



# MySQL Proxy

## From architecture to implementation

**Giuseppe Maxia**  
MySQL Community Team Lead  
Sun Microsystems

**Ronald Bradford**  
Independent consultant

# Raise your hand

# Raise your hand

- Who has heard of MySQL Proxy ?

## Raise your hand

- Who has heard of MySQL Proxy ?
- Who has tried out MySQL Proxy ?

## Raise your hand

- Who has heard of MySQL Proxy ?
- Who has tried out MySQL Proxy ?
- Who runs MySQL Proxy in production ?

## Raise your hand

- Who has heard of MySQL Proxy ?
- Who has tried out MySQL Proxy ?
- Who runs MySQL Proxy in production ?
- Who wrote scripts for MySQL Proxy ?

## Raise your hand

- Who has heard of MySQL Proxy ?
- Who has tried out MySQL Proxy ?
- Who runs MySQL Proxy in production ?
- Who wrote scripts for MySQL Proxy ?
- Who runs MySQL Proxy from the source tree ?

## Raise your hand

- Who has heard of MySQL Proxy ?
- Who has tried out MySQL Proxy ?
- Who runs MySQL Proxy in production ?
- Who wrote scripts for MySQL Proxy ?
- Who runs MySQL Proxy from the source tree ?
- Who runs another MySQL Proxy like application ?



# Agenda

- Proxy concepts
- MySQL Proxy architecture
- Proxy with a single backend
- Uses in the field
- Proxy with multiple backends

