

Multi-Master and Multi-Source Replication

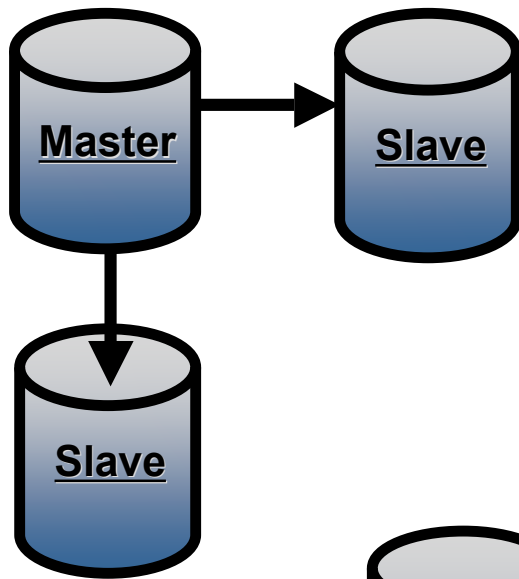


© Continuent 2011

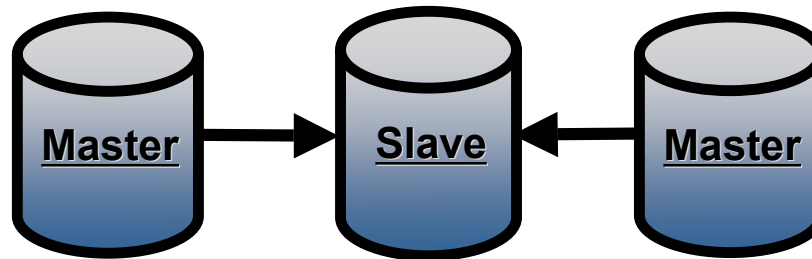
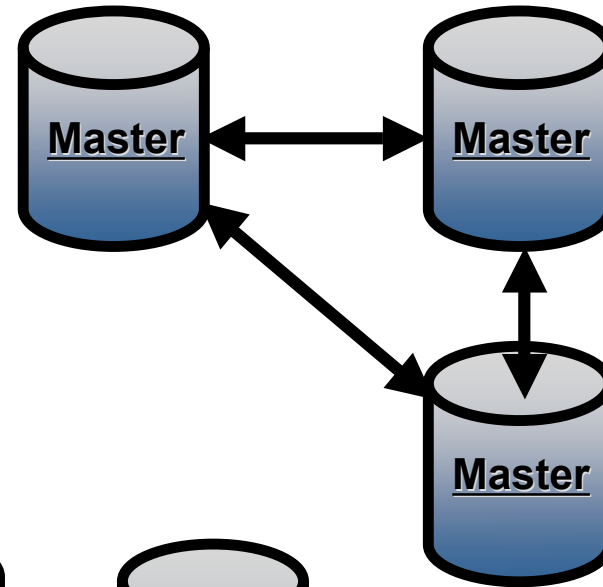
continuent
Open. Always Available.

Replication Terminology

Master/slave: One master replicates to one or more slaves



Multi-master: 2 or more masters accept updates



Multi-source: 2 or more masters replicate to a single slave (aka “fan-in”)



Multi-Source Replication



© Continuent 2011

continuent
Open. Always Available.

