Alibaba and PostgreSQL

Practices on providing PG as a **cloud** service in Alibaba

Guangzhou Zhang
guangzhou.zgz@alibaba-inc.com

About this talk

- Firstly some background about PG in Alibaba
- Then issues and our solutions with providing PostgreSQL as a cloud service
- Finally future directions of our database cloud service
- Focuses on technical details and NOT from business-level or application development perspectives

About me

- Lead of database engine development team in cloud service of Alibaba
- Worked as a database engine developer since yr. 2006
- Email: guangzhou.zgz@alibaba-inc.com

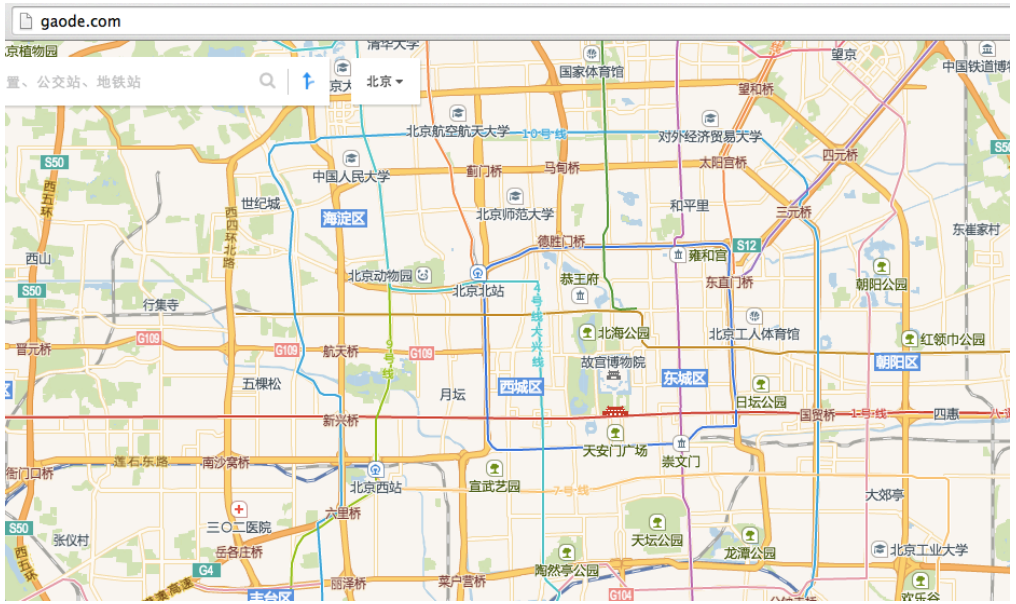
PostgreSQL and Alibaba

- Alibaba
  - Founded in 1999，now one of the world largest internet companies
  - Its cloud services (alicloud.com) cover both private and public cloud, which have gained the largest market share in China and being growing fast (revenue up 126% in 2015 Q4)

- Alibaba loves PostgreSQL
  - Uses PG in its own internet businesses
  - Government and state-owned companies are moving out of Oracle solutions. PG is the most perfect replacement for Oracle
  - Counts on PG to attract Oracle users to its cloud services

PostgreSQL and Alibaba

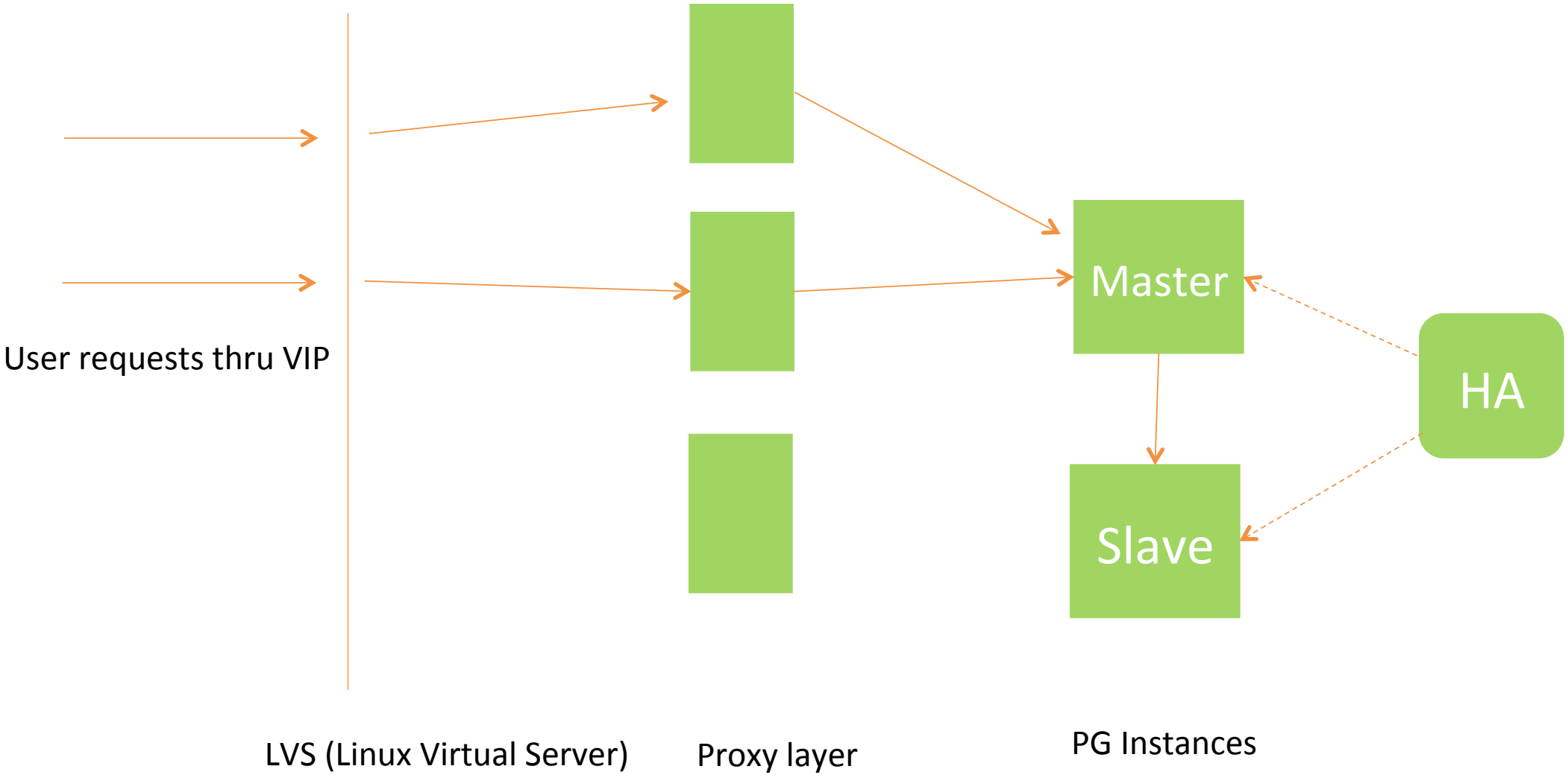
- Alibaba uses PG for its own internet businesses
  - Map service
  - Online to Offline services
  - CRM