# Solving database problems



- **broken?**
- **missing feature?**
- **not flexible?**

# Solving database problems - traditional way

# Solving database problems - traditional way

- I. file a bug report

# Solving database problems - traditional way

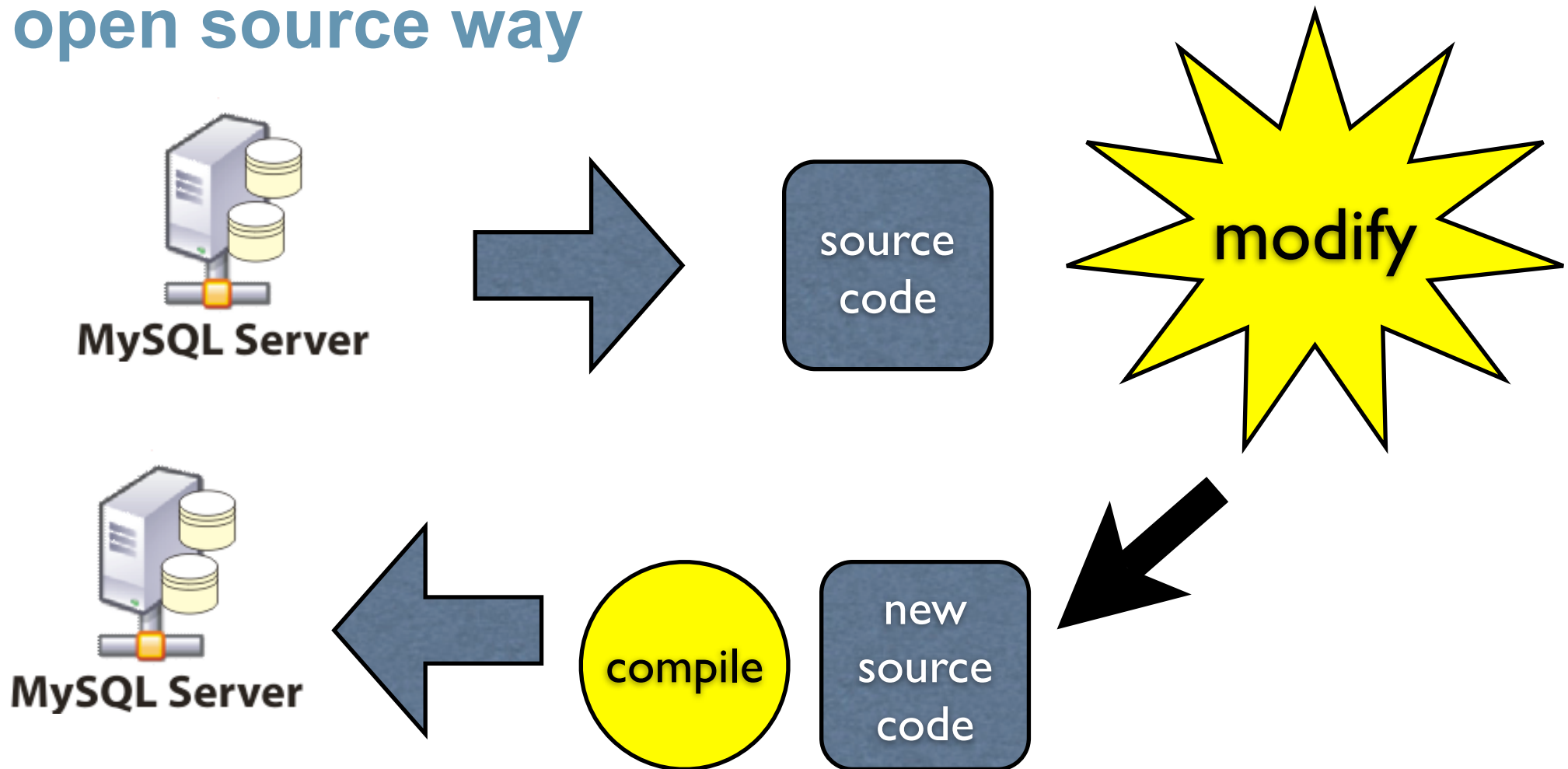
1. file a bug report
2. wait

# Solving database problems - traditional way

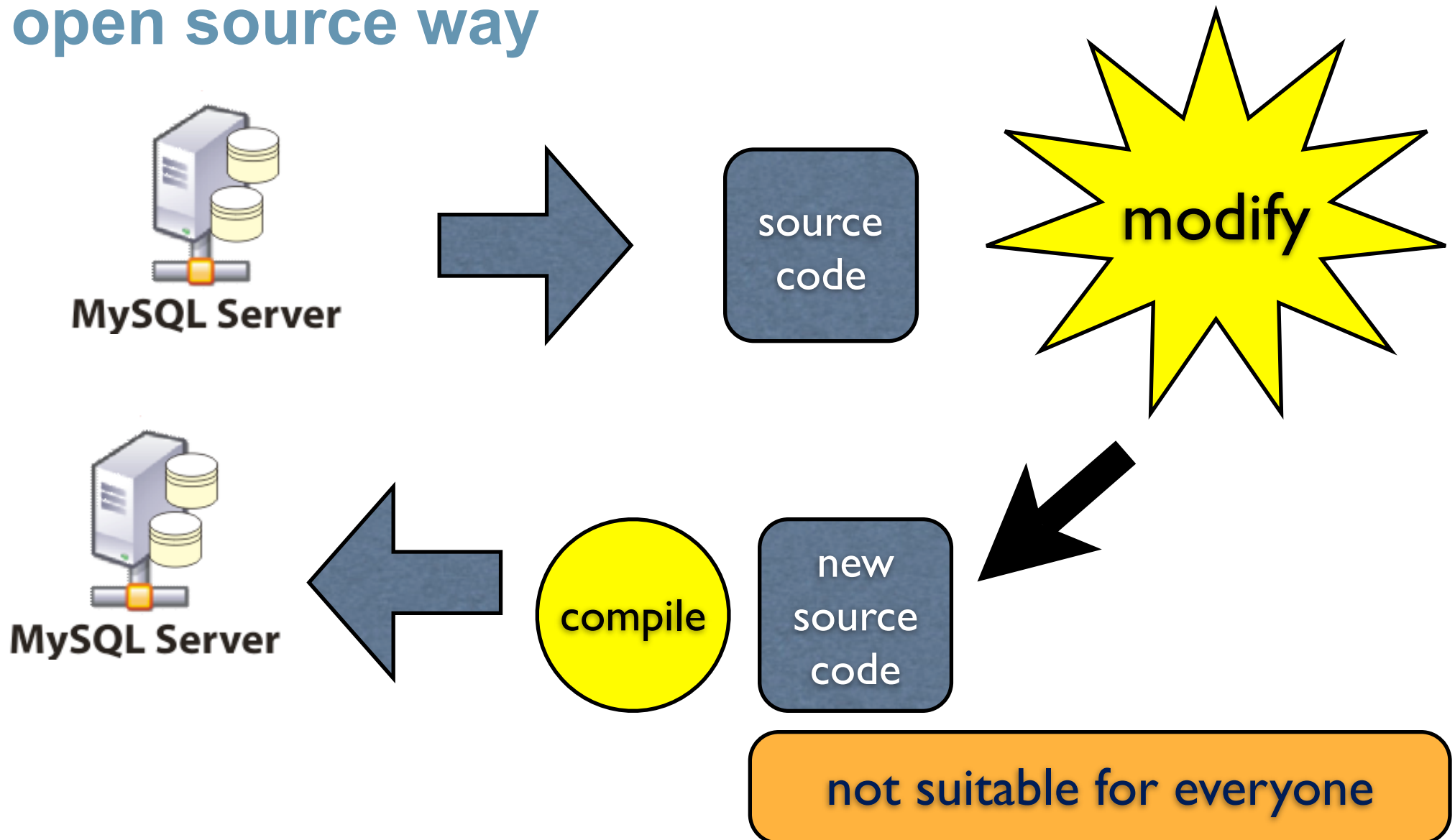


1. file a bug report
2. wait

# Solving database problems open source way



# Solving database problems open source way