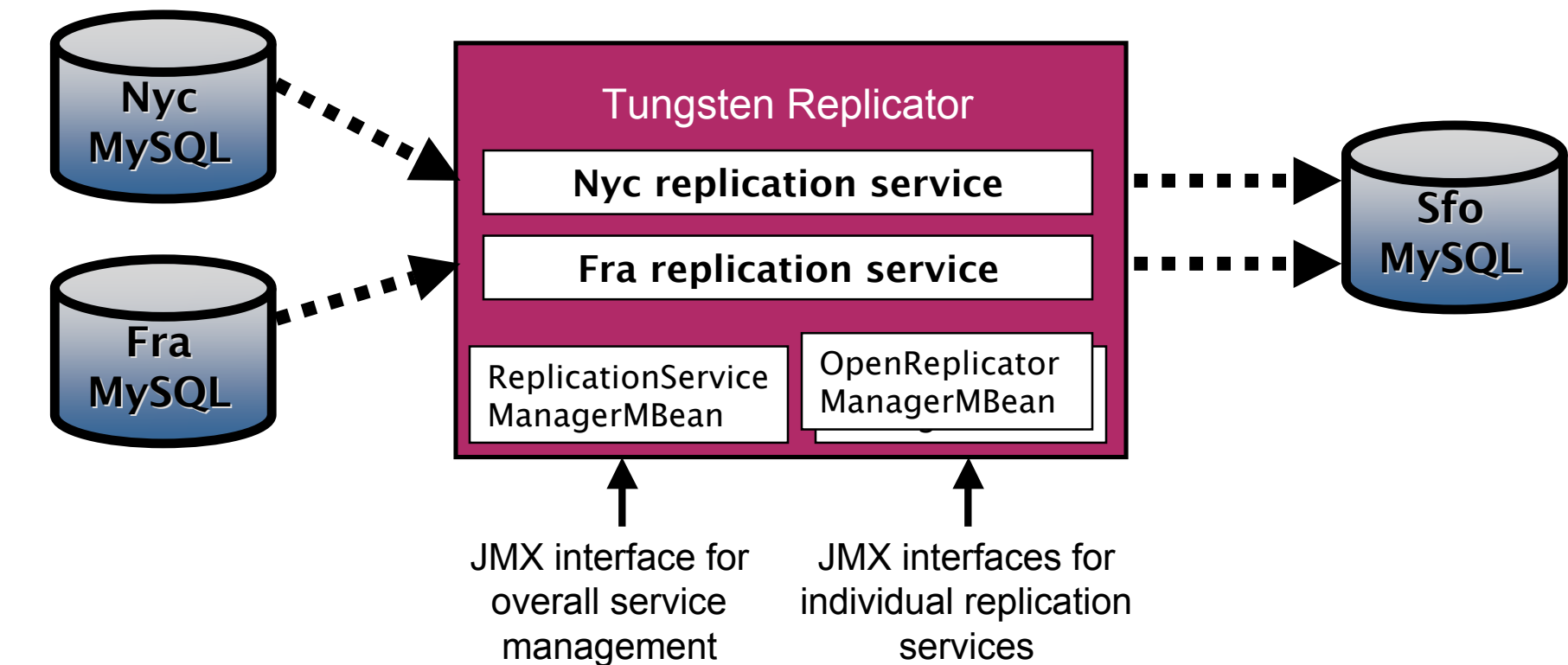
Multi-Source Replication

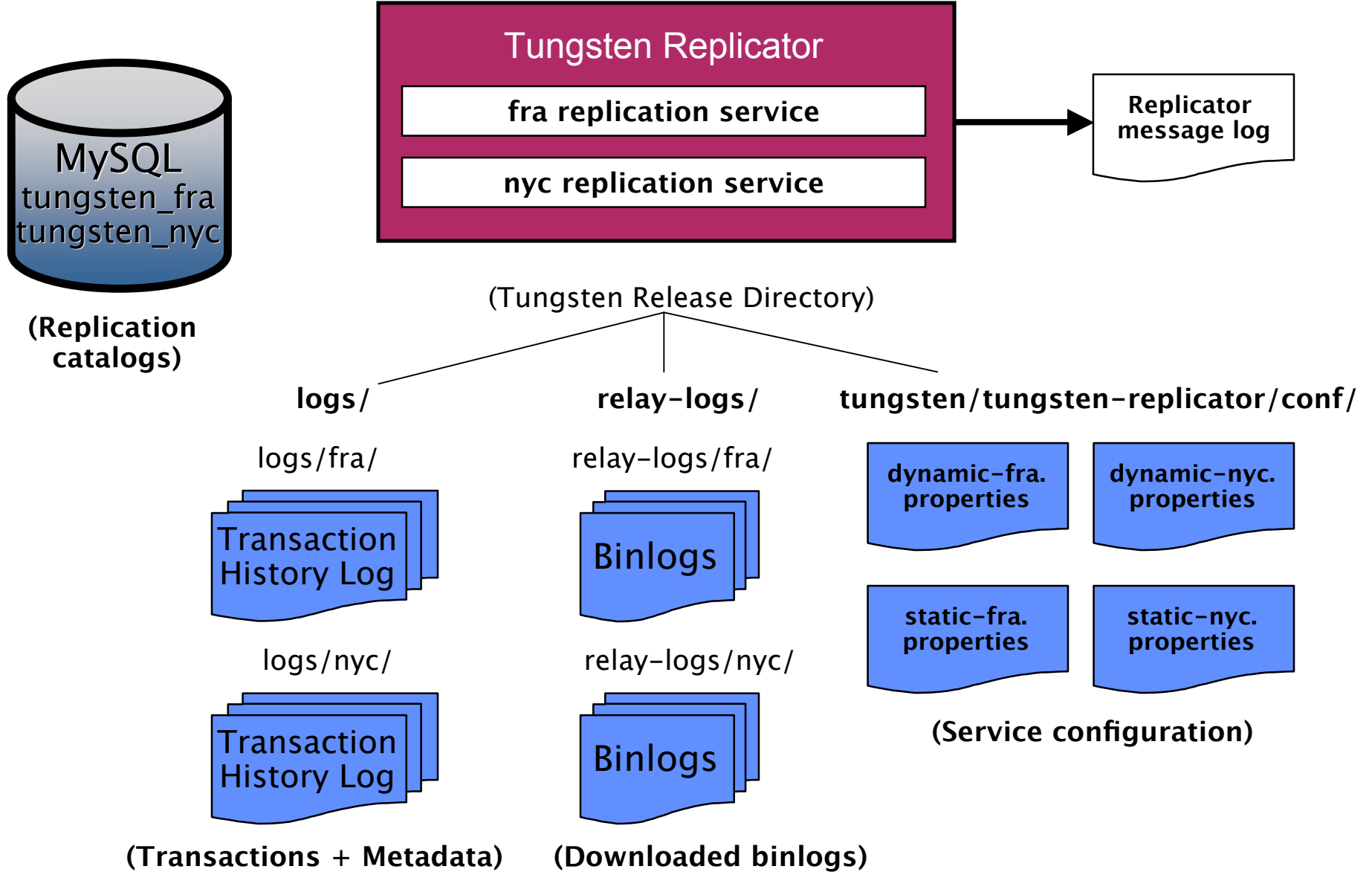
Let's do a demo!