PostgreSQL and Alibaba

- Provided PG as a cloud service at alicloud.com
  - AliCloudDB database service for PG went online at alicloud.com in 2015.6
  - Cooperated with EDB to provide EDB’ s Postgres Advanced Server as a cloud service

Architecture of cloud service for PG



Issues on enabling PG as a cloud service

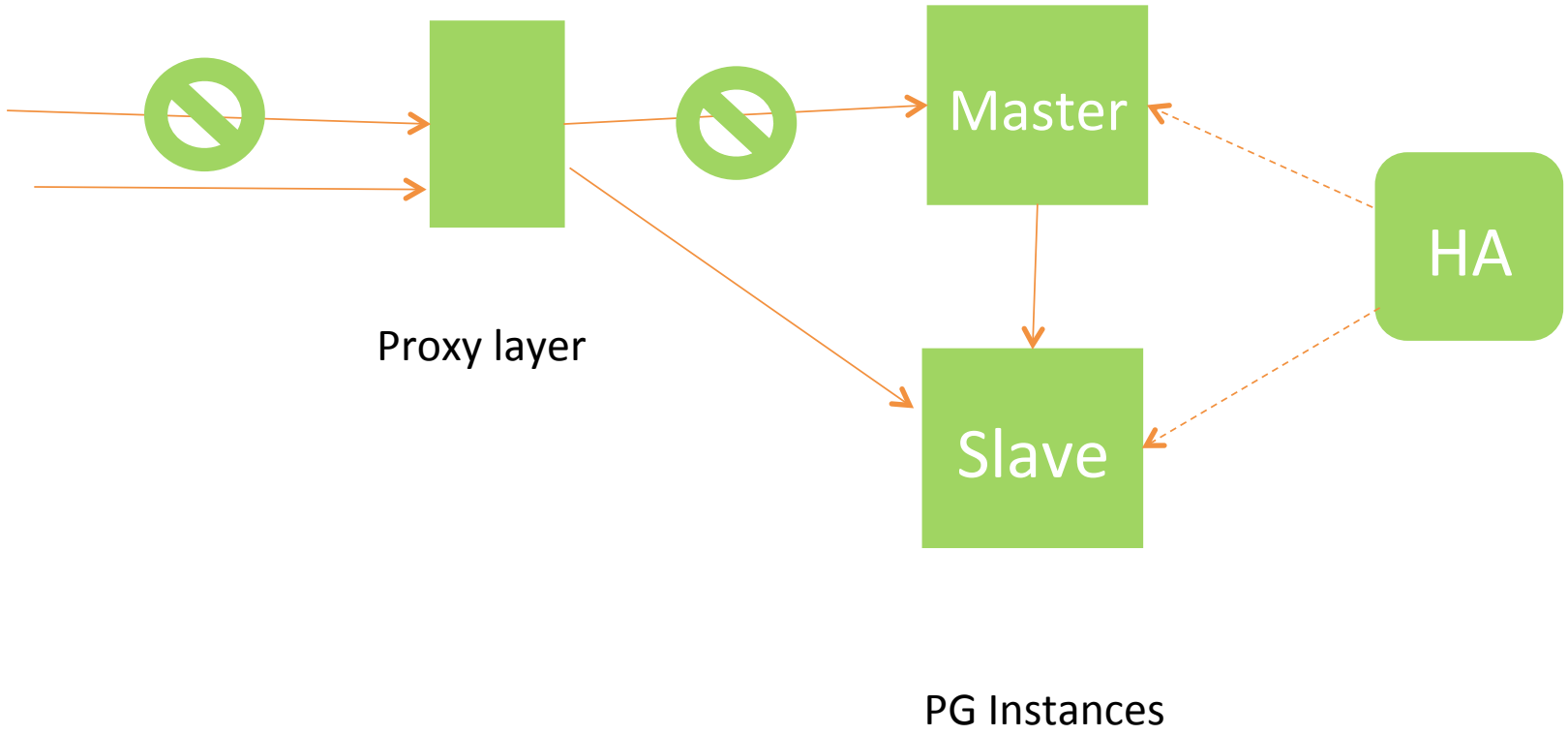
- Transparent switch-over with Proxy
- Transparent connection pool with Proxy
- Handling OOM
- Handling IO hang
- Privilege Management

Needs for transparent switch-over

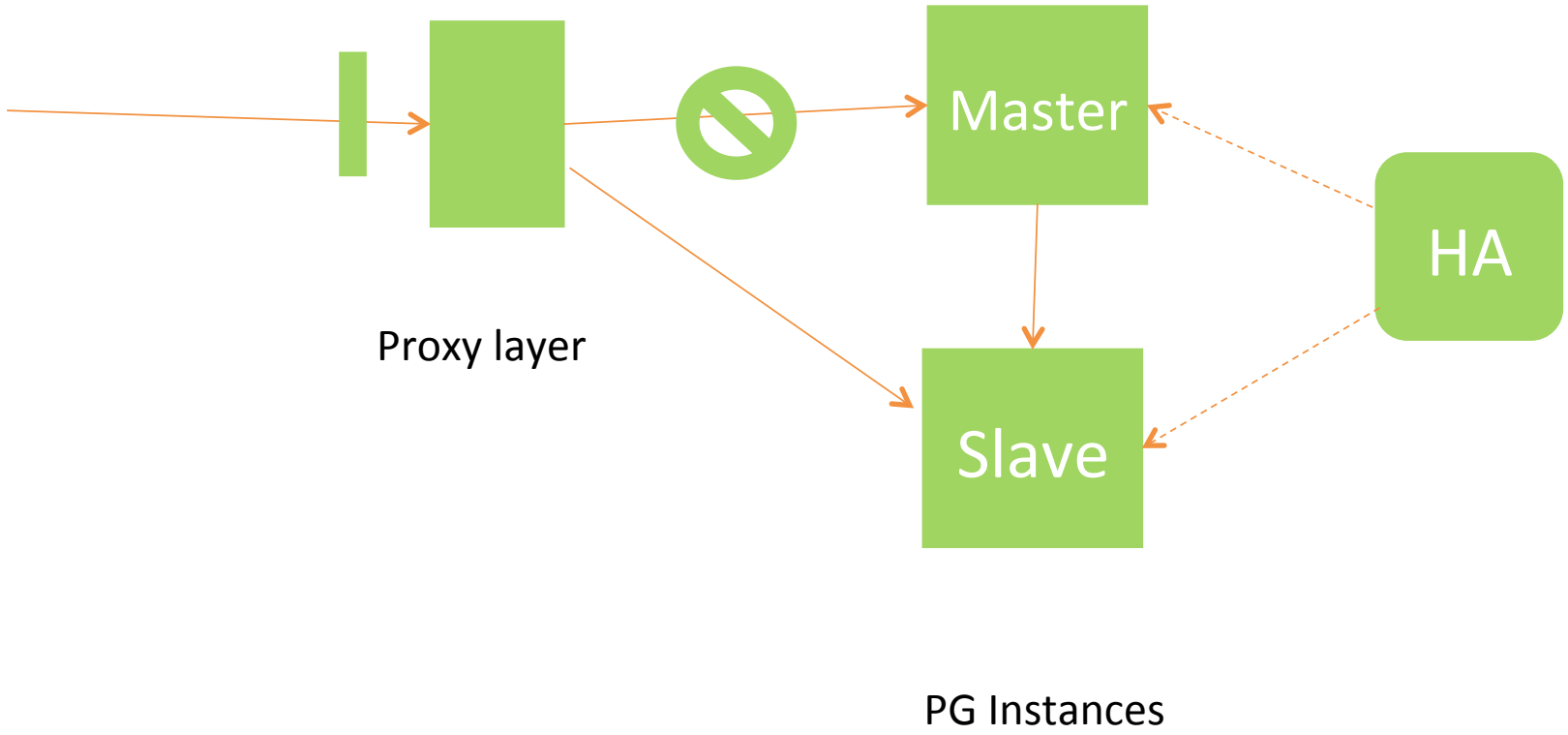
- There are quite a few cases requiring instance restart. And we implement a restart with switch-over
  - Upgrade or degrade an instance’ s class (e.g. from 2G mem to 4G)
  - Move an instance from one machine to another
  - PG kernel version upgrades