# Solving database problems

## Creative (shortsighted) way

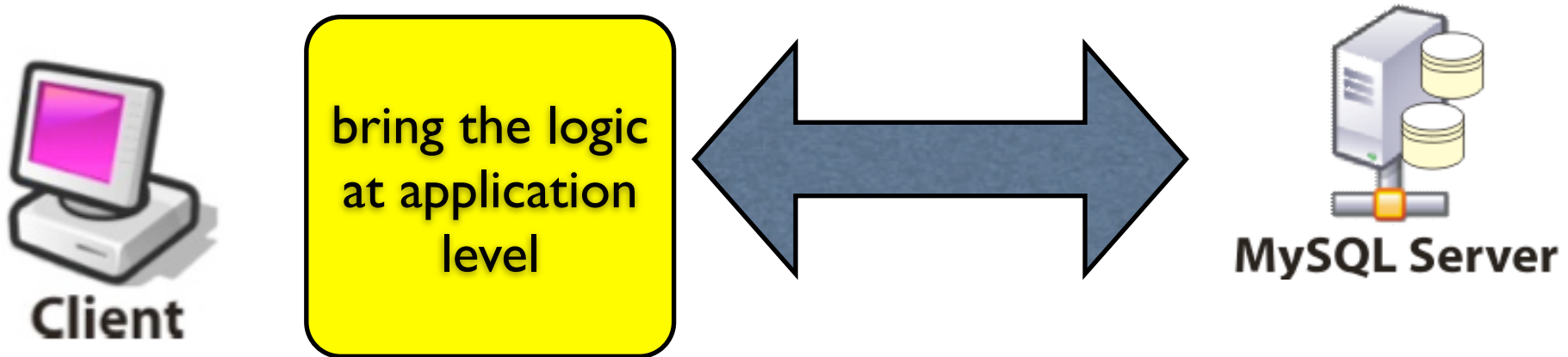


bring the logic  
at application  
level



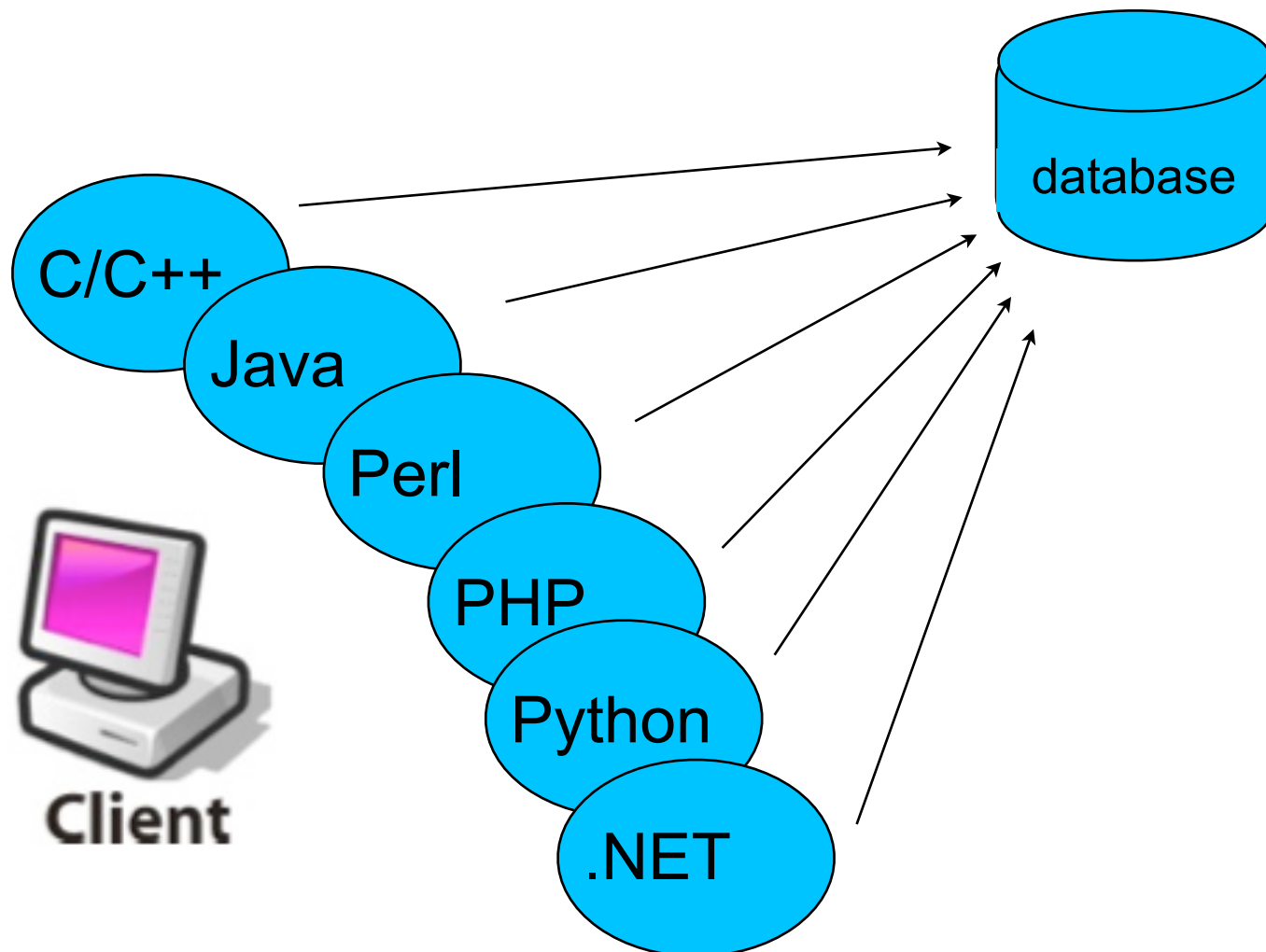
# Solving database problems

## Creative (shortsighted) way

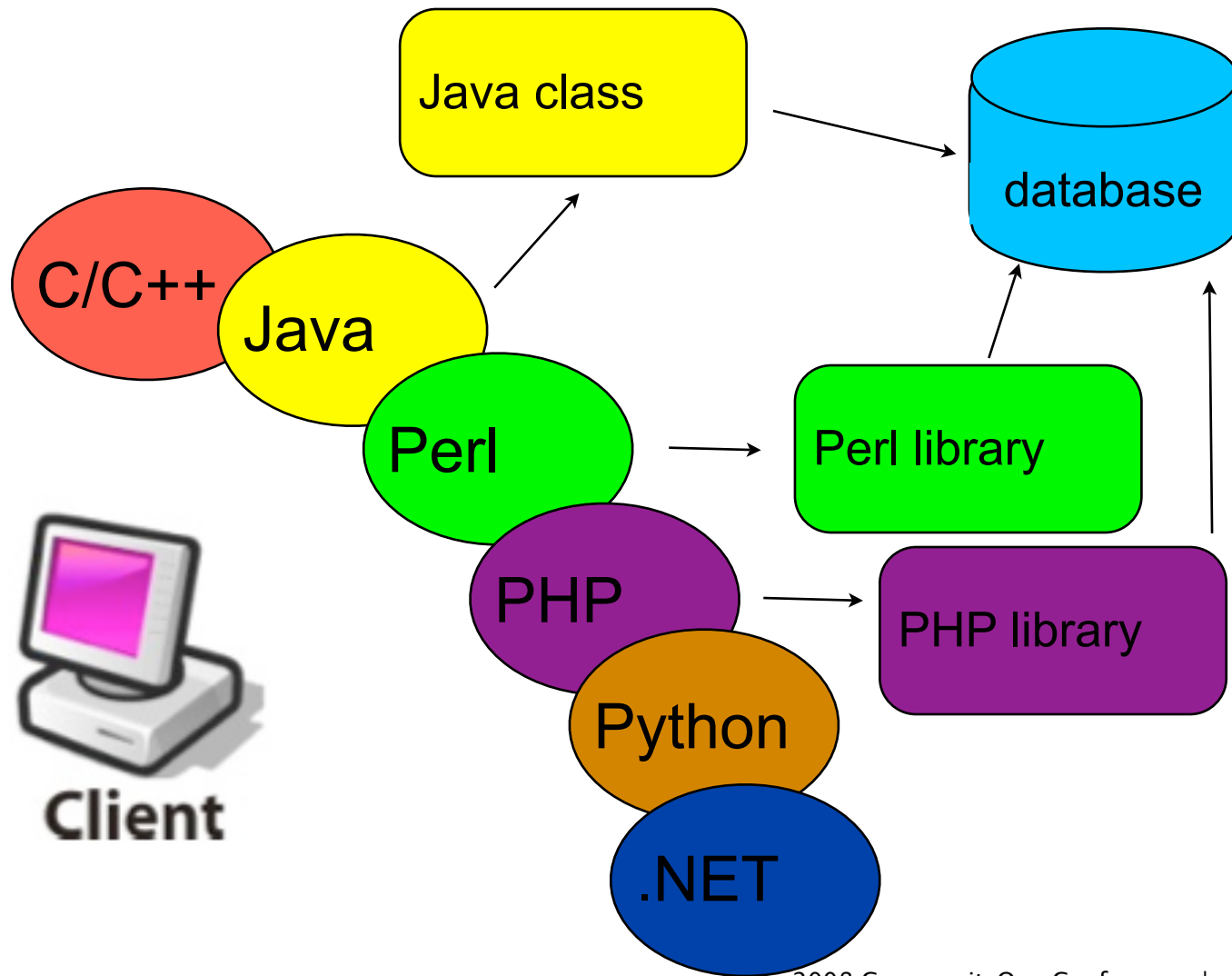


**NOT flexible - Unaccessible from other applications**

# database handling



# database handling too many languages



# Solving database problems

## Creative (enlightened) way



**Client**



set the logic at  
server level  
(stored  
routines)