Building Block: Replication Services

/ Tungsten allows multiple *replication services* per replicator process





Multi-Master Replication



© Continuent 2011

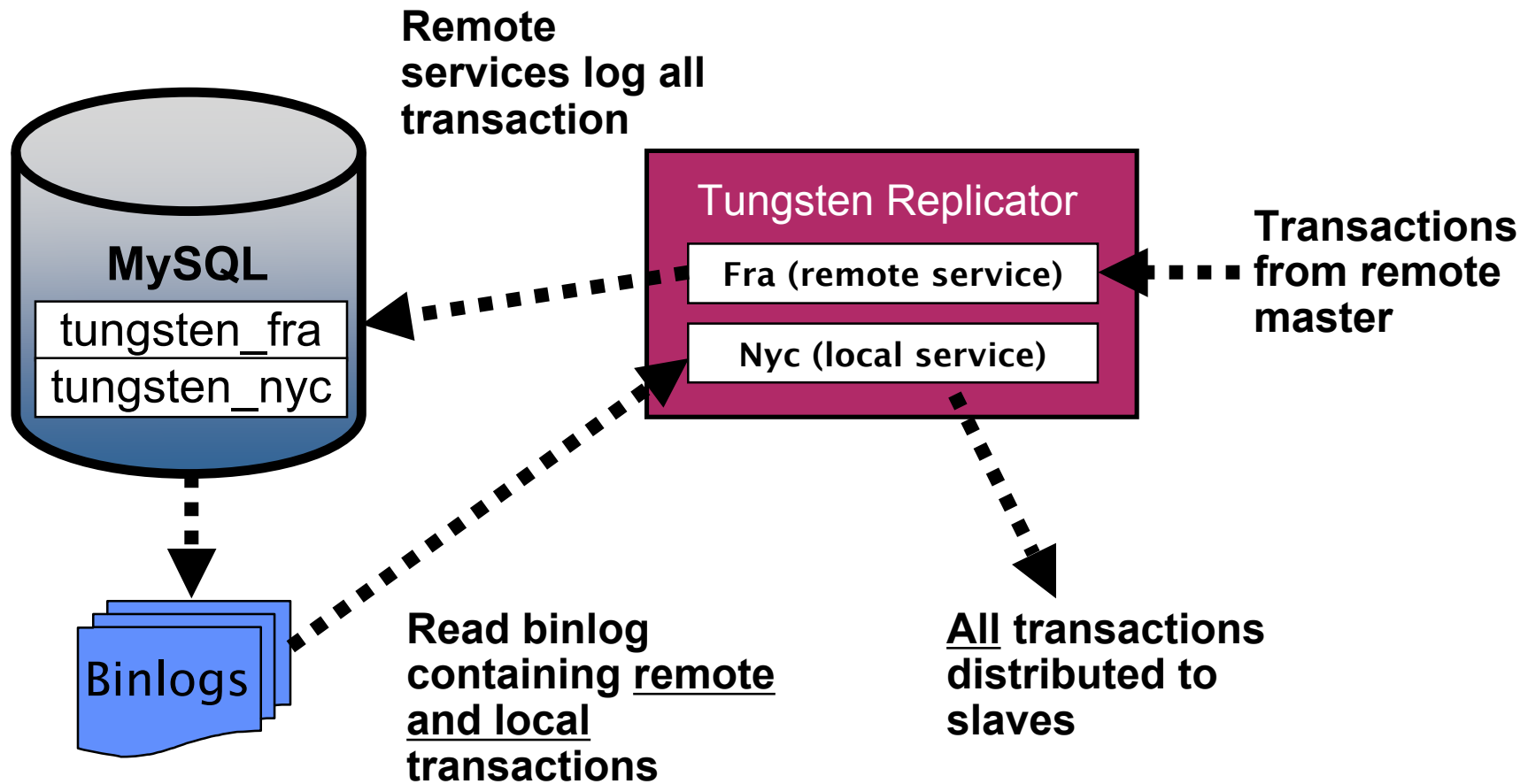
continuent
Open. Always Available.

