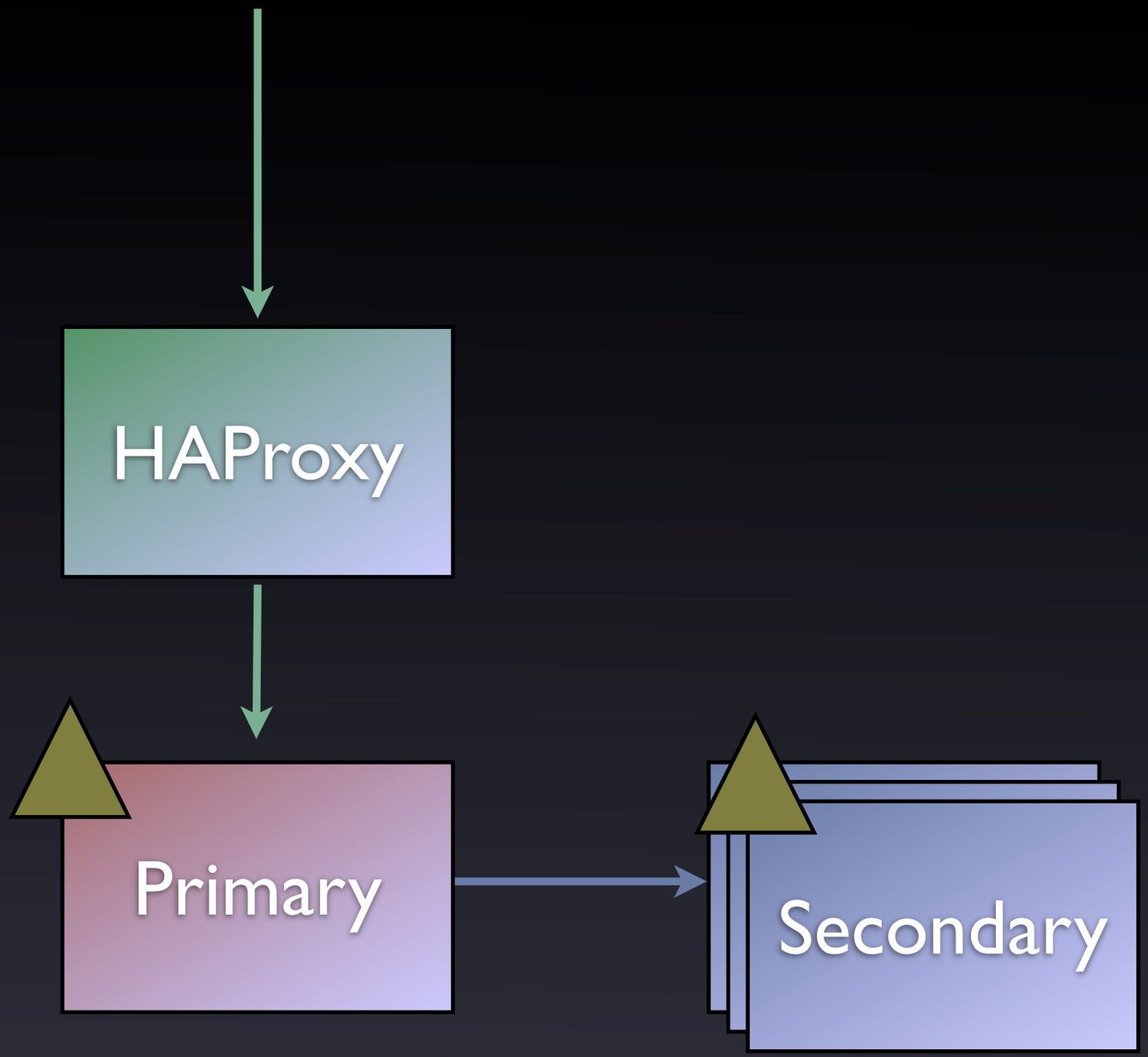
- Strange environment if you are used to community PostgreSQL.
- Can put pgbouncer in front for pooling.
- Can create secondaries, but they are load-balance only, not HA.
- You do not have superuser on the database.