**MySQL Server**

# Solving database problems

## Creative (enlightened) way



**Client**



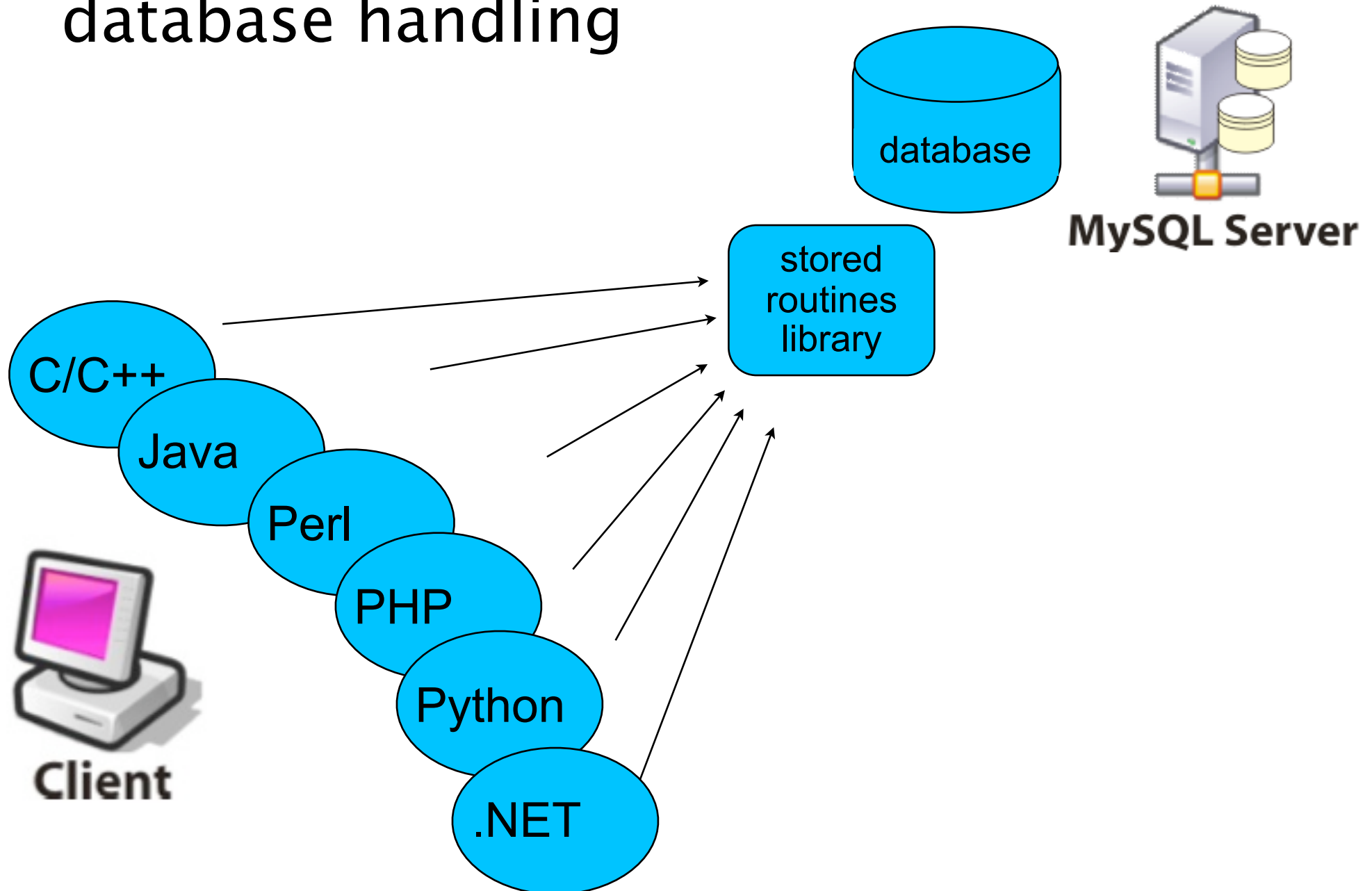
set the logic at  
server level  
(stored  
routines)



**MySQL Server**

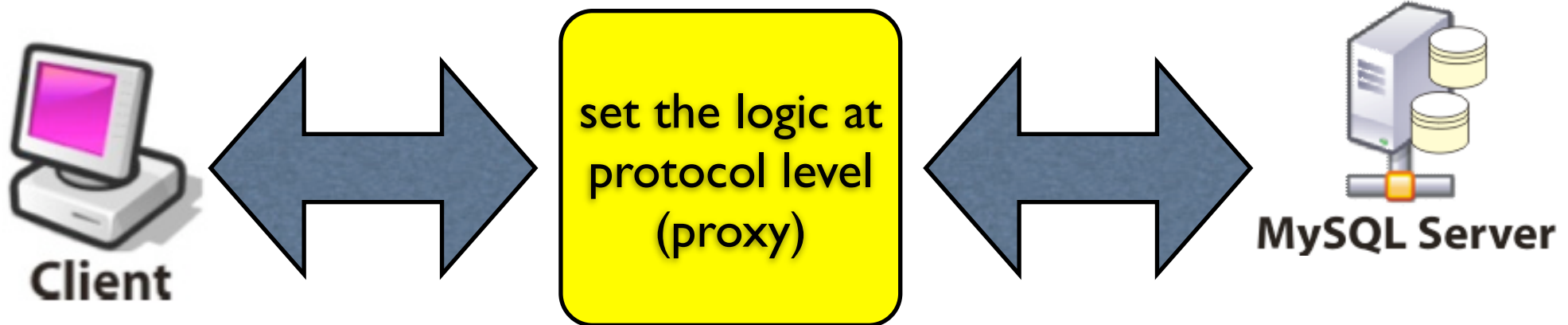
**language limitations - privilege issues - performance**