Multi-Master Replication

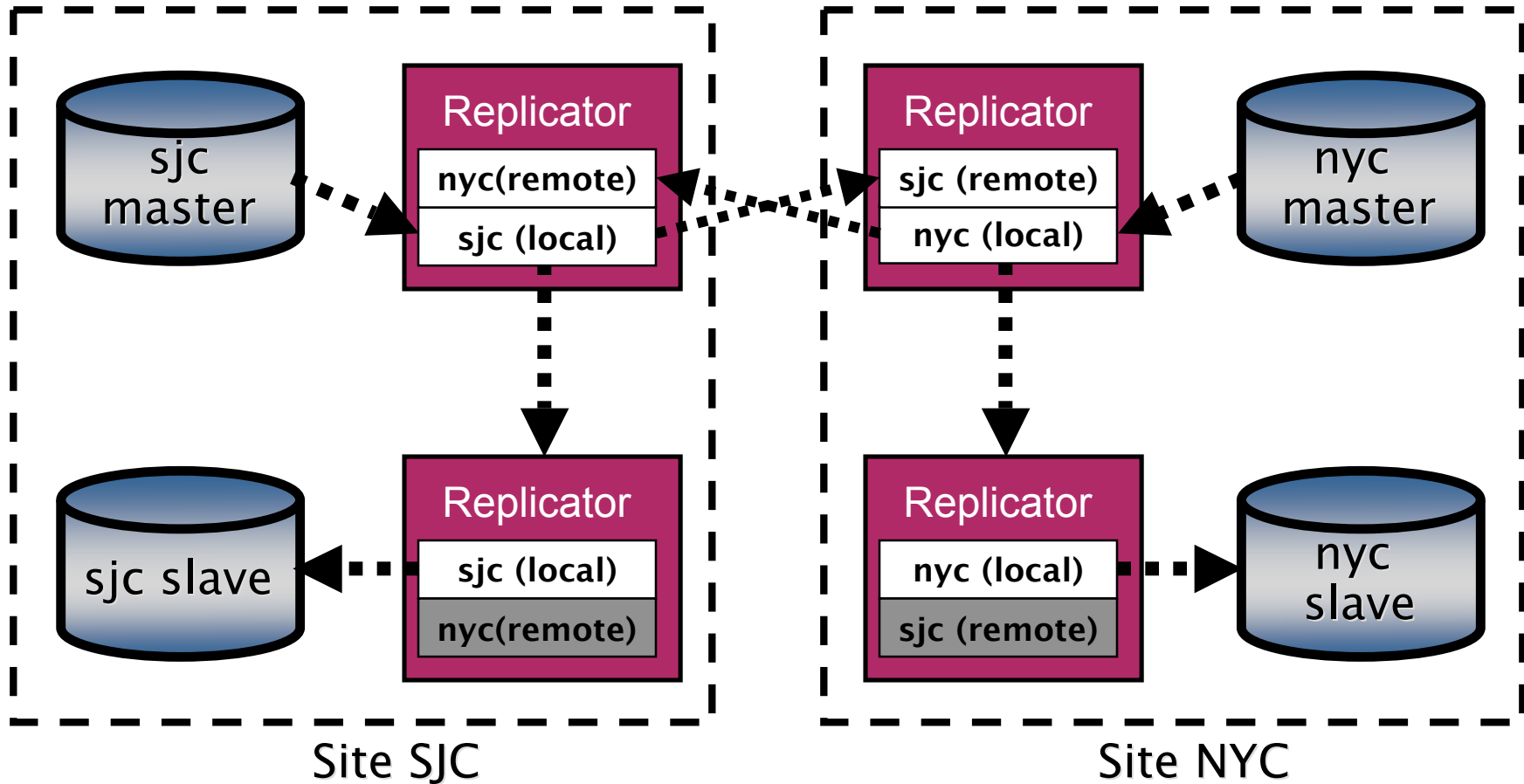
Let's do a demo!