- Transparent switch-over is the process that does a switch-over without breaking existing user connections. This is key to SLA and customer satisfactory

A normal switch-over



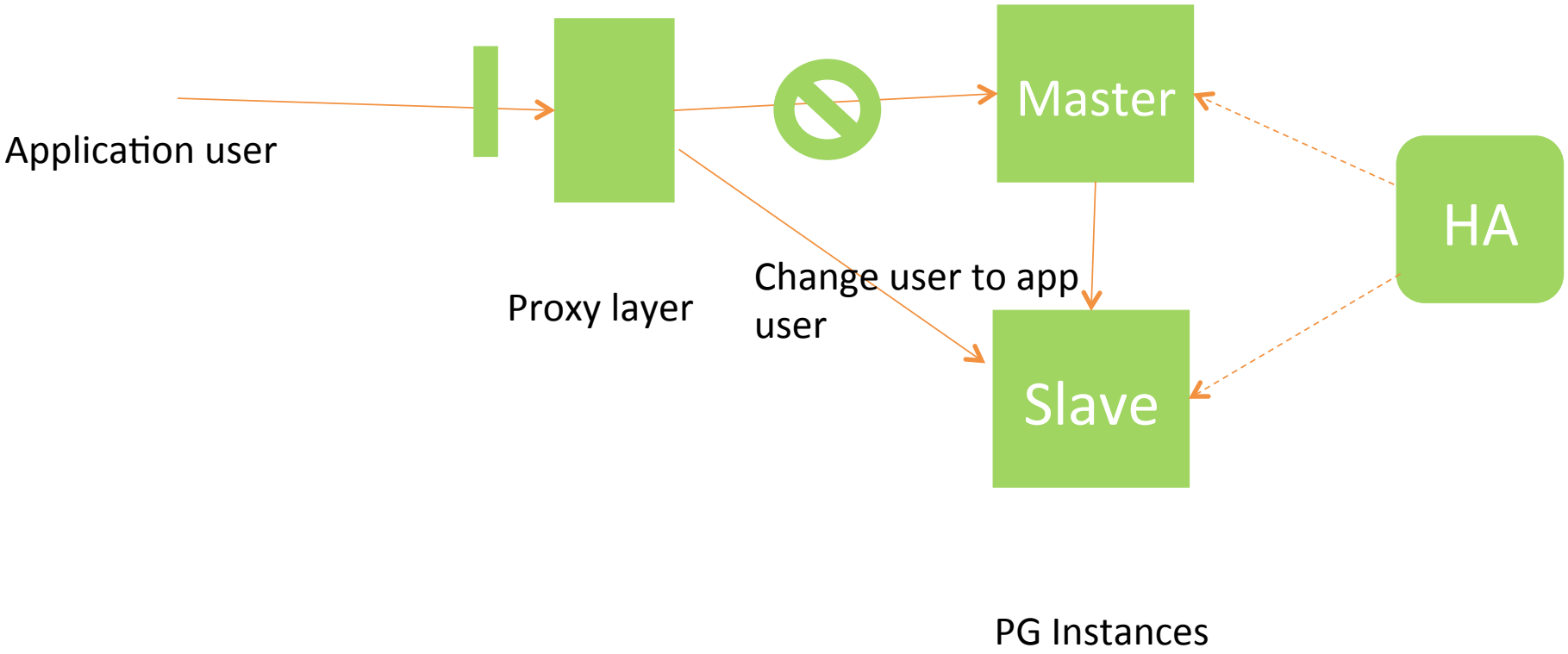
A transparent switch-over



Transparent switch-over with Proxy

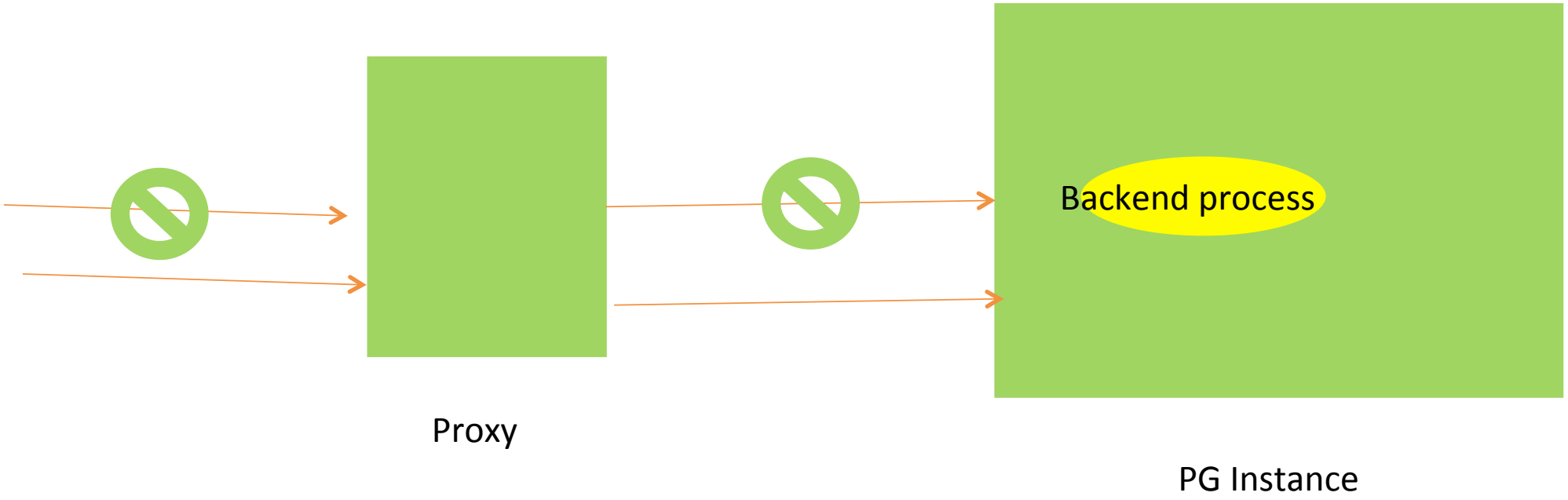
- Proxy does the connection re-establishing out of transaction boundaries
- Only for connections that have not done anything not re-creatable
  - Temporary table usage
  - Statement preparation