# database handling



# Solving database problems

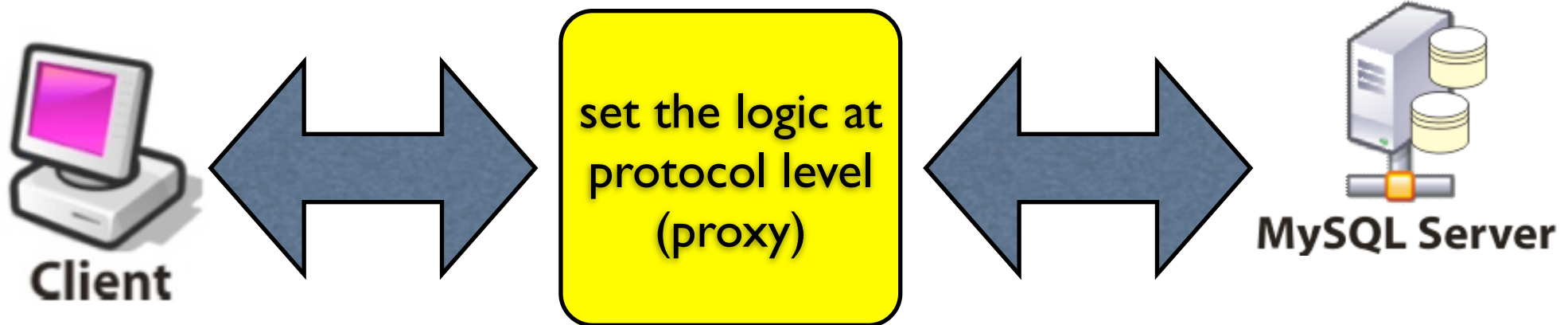
## Creative (more enlightened) way





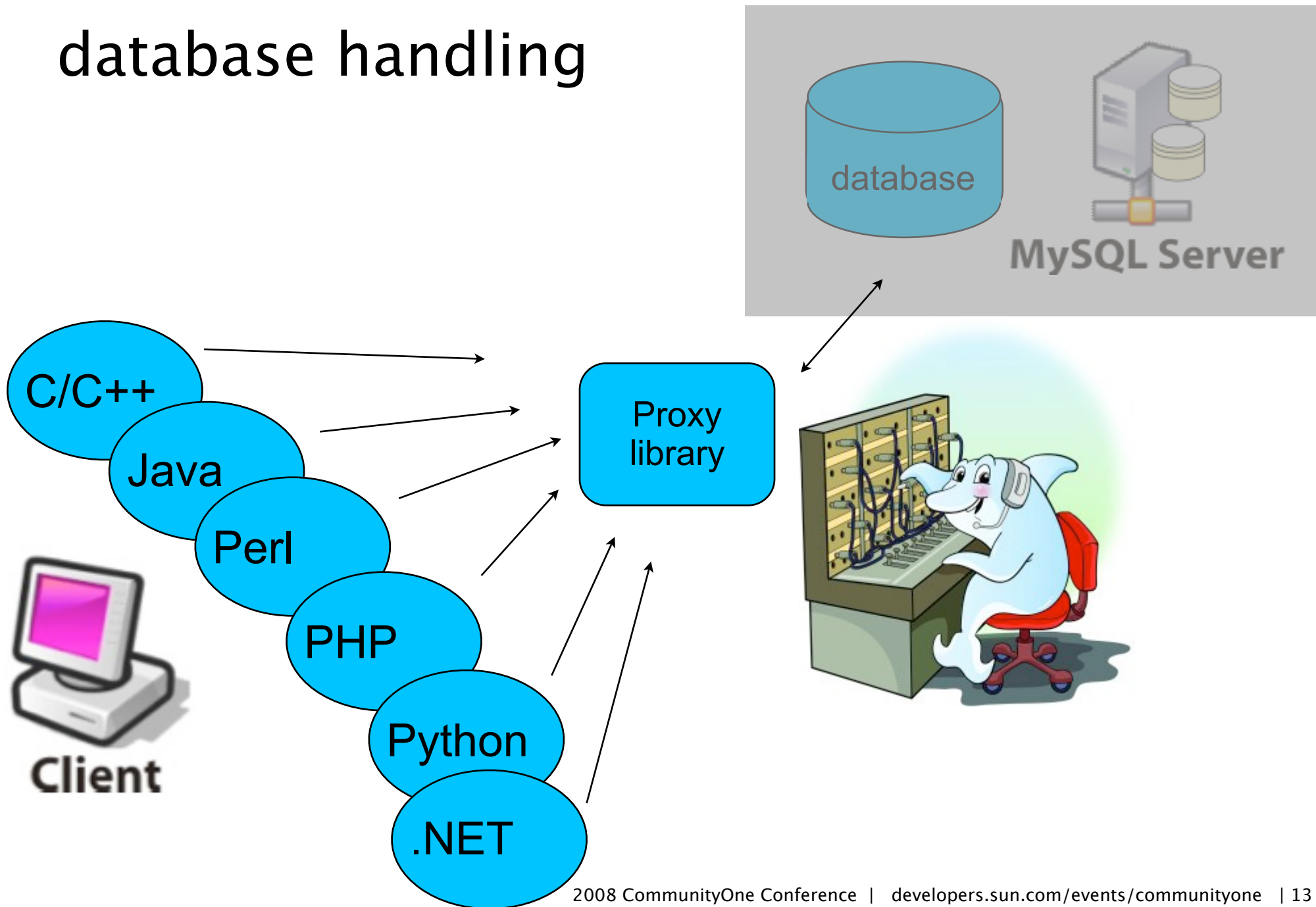
# Solving database problems

## Creative (more enlightened) way



**flexible and suitable for many**

# database handling



# what can you do with MySQL Proxy

# what can you do with MySQL Proxy

- create new commands

## what can you do with MySQL Proxy

- create new commands
- filter queries (deny specific queries)

## what can you do with MySQL Proxy

- create new commands
- filter queries (deny specific queries)
- collect statistics on usage

## what can you do with MySQL Proxy

- create new commands
- filter queries (deny specific queries)
- collect statistics on usage
- implement usage quotas

## what can you do with MySQL Proxy

- create new commands
- filter queries (deny specific queries)
- collect statistics on usage
- implement usage quotas
- execute shell commands



## what can you do with MySQL Proxy

- create new commands
- filter queries (deny specific queries)
- collect statistics on usage
- implement usage quotas
- execute shell commands
- create customized logs

## what can you do with MySQL Proxy

- create new commands
- filter queries (deny specific queries)
- collect statistics on usage
- implement usage quotas
- execute shell commands
- create customized logs
- implement server-side pivot tables

## what can you do with MySQL Proxy

- create new commands
- filter queries (deny specific queries)
- collect statistics on usage
- implement usage quotas
- execute shell commands
- create customized logs
- implement server-side pivot tables
- start/stop a MySQL server remotely

## what can you do with MySQL Proxy

- create new commands
- filter queries (deny specific queries)
- collect statistics on usage
- implement usage quotas
- execute shell commands
- create customized logs
- implement server-side pivot tables
- start/stop a MySQL server remotely
- play movies (seriously!)

## what can you do with MySQL Proxy

- create new commands
- filter queries (deny specific queries)
- collect statistics on usage
- implement usage quotas
- execute shell commands
- create customized logs
- implement server-side pivot tables
- start/stop a MySQL server remotely
- play movies (seriously!)
- make coffee (really?)