Remote and Local Replication Services

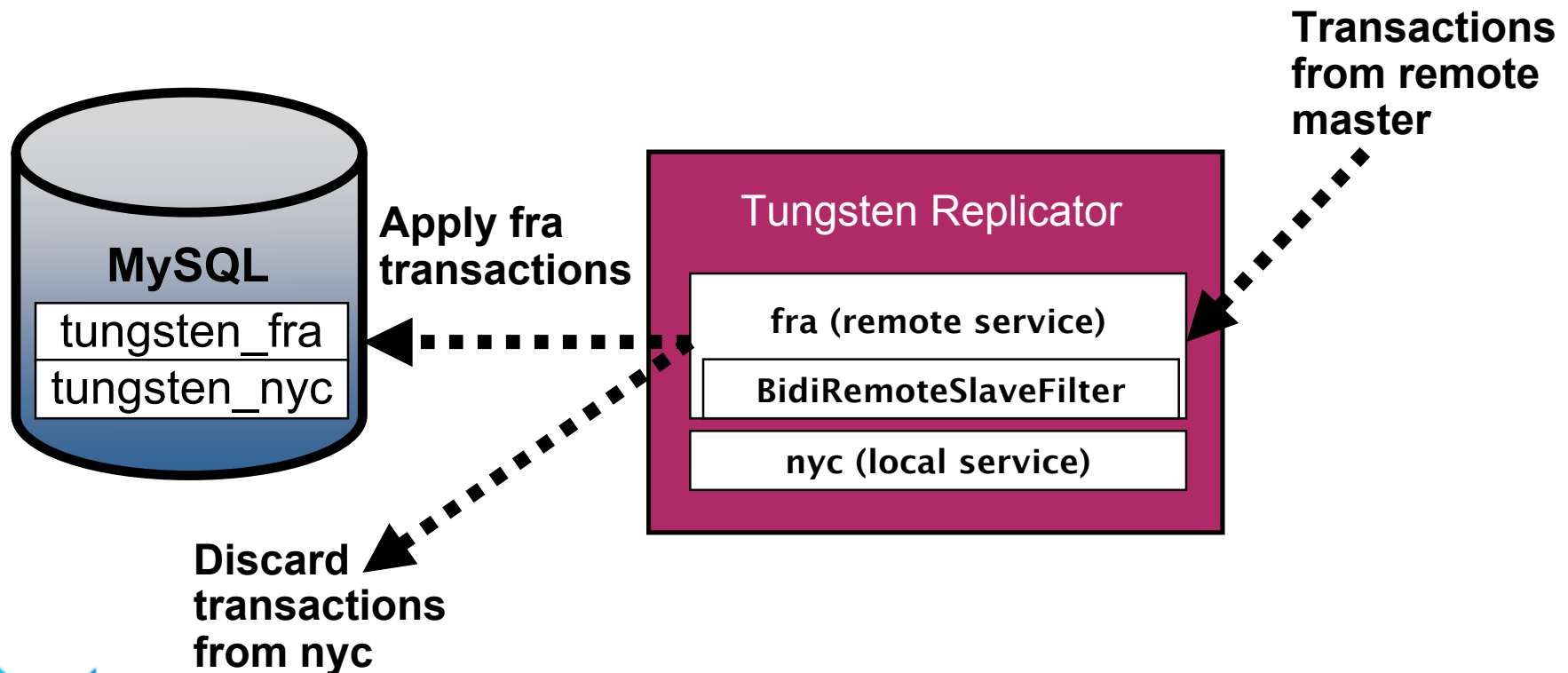


Example: Bi-Directional Replication



Preventing Replication Loops

- / Replication loop = same transaction keeps getting applied over and over
- / Specialized filter discards transactions that try to loop back



