



# MySQL Proxy meets: binlogs

Jan Kneschke  
MySQL Enterprise Tools  
mailto: [jan@mysql.com](mailto:jan@mysql.com)

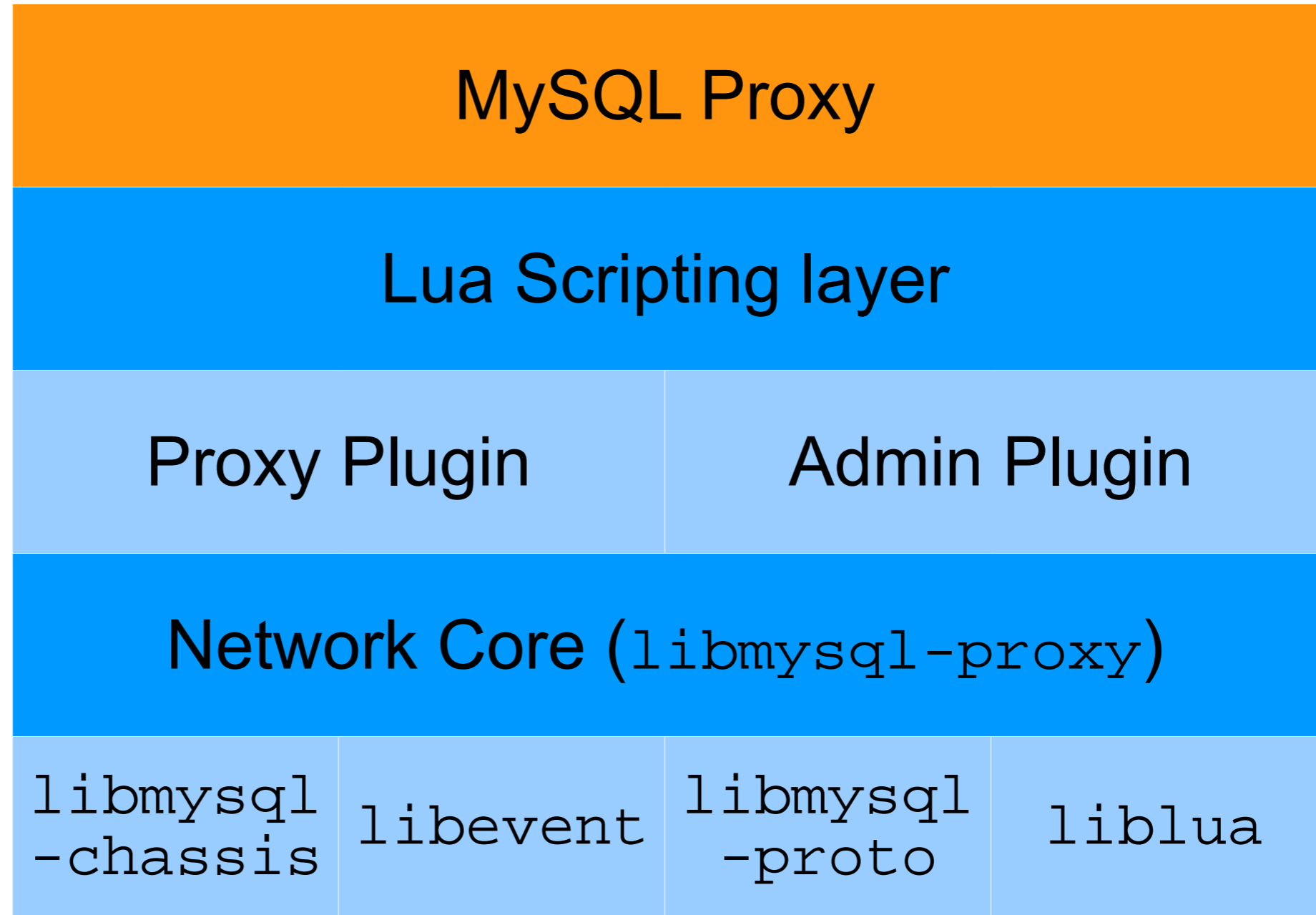
Innovation Everywhere

# What is MySQL Proxy



- Started Feb 2007
- Current: MySQL Proxy 0.7.0
- Source available on [launchpad.net](http://launchpad.net)  
`$ bzr branch lp:mysql-proxy`
- Foundation of the MEM 2.0 Agent
- Fully support for MySQL Enterprise Customers
- 2 (+ 1) Developers, 1 QA