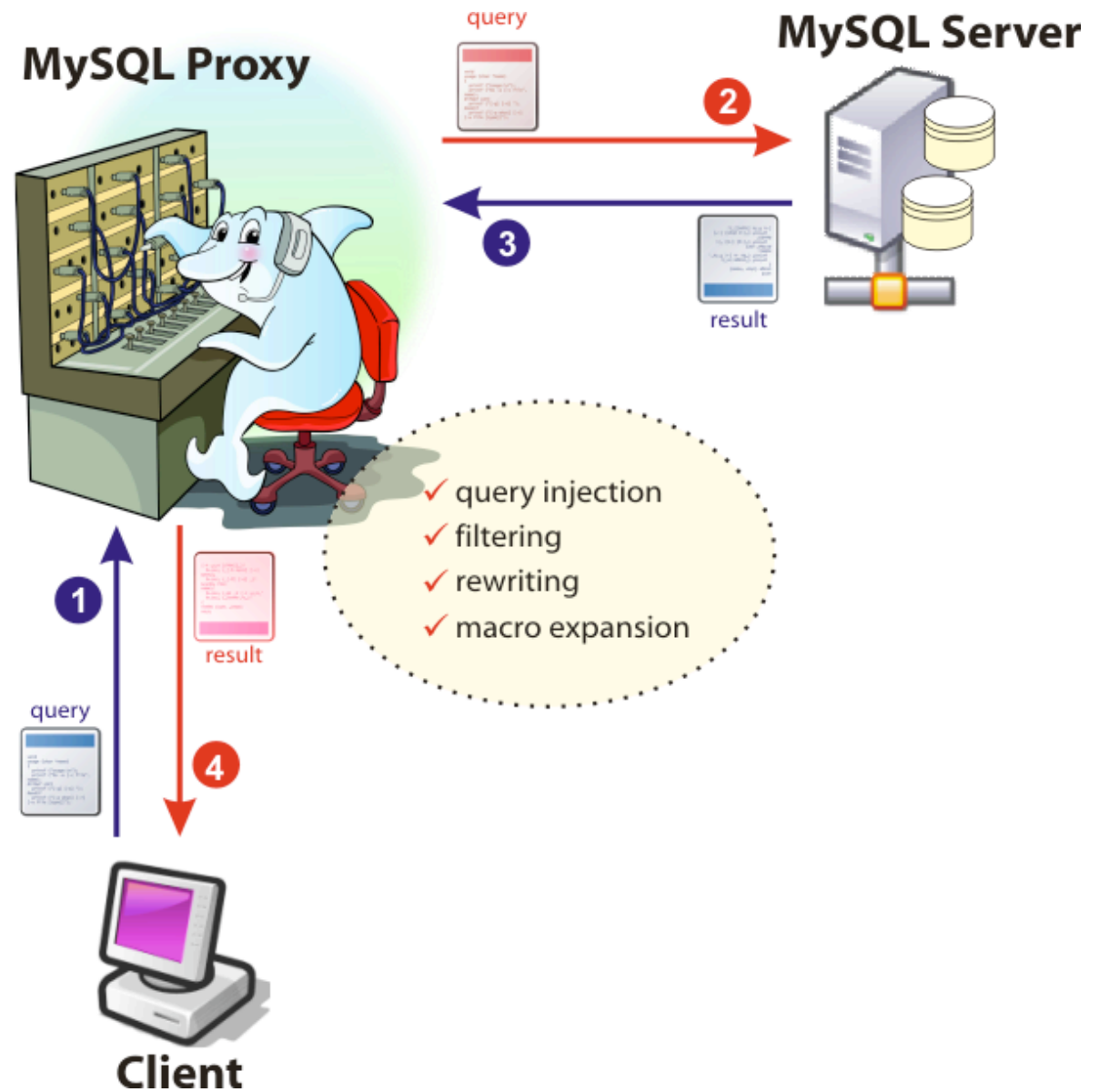
## what can you do with MySQL Proxy

- create new commands
- filter queries (deny specific queries)
- collect statistics on usage
- implement usage quotas
- execute shell commands
- create customized logs
- implement server-side pivot tables
- start/stop a MySQL server remotely
- play movies (seriously!)
- make coffee (really?)
- sharding

## what can you do with MySQL Proxy

- create new commands
- filter queries (deny specific queries)
- collect statistics on usage
- implement usage quotas
- execute shell commands
- create customized logs
- implement server-side pivot tables
- start/stop a MySQL server remotely
- play movies (seriously!)
- make coffee (really?)
- sharding
- load balancing servers