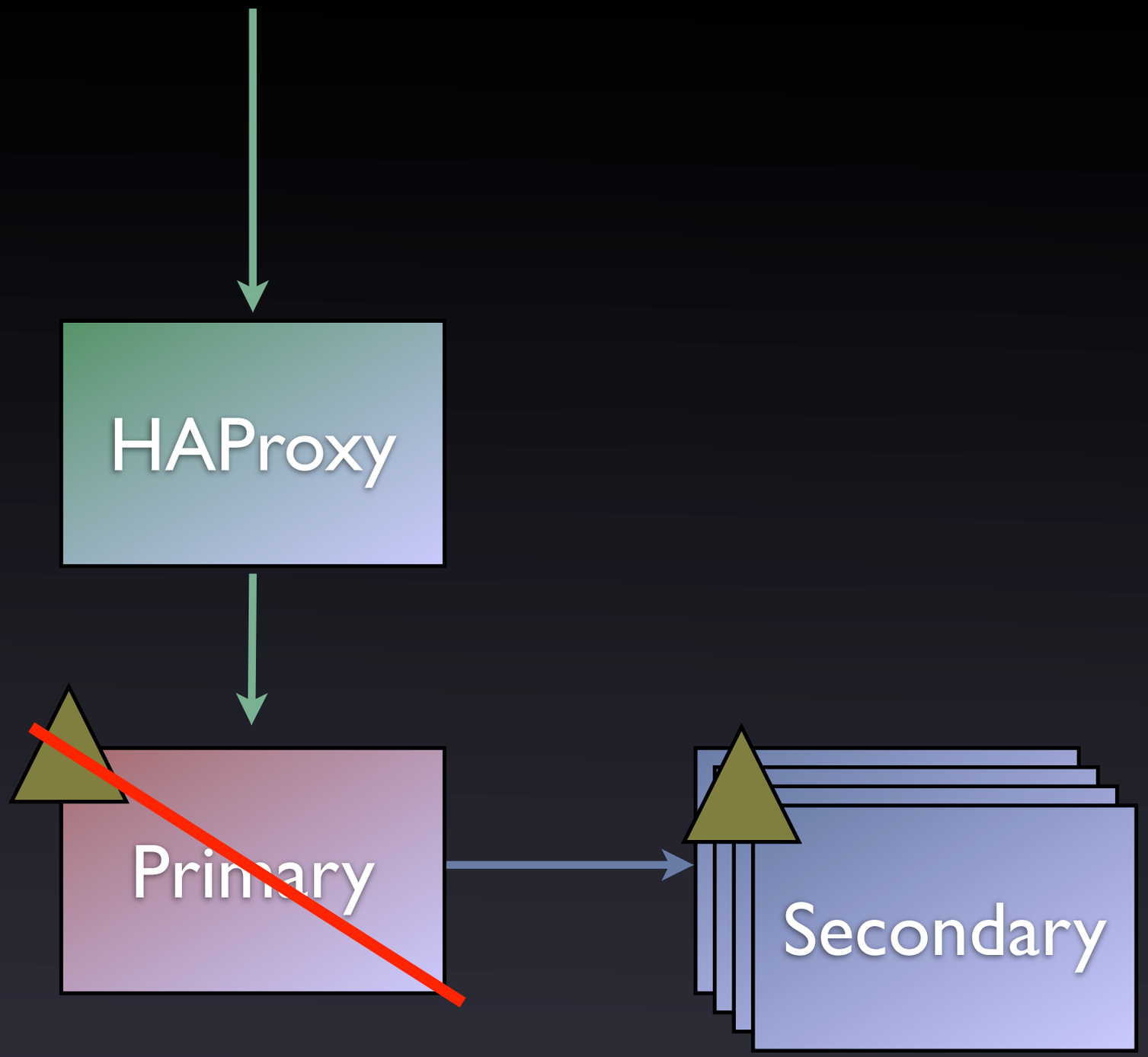
Patroni

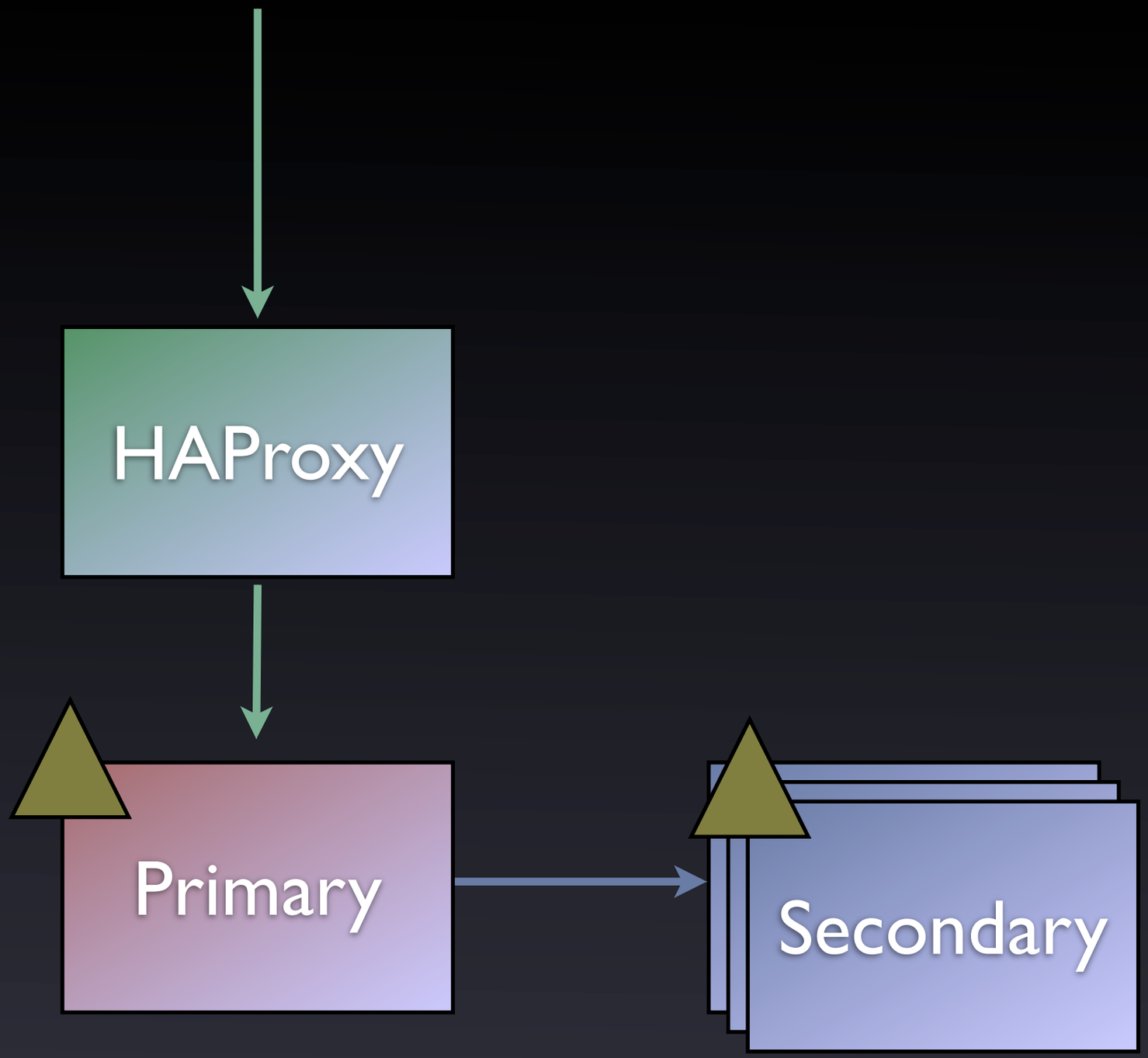
- https://github.com/zalando/patroni
- Python tool/daemon for managing PostgreSQL servers.
- New, under active development.
- Uses HAProxy as its front end tool.
- Uses etcd or Zookeeper as a distributed system config database.