# What MySQL Proxy really is



# The pieces - chassis

- Command Line Interface
- Config-File support
- Logging (event-log, syslog, stderr, logfiles)
- Plugin Loading



# The pieces – libmysql-proxy

- Protocol live-cycle (auth, old-auth, query)
- Provides the plugin hooks
- Global Lua scope
- Network interface
- MySQL Protocol encoding/decoding
  - `libmysql-proto`



# Protocols

- Protocols are stacked  
( Looks nice, but isn't )
- MySQL Client Protocol
  - Binary Logs
    - MyISAM Row Format for RBR



# Proof of Concept

- `mysql-binlog-dump`
  - Dumps binlogs
  - First implementation of RBR decoding
- `mysql-myisam-dump`
  - Decodes `.frm` and `.myd` files
  - Uncoveres “screens”



# Lua iterator for binlogs

- Expose binlog events into lua
- Reading and writing

```
local binlog = require("mysql.binlog")
local f = binlog.open("jan-binlog.log")
for event in f:next() do
    if event.type == "QUERY_EVENT" then
        print(event.query.query)
    end
end
end
```

# Filtering

- Filter statements from a binlog
- Remove or rewrite Queries

```
INSERT INTO answers  
VALUES ( 42, "the Answer" )
```

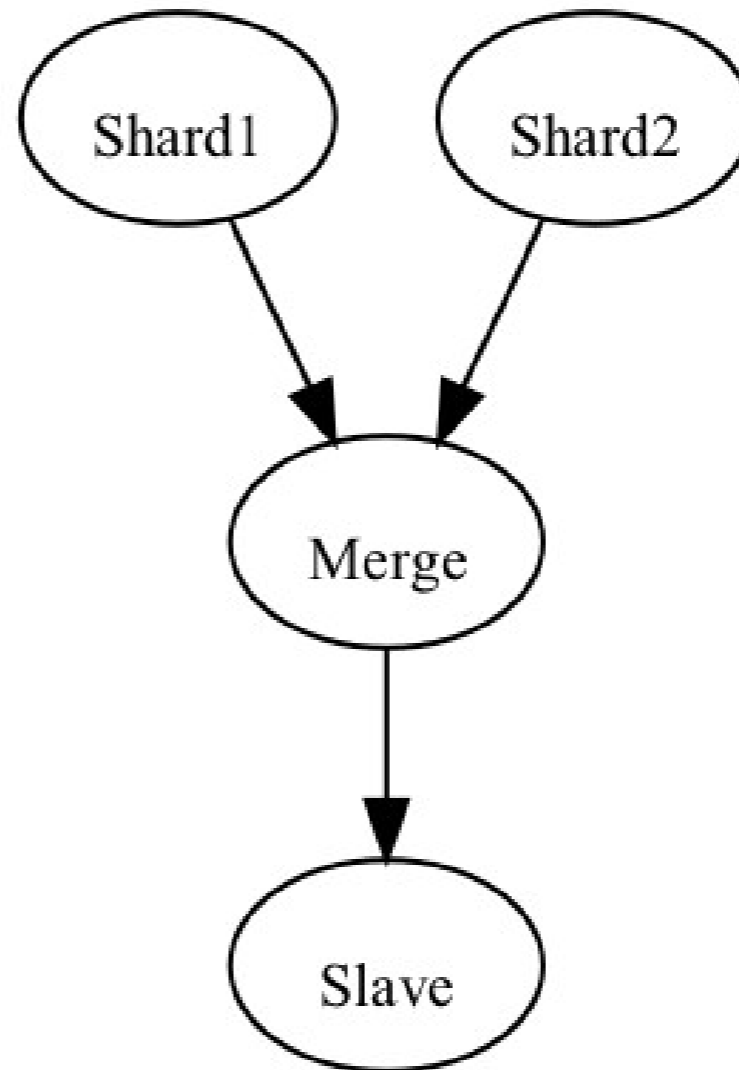
```
UPDATE answers  
SET answer = "Thanks for the Fish"  
WHERE id = 2
```

```
ALTER TABLE answers  
ADD INDEX (answer) ;
```

```
DROP TABLE answers ;
```