# Basic principles





# Basic principles

# Basic principles

## **PROXY CORE**

**connection  
hook**

**read query  
hook**

**read result  
hook**

# Basic principles

## PROXY CORE

**connection  
hook**

**read query  
hook**

**read result  
hook**

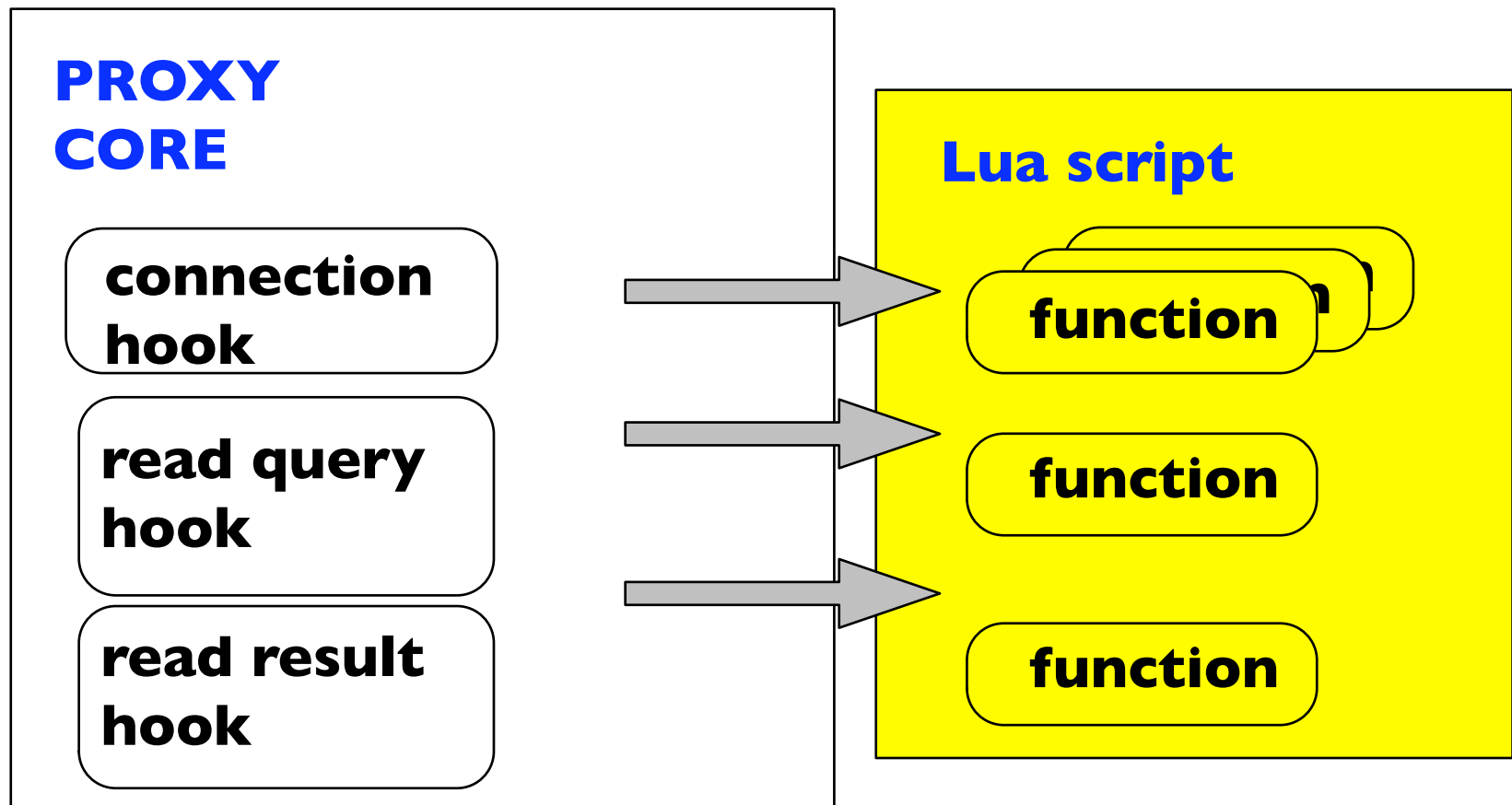
## Lua script

**function**

**function**

**function**

# Basic principles



# Lua



**Why not ...**

**{ Perl ?  
PHP?  
Javascript?  
[whatever]?**

# Lua



# Lua



- **SMALL ( < 200 KB)**

# Lua



- **SMALL ( < 200 KB)**
- **DESIGNED for EMBEDDED systems**



# Lua



- **SMALL ( < 200 KB)**
- **DESIGNED for EMBEDDED systems**
- **Widely used (lighttpd)**

# Lua



- **SMALL ( < 200 KB)**
- **DESIGNED for EMBEDDED systems**
- **Widely used (lighttpd)**