- Change the kernel to support a command like “set connection_user to xxx” for proxy to re-establish connection without an explicit authentication process

A transparent switch-over

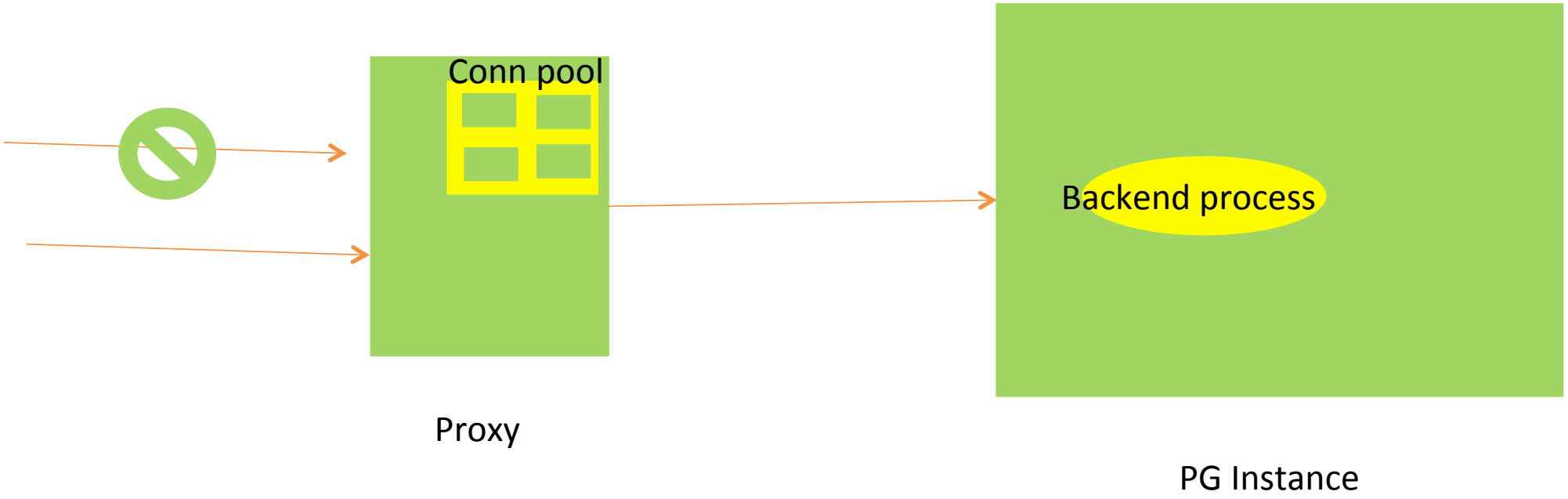


Needs for transparent connection pool

- Connection establishment is costly for PG
- Performance is bad for short-connection applications



Transparent connection pool with Proxy



Transparent connection pool with Proxy

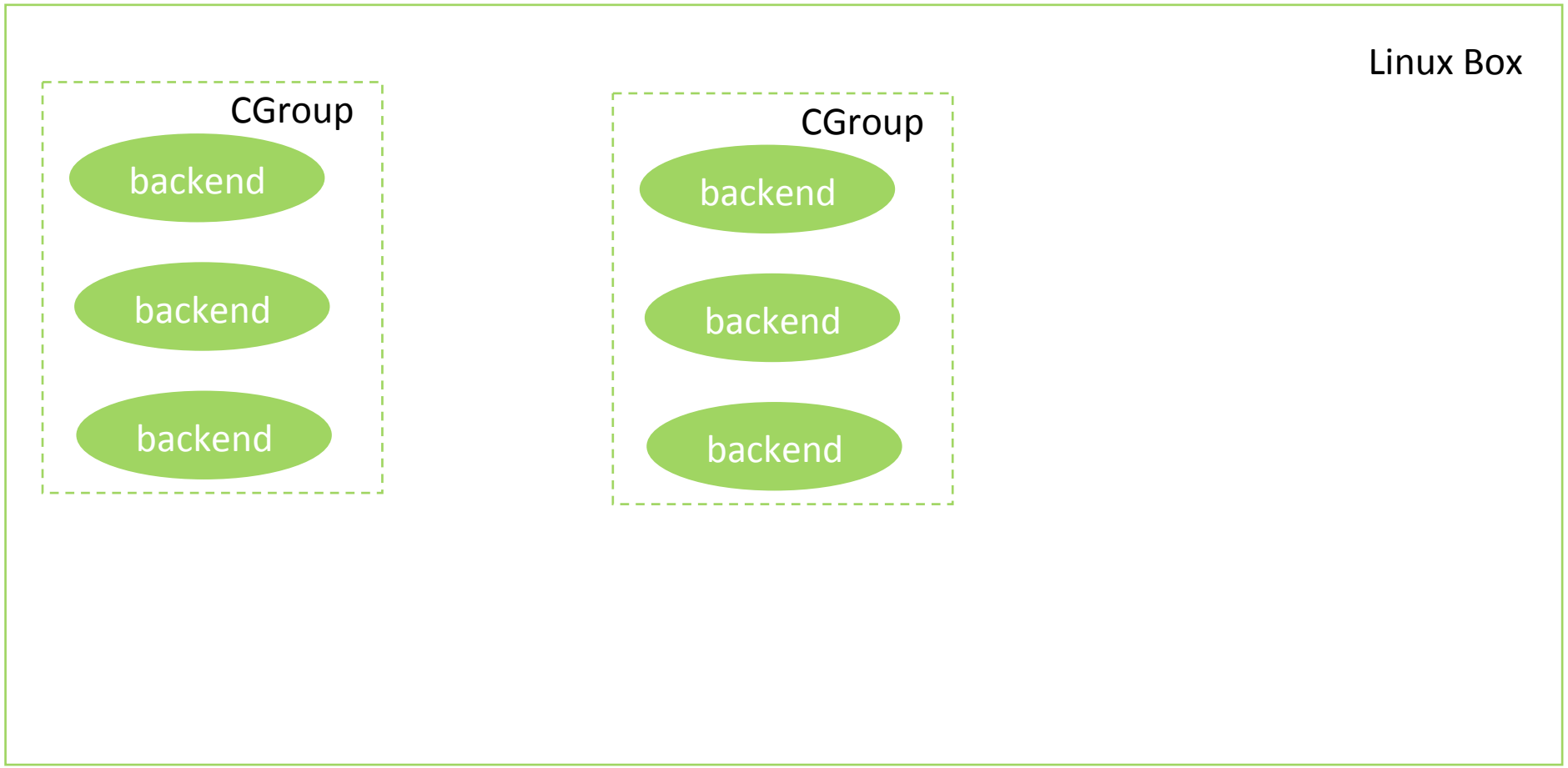
- Proxy needs to restore all the resources held by this connection before putting it into the connection pool
  - DISCARD ALL is issued for this purpose