# Merging binlogs

- independent Masters like shards
- Merge on timestamp



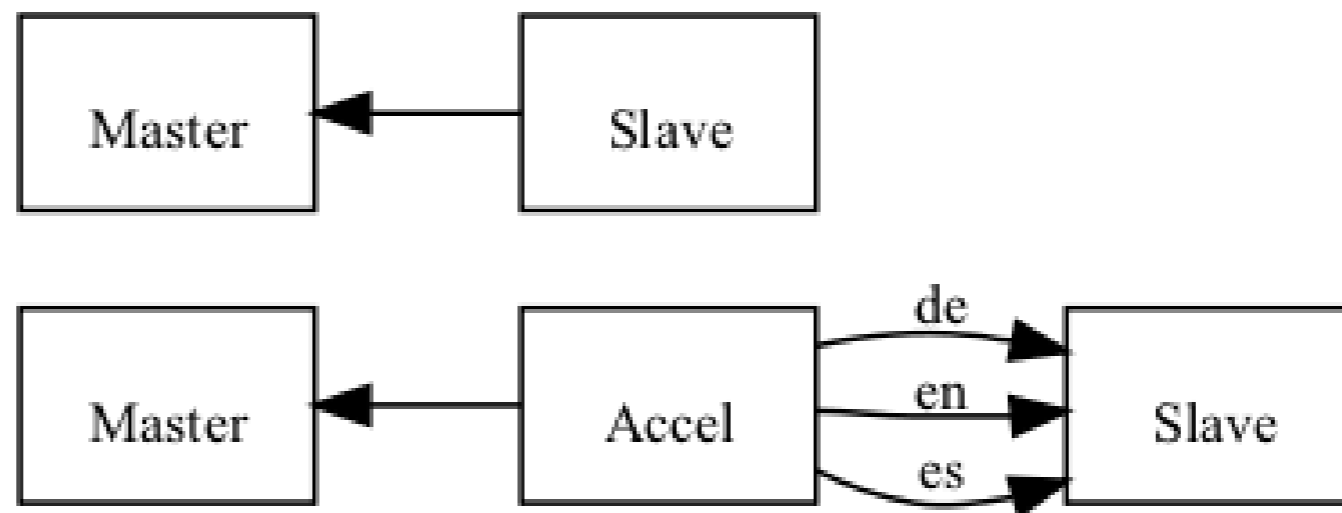
# Druckbetankung



- Use RBR to push data into a server
- RBR is already in the native format
- `mysql-data-source`
  - Convert input data to a live RBR-binlog-stream
- `mysqld`
  - `CHANGE MASTER TO ... our data-source`
  - `START SLAVE`

# Slave Accelerator

- Problem: Replication is synchronized, single thread
- Solution: move replication out of the slave
  - Don't pull, push
  - Parallel connections