Resolving Conflicts Between Masters

/ Use key offsets in MySQL

- Use auto_increment_increment && auto_increment_offset options
- Only deals with problem of colliding keys

/ Change applications

- Use natural keys
- Avoid writing to same tables/databases
- Can avoid all conflicts but requires discipline

/ Use Tungsten filters

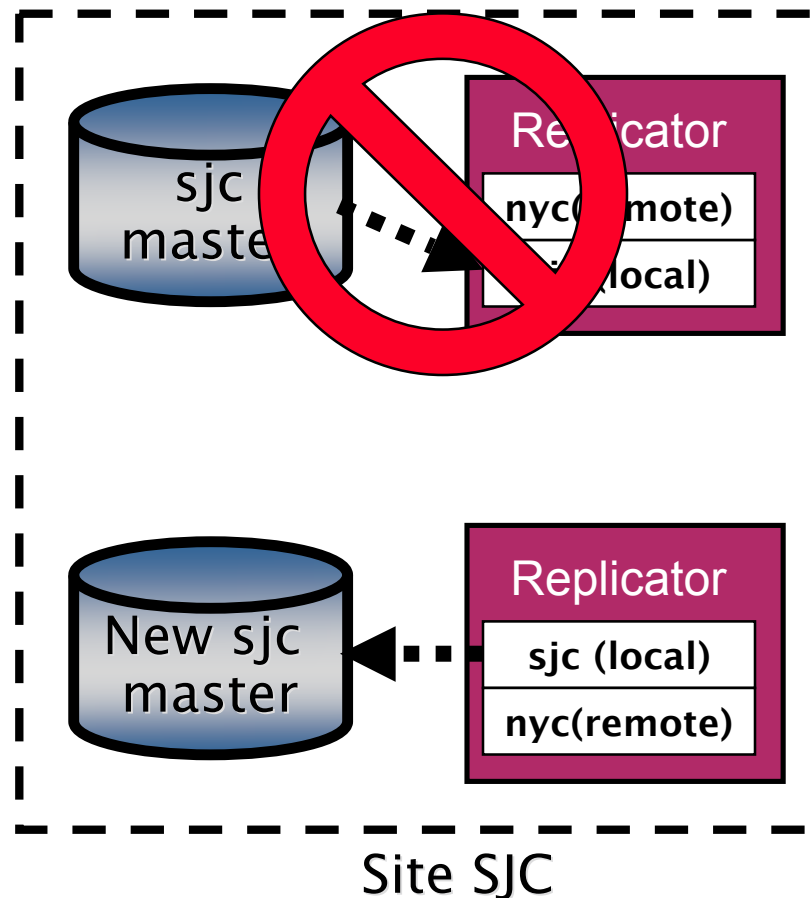
- Tungsten anti-looping filter resolves conflict by avoiding it
- Write filters to assign “ownership” of tables to specific services
- Write filters to pick transactions based on application criteria



Failover and Multi-Master Replication

/ Tungsten supports failover of master to a backup slave

- Key feature: Remote service position is copied into slaves



1. Promote slave sjc service to master
2. Start nyc remote service on new master



Handy Commands for Monitoring

`trepctl services`

/ Show status of each replication service

/ Look for failed remote services

/ Look for replication loops

`trepctl -service <svc> status`

/ Show status of individual services



Tungsten Multi-Master Best Cases

/ **Cross-site distribution of global data**

- Independent master on each site
- Slave(s) on each master for local failover

/ **Multi-Source Replication**

- Merging data from two or more masters into one slave

/ **Replacement for MySQL circular replication**

- Adding and removing nodes does not require full stop
- Simpler failover semantics

These are all cases MySQL replication does not handle well



Multi-Master Worst Cases

/ **Write scaling**

- Multi-master does not scale writes!

/ **Single-site multi-master**

- Master/slave is simpler and has better performance

/ **Multi-master on same tables**

- Applications must be designed to avoid conflicts
- Recovery from errors is complex
- You may need synchronous multi-master to avoid data consistency problems



Multi-Master/Multi-Source Roadmap

- / **Fix bugs**
- / **Eliminate need for multiple remote services**
- / **Conflict resolution (start small, work up)**
- / **And much more...**