- Authentication needs to be performed against the incoming user when connection is reused
  - Change the kernel to support “SET AUTH_REQ = ‘user/database/md5password/salt’ ” for authentication

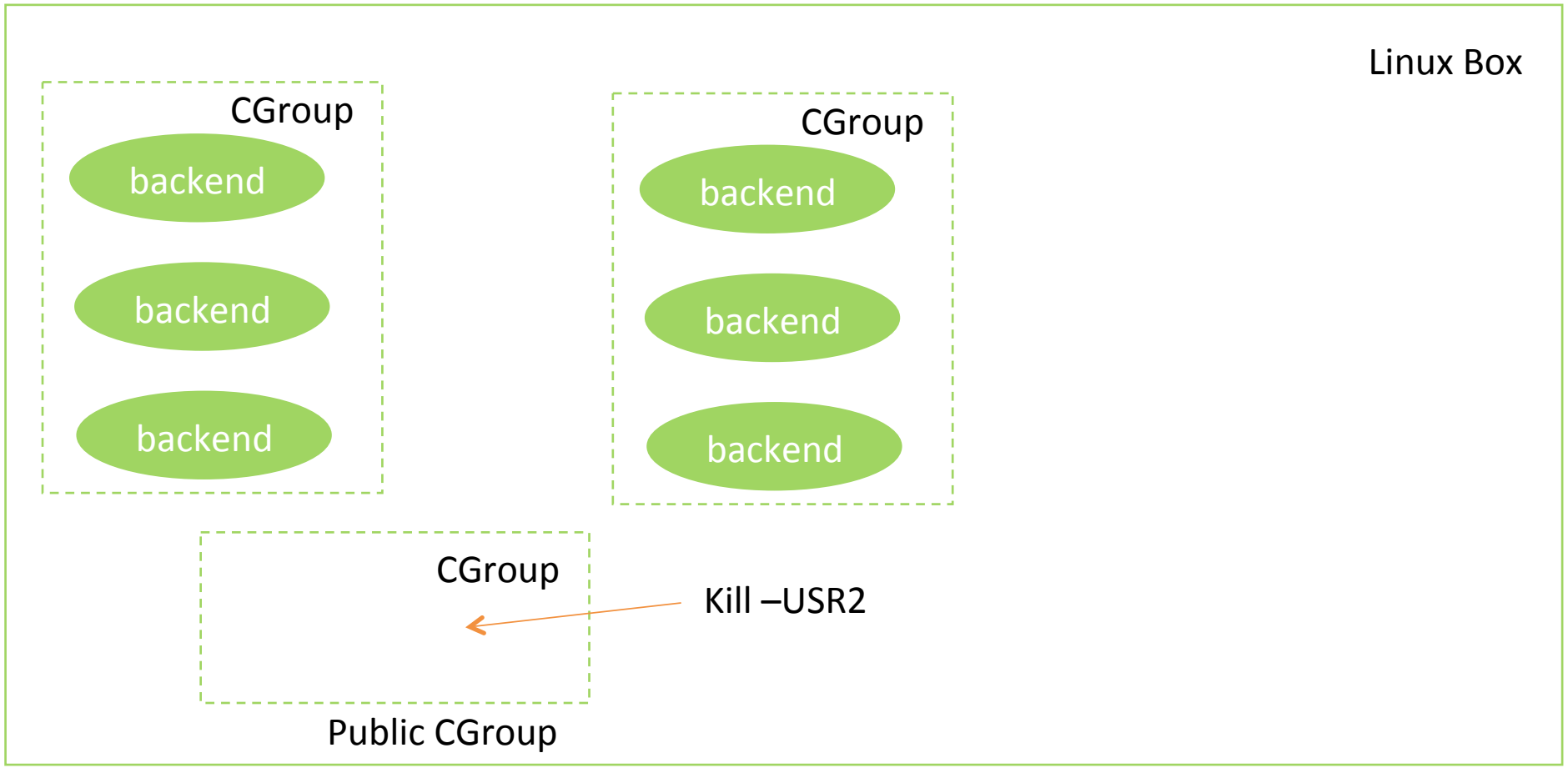
Handling OOM

- OOM (out of memory) cases were frequently observed
  - Cloud users tend to buy small class instance firstly, then regularly upgrade to higher class
  - PostGIS large objects or big json objects are widely used in some instances
  - Concurrent connections are not suitably configured by application
- When an instance goes OOM, one of its processes is picked up and killed by the Linux kernel. Postmaster process then detects the kill and restart the whole instance. All connections are then lost.
- How can we minimize OOM impact for users?