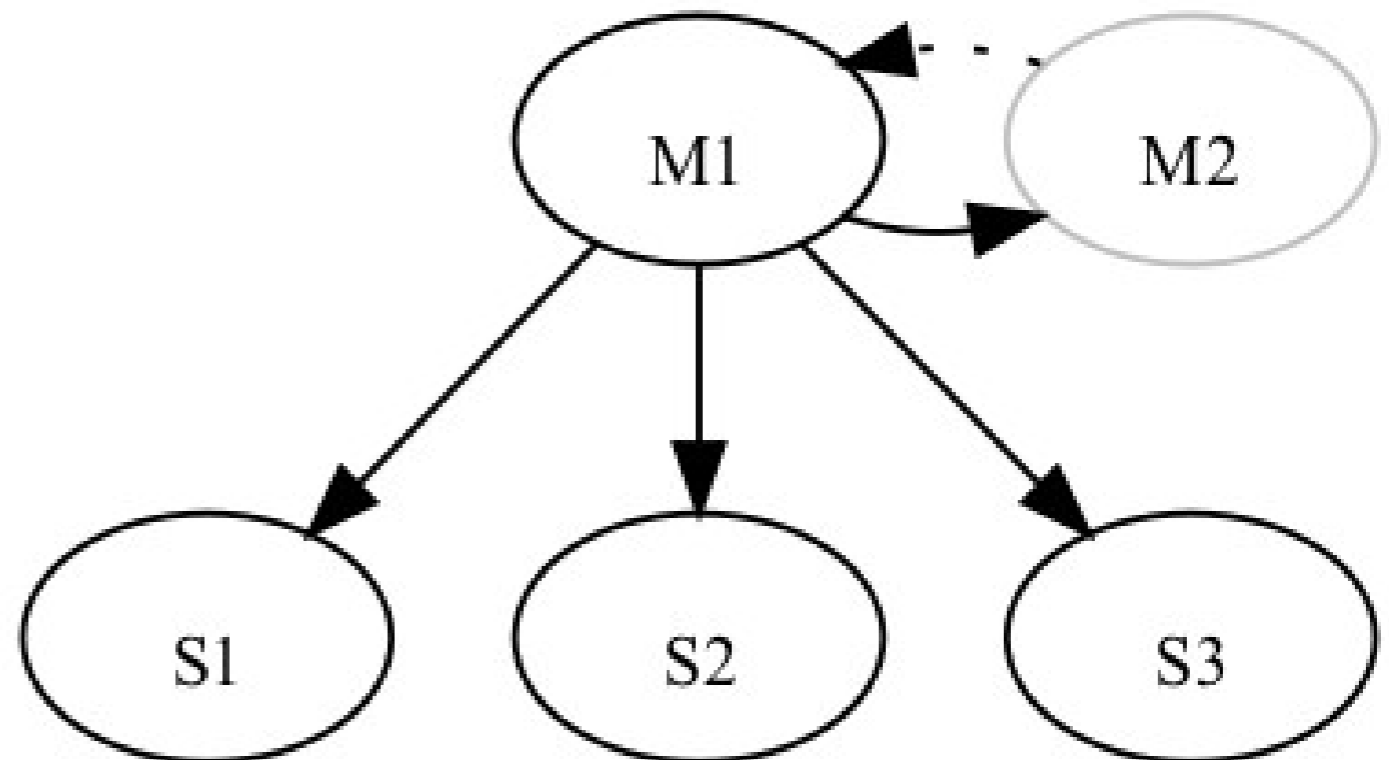
# mysql-slave-accel

- Read data from master
- Maintain slave-state centrally
- Rules to detect if data is independent
  - e.g. one connection per schema
- Works transparently for SBR, tricky for RBR