It does:

- Automatic promotion
- Reprovisions failed servers
- Single endpoint
- Any number of secondaries
- Open source

It doesn't:

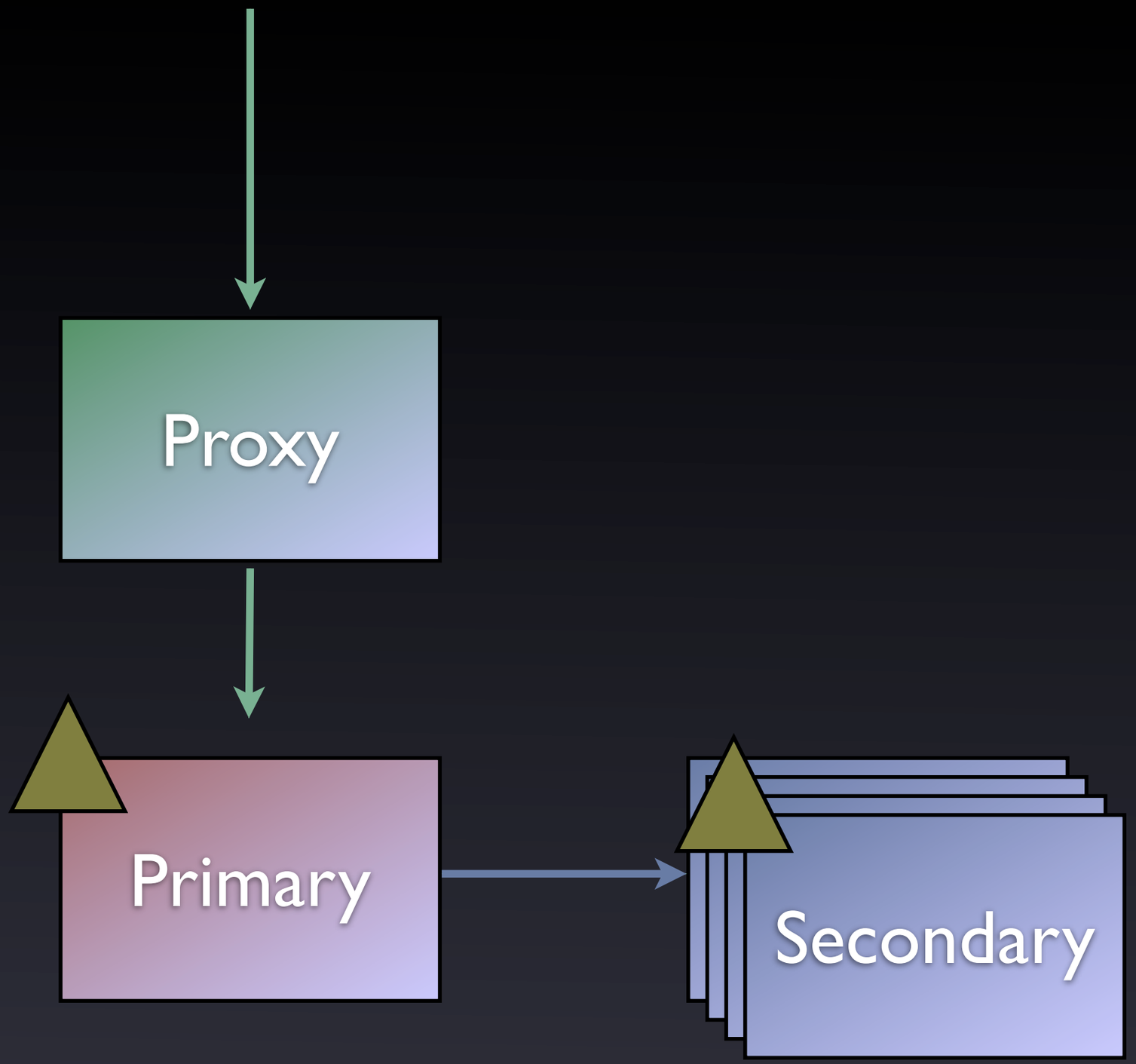
- Load balancing
- Environment agnostic
- Connection pooling