Handling OOM



Handling OOM



Handling IO Hang

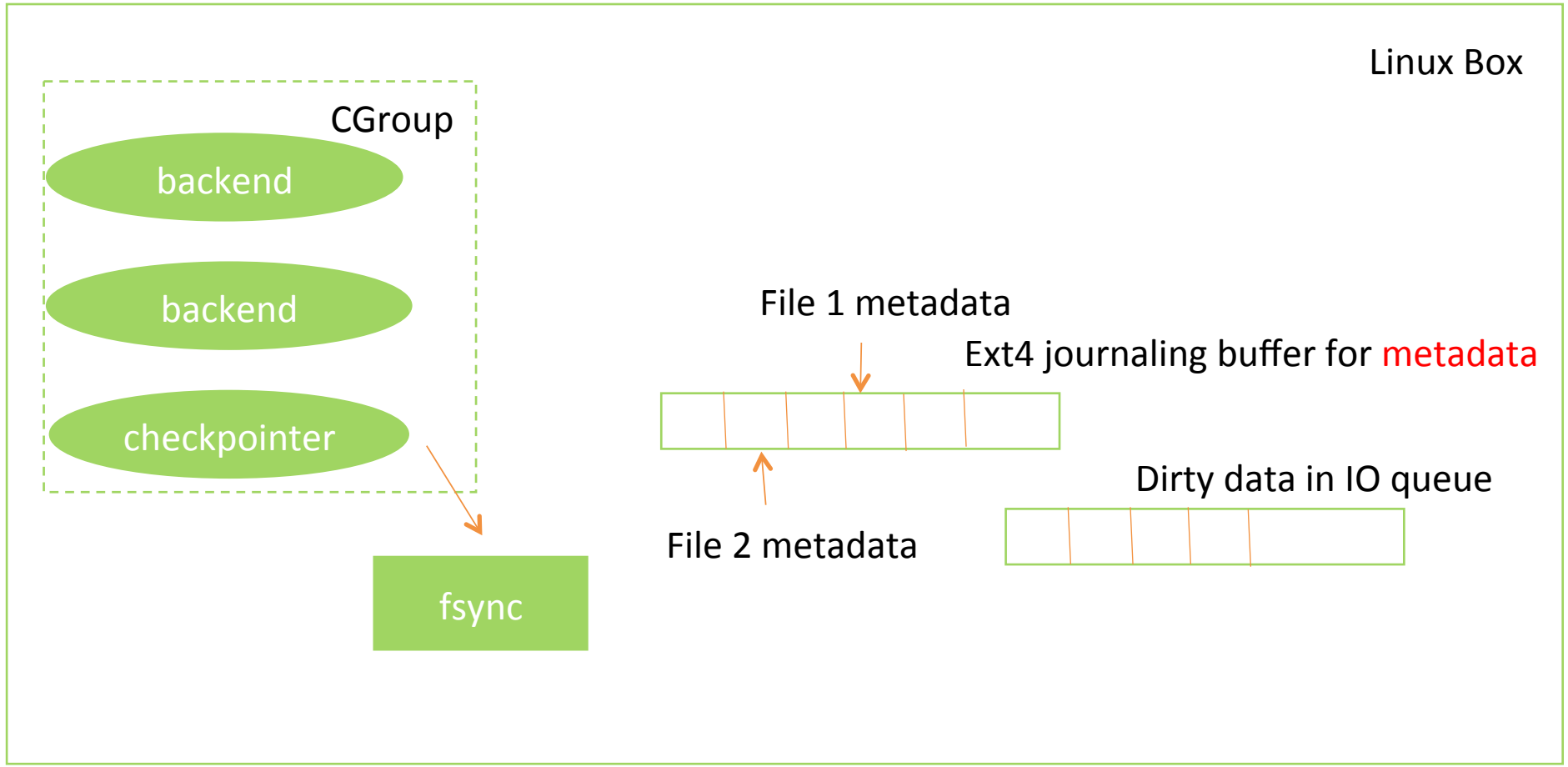
- Symbols of IO hangs (ext4 filesystem with mount –data=order on Linux)
  - Message in PG error log: “WARNING: using stale statistics instead of current ones because stats collector is not responding”
  - Long checkpoint fsync time (observed from error log when log_checkpoints set to on)
  - Nearly all operations including setup of a new connection hang for > 10s. All instances on the same machine are affected.
  - But IO usage is low (< 10%) for the problematic disk.

Handling IO Hang

- We observed such hangs are quite frequent. We found it gets worse when:
  - Many instances shares one same disk and ext4 file system in our environment
  - “create database <YYY> template <XXX>” which will cause large file copying and fsyncing if template db <XXX> is large

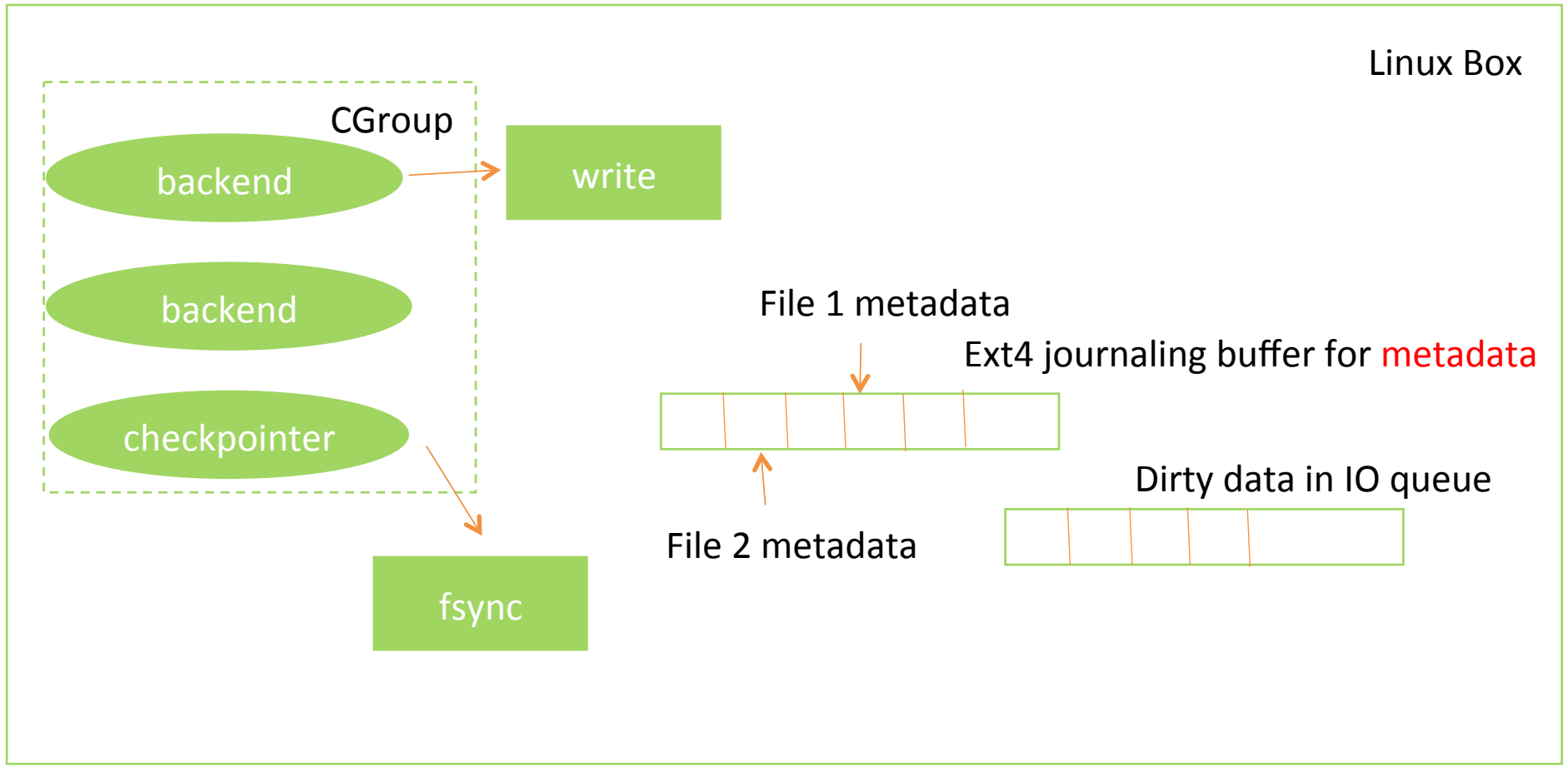
Why IO Hang?