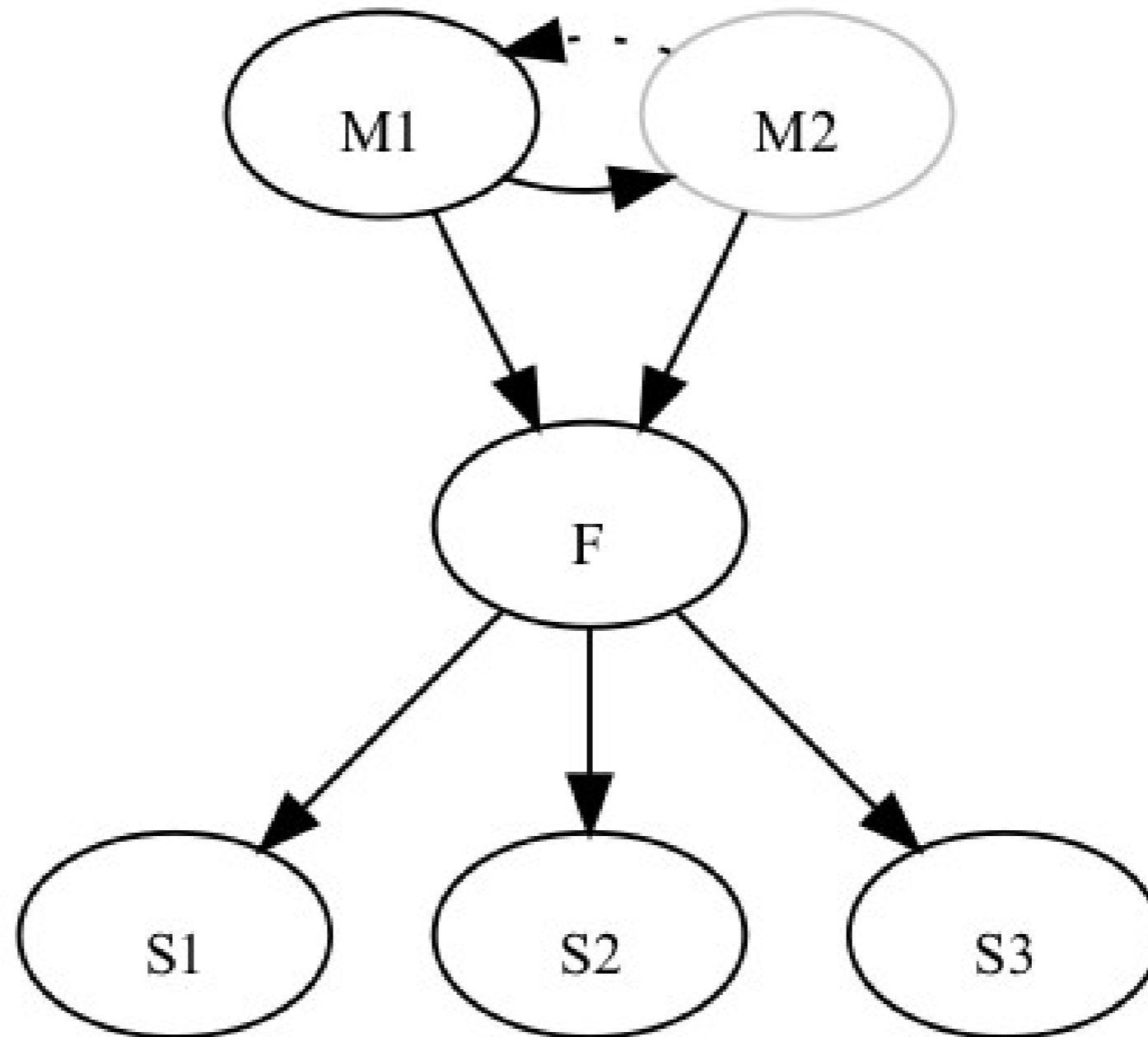


# Master Failover - Classic

- Master on stand-by
- On failure, **CHANGE MASTER** on all slaves
- Promote the *best* slave



# Automatic Master Failover



# mysql-master-failover



- Track active and passive master
- Only forward data received from both masters
- Track matching binlog-file and -position
- Active dies
  - Trigger switch over
  - Slaves get a continuous binlog stream
- States
  - Synched, Single, Out-of-Sync

# Thanks



- Code is on [launchpad.net](http://launchpad.net)
- <http://launchpad.net/mysql-proxy/>
- Questions to: [jan@mysql.com](mailto:jan@mysql.com)
- BoF
- MySQL Proxy: What's Started, What's Done and What's Next?"  
Tue 20:30