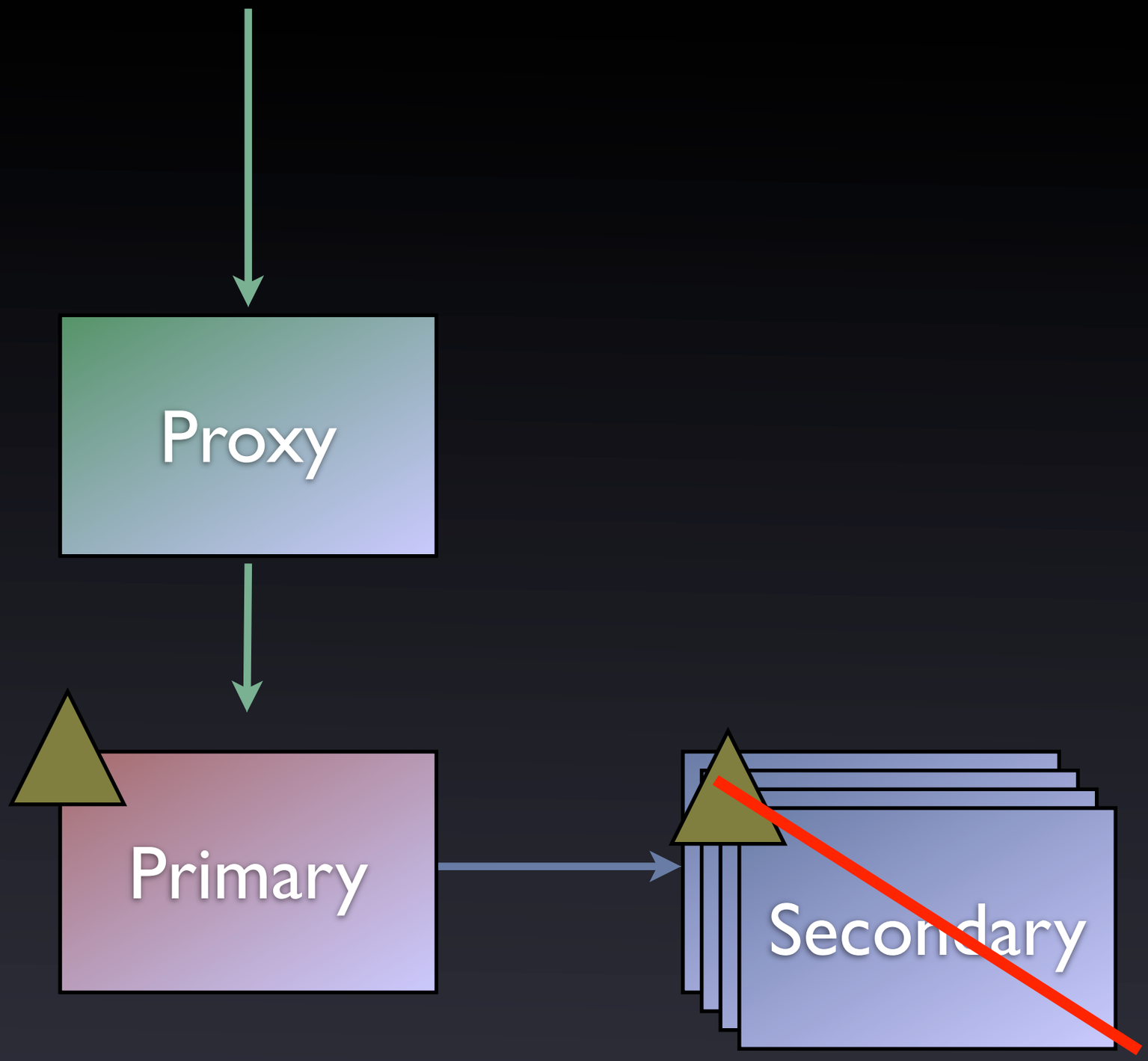
Notes.