- fsync can be slow



Why IO Hang?

- write() can be blocked by fsync()



Handling IO Hang

- Mount ext4 with option data=writeback
  - Do this only when your PG full_page_image has been set to on

- Change the kernel to call sync_file_range() before calling fsync in checkpointer process

Privilege Management

- Superuser privileges are kept from the cloud service users for security reason
- Users keep on asking for more “superuser” privileges for managing roles and data of the whole instance
- We have to loose privilege check for some cases to make users happy

Our solution for Privilege Management

- Creating a new role – rds_superuser
  - Allow this role to manage all other non-superuser’ s data
  - Allow it to set specific configuration parameters

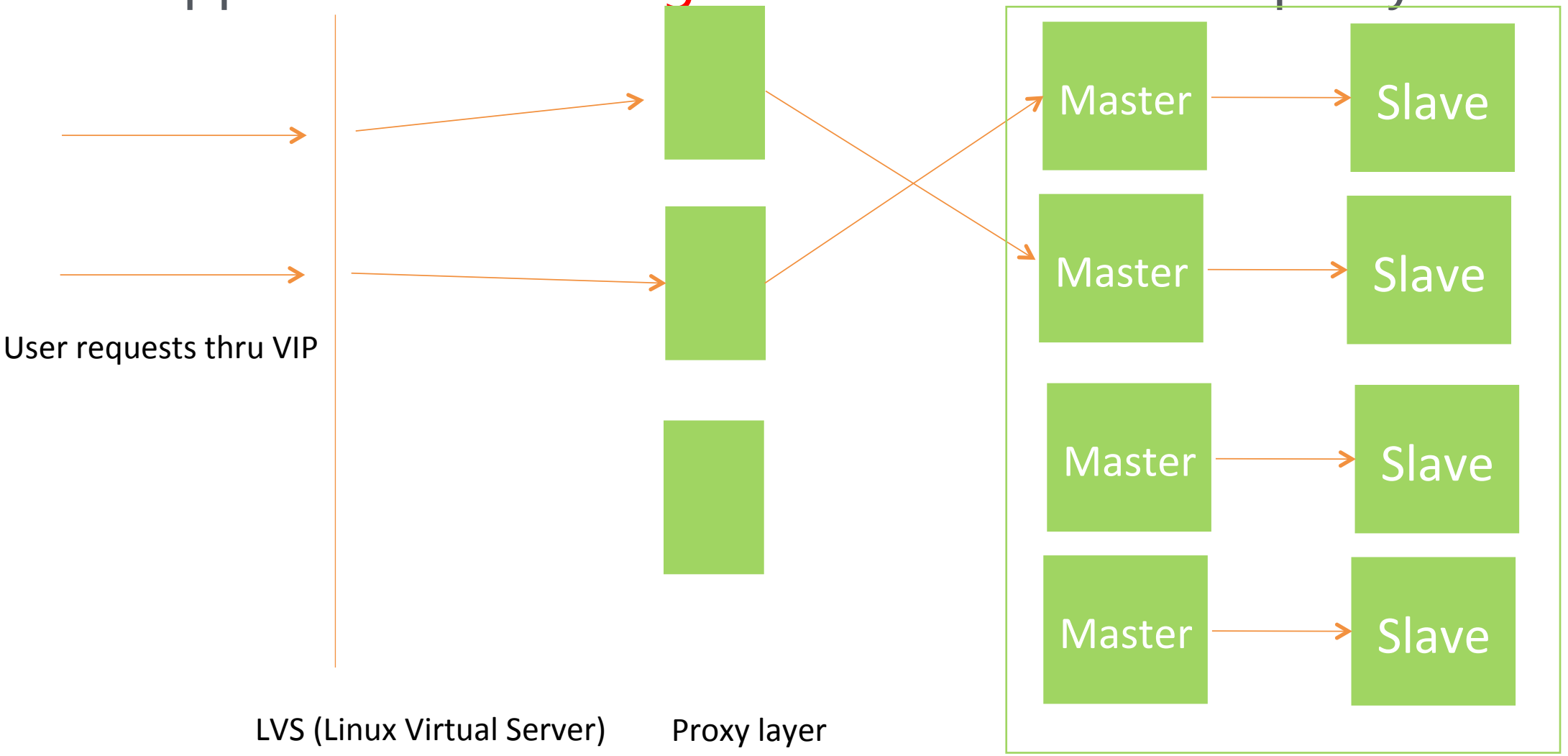
- In order not to break the catalog compatibility, we manage to reuse a flag in pg_authid to identify rds_superuser roles

Our solution for Privilege Management

- Table "pg_catalog.pg_authid"
- Column | Type | Modifiers
- ---+---+---
- rolname | name | not null
- rolsuper | boolean | not null
- rolinherit | boolean | not null
- rolcreaterole | boolean | not null
- rolcreatedb | boolean | not null
- **rolcatupdate | boolean | not null**
- rolcanlogin | boolean | not null