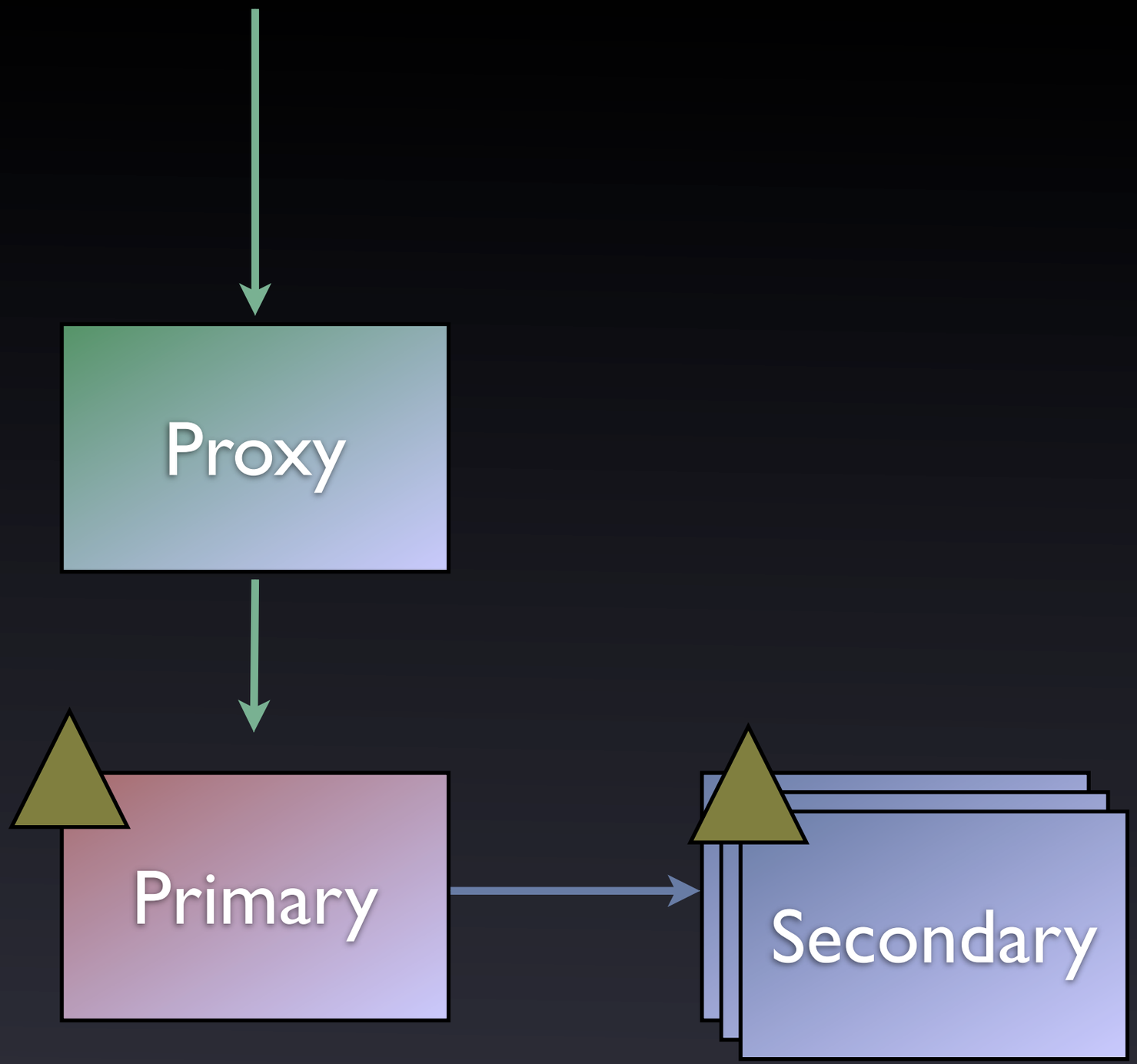
- The HAProxy is used to route to the current primary.
- You can provision secondaries, but they're on a different endpoint.
- Hear more about it later today!