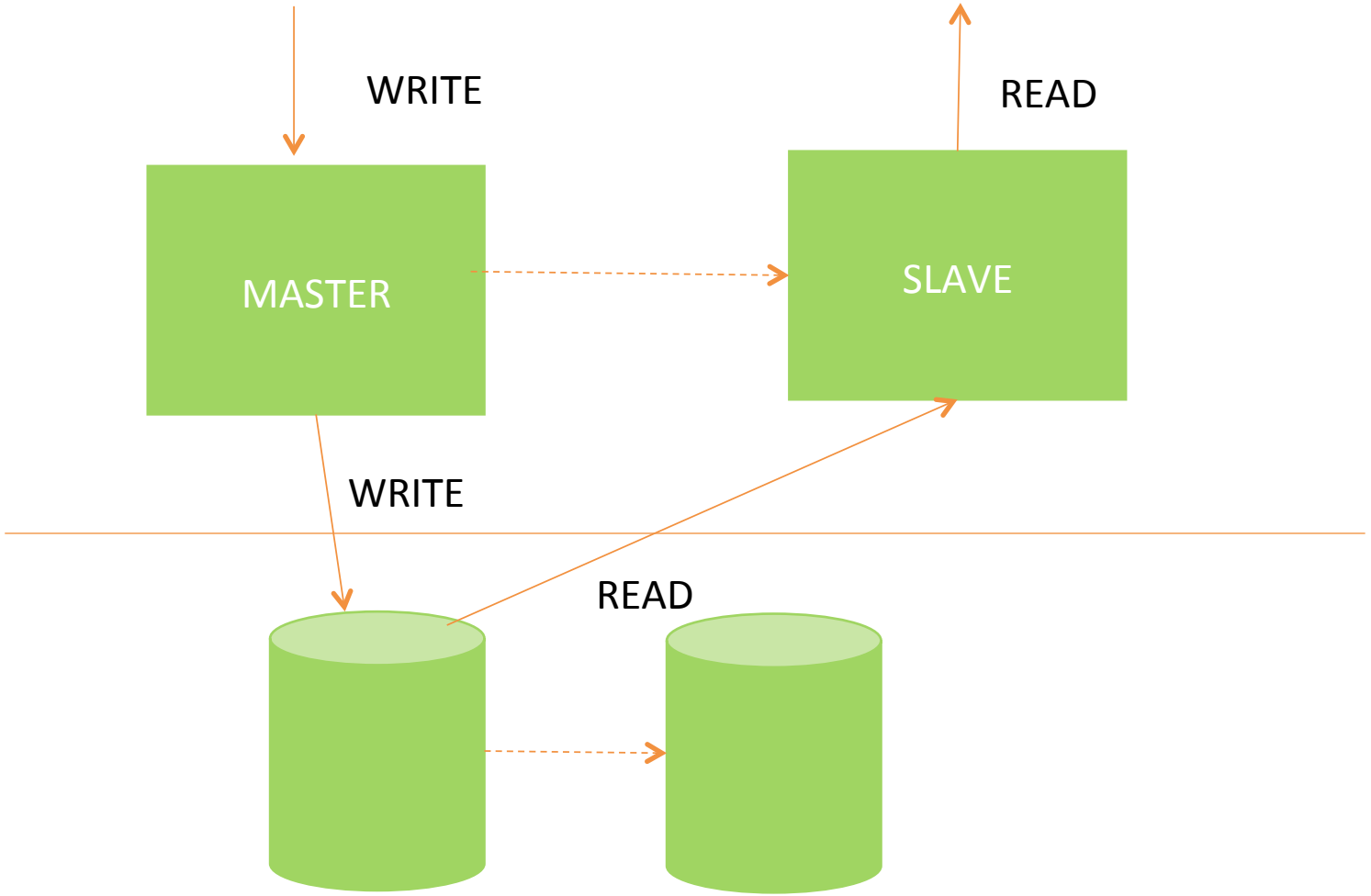
Future Directions

- Support data sharding across machines with proxy



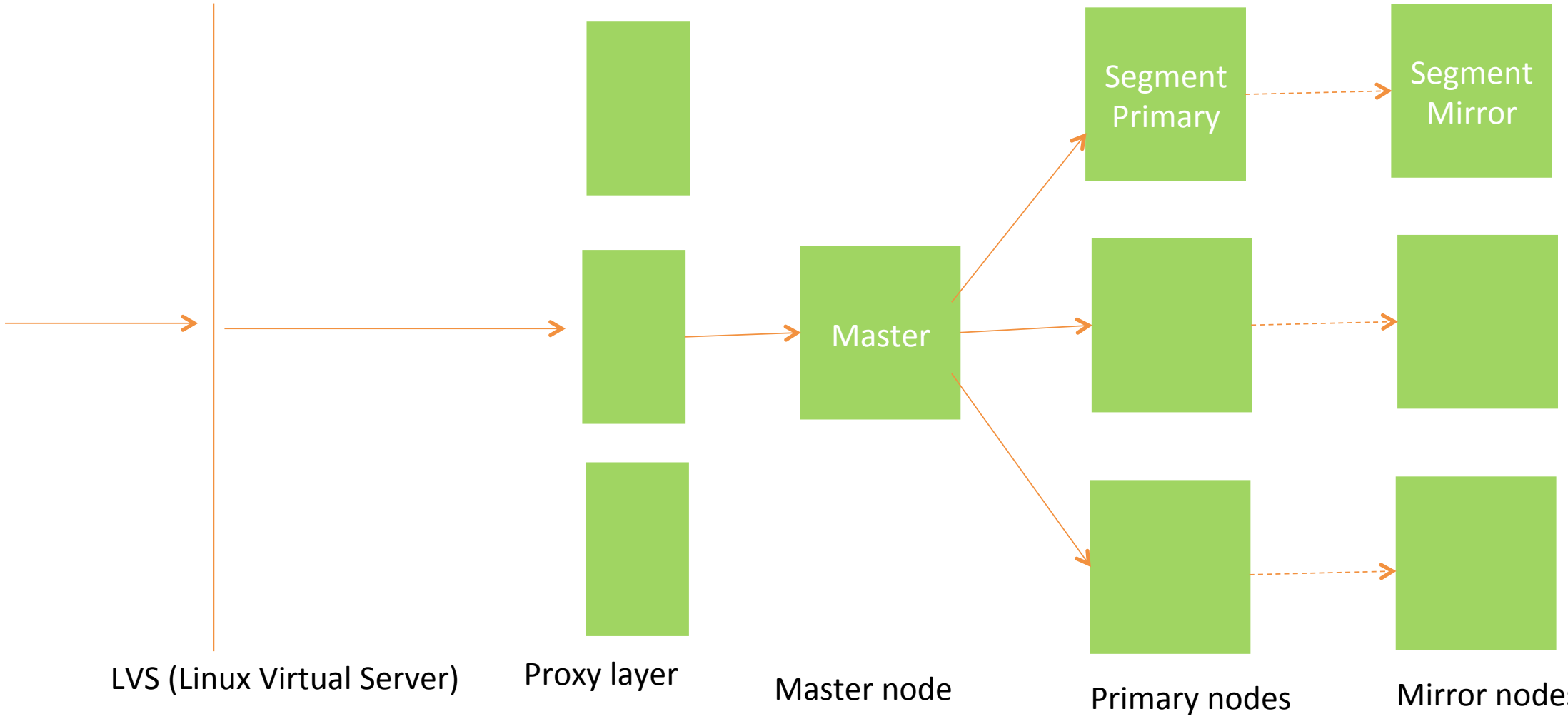
Future Directions

- Support shared-data architecture



Future Directions

- Providing Greenplum as a cloud service



Q&A

THANKS!