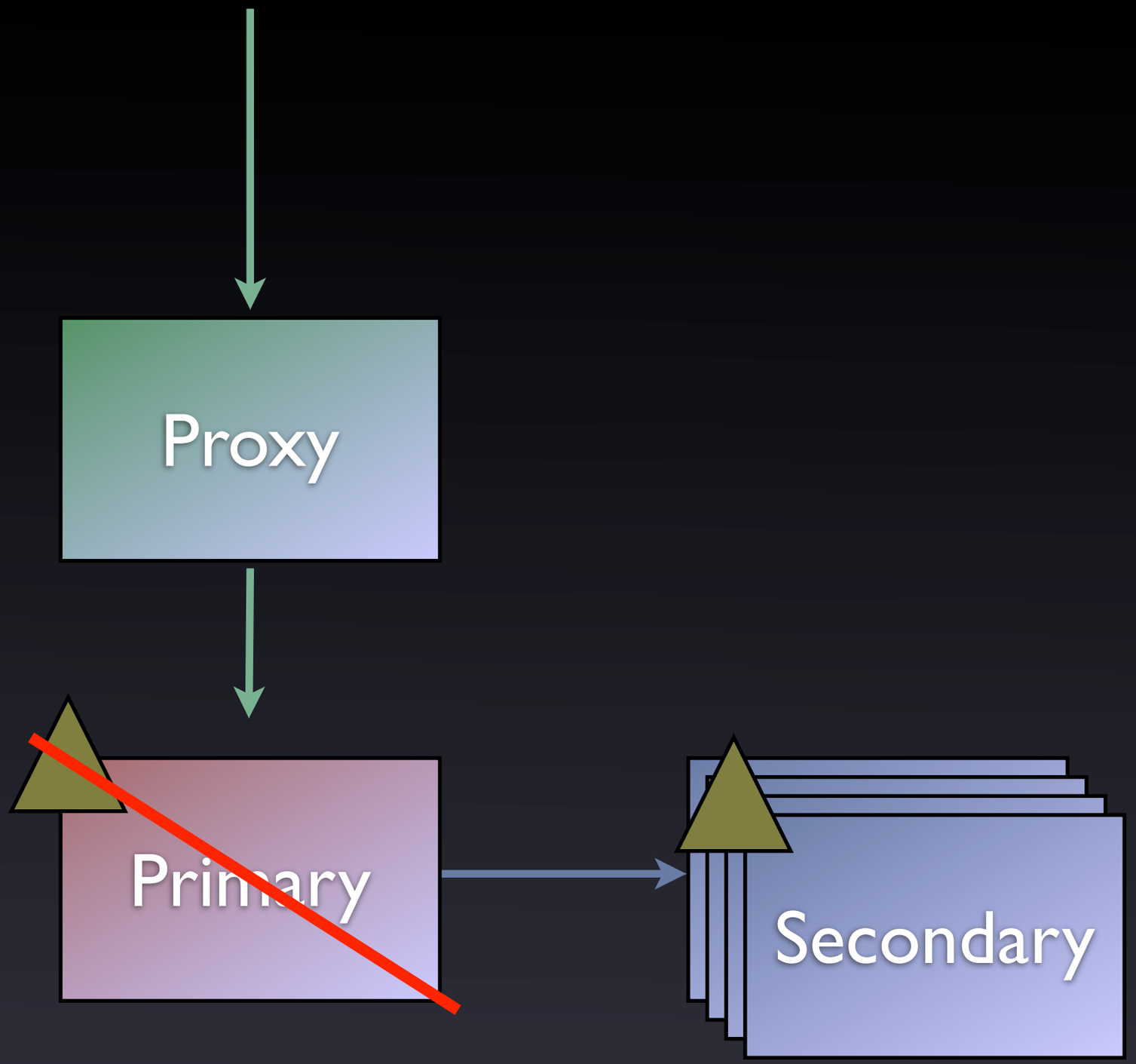
Stolon

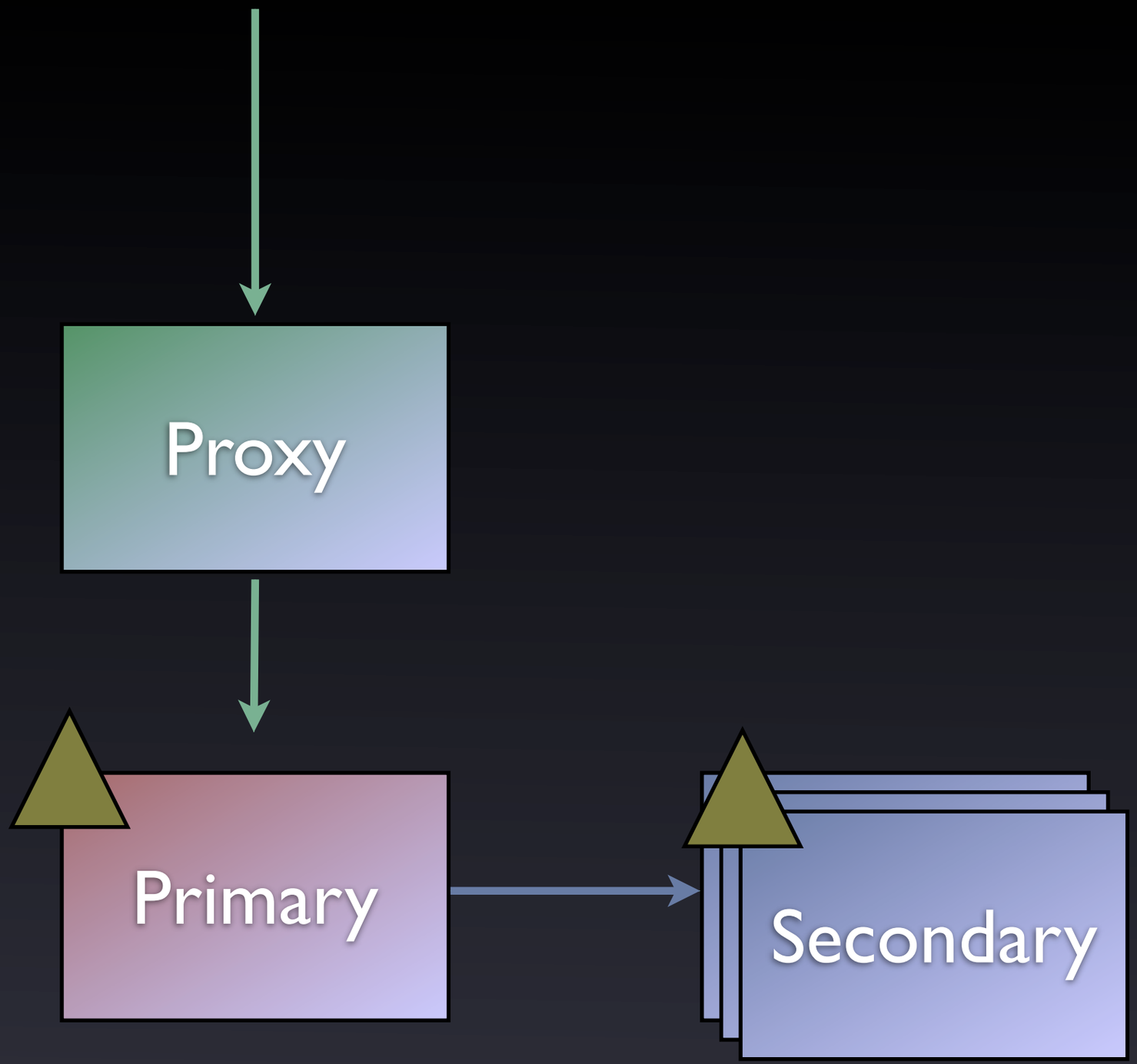
- https://github.com/sorintlab/stolon
- Relatively new.
- Under active development.
- Written in Go.








Proxy
Primary
Secondary



It does:

- Automatic promotion
- Single endpoint
- Any number of secondaries
- Open source

It doesn't:

- Reprovisions failed servers
- Load balancing
- Environment agnostic
- Connection pooling

Notes.

- Requires etcd or consul.
- Has a custom proxy.
- New secondary provisioning possible (?) if using kubernetes.

Another way to look at it.

How much do you like systems administration?

How much do you like system adminstration?

- I would rather eat my own foot: Heroku, Amazon RDS.
- Some, but I don't want to live it: Bare streaming replication, HAProxy.
- I'm OK with it (and don't mind some development): pg_shard, Patroni, Stolon.
- I laugh at danger: pgpool2.

One thing nothing does.

- Completely transparent failover.
- All solutions will break connections or cancel queries on a failure.
- Applications recover from this with varying degrees of grace.

So.

The perfect HA tool is yet to be.

- More work needs to be done here.
- … and a lot **is** being done.
- … but for deployments right now, you need to make some choices among the available tools.

Some day.

Thank you!

Christophe Pettus
@xof

thebuild.com
pgexperts.com