**lighttpd, like MySQL Proxy, was created by Jan Kneschke**

# Lua



**Very popular among  
game writers**

# Lua



**Very popular among  
game writers**



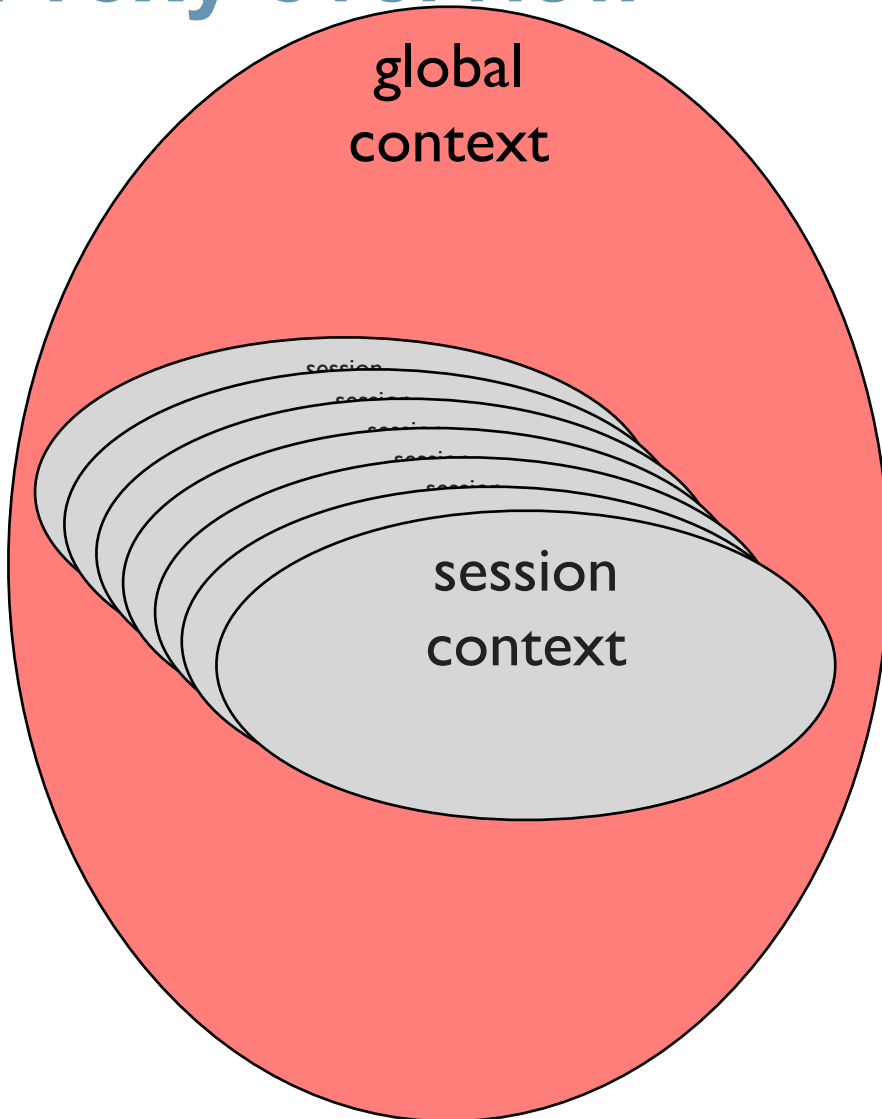
# Lua



**Very popular among  
game writers**



# Proxy overview



## Lua script

connect\_server

read\_handshake

read\_auth

read\_auth\_result

read\_query

read\_query\_result

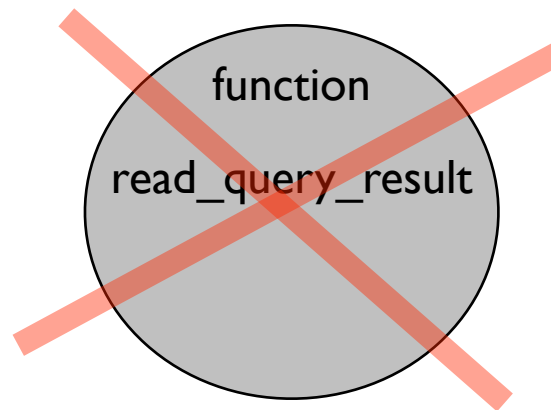
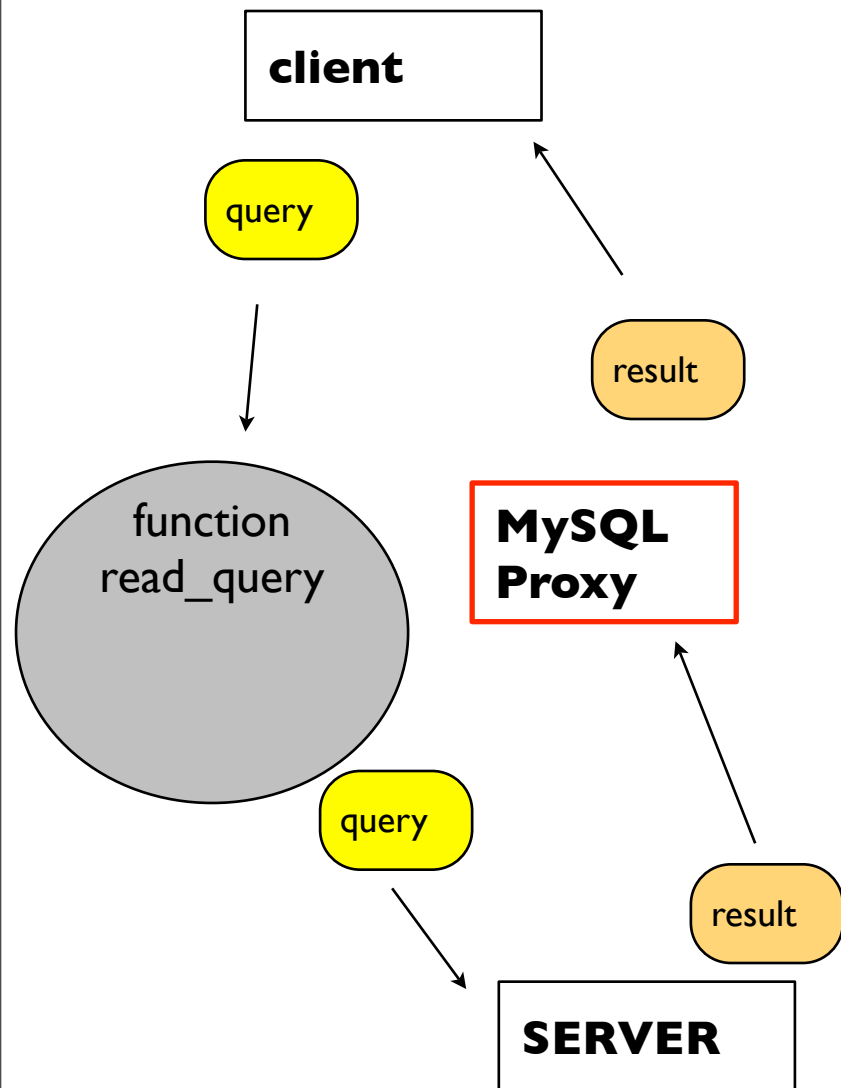
disconnect\_client

## Proxy overview

```
/usr/local/sbin/mysql-proxy \  
--proxy-lua-script=/path/name.lua
```

**IMPORTANT!**  
**THE SCRIPT DOES NOT START UNTIL THE FIRST  
CLIENT CONNECTION**

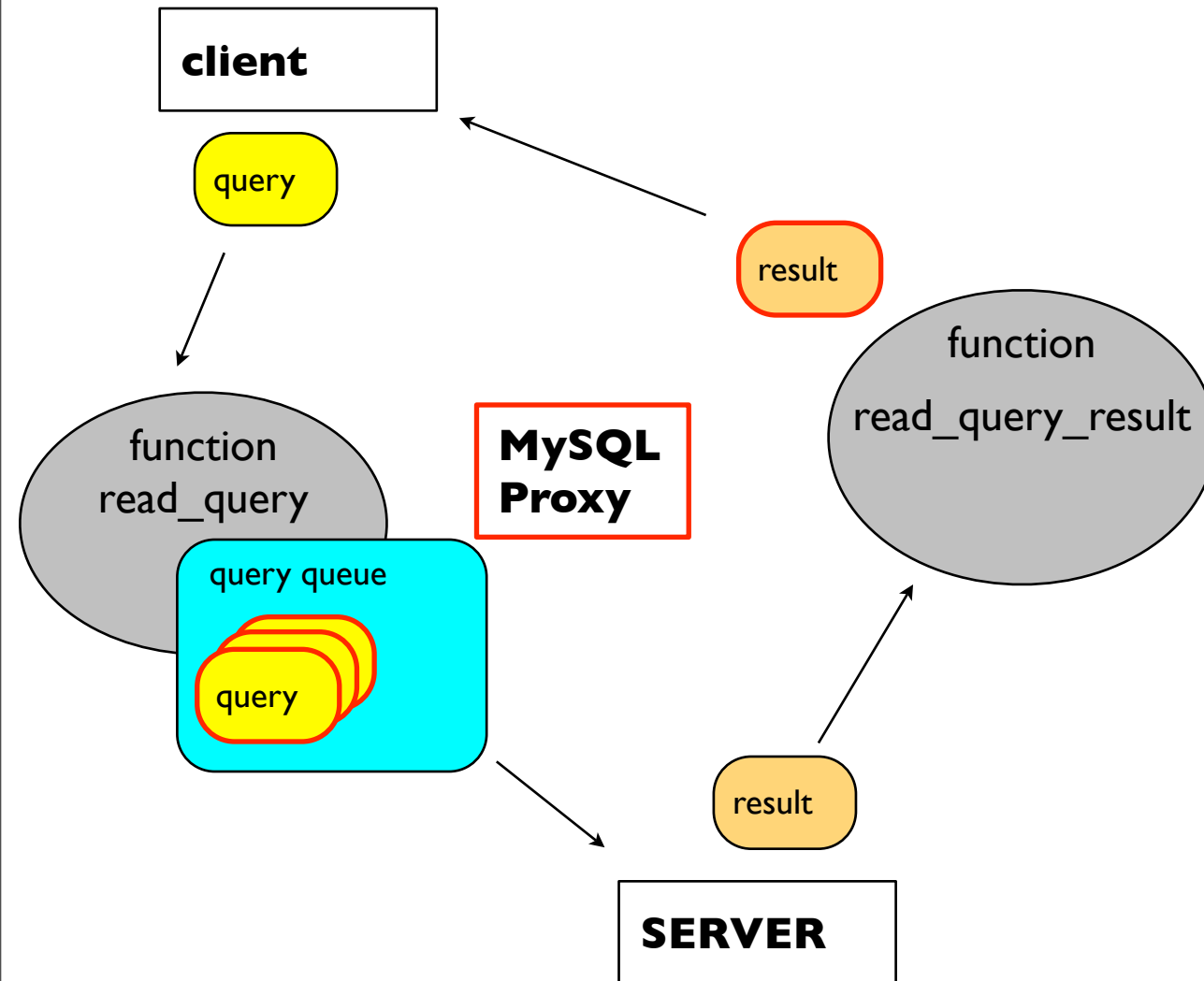
# read\_query and read\_query\_result



if a query is passed directly to the server, its result is **NOT** evaluated by read\_query\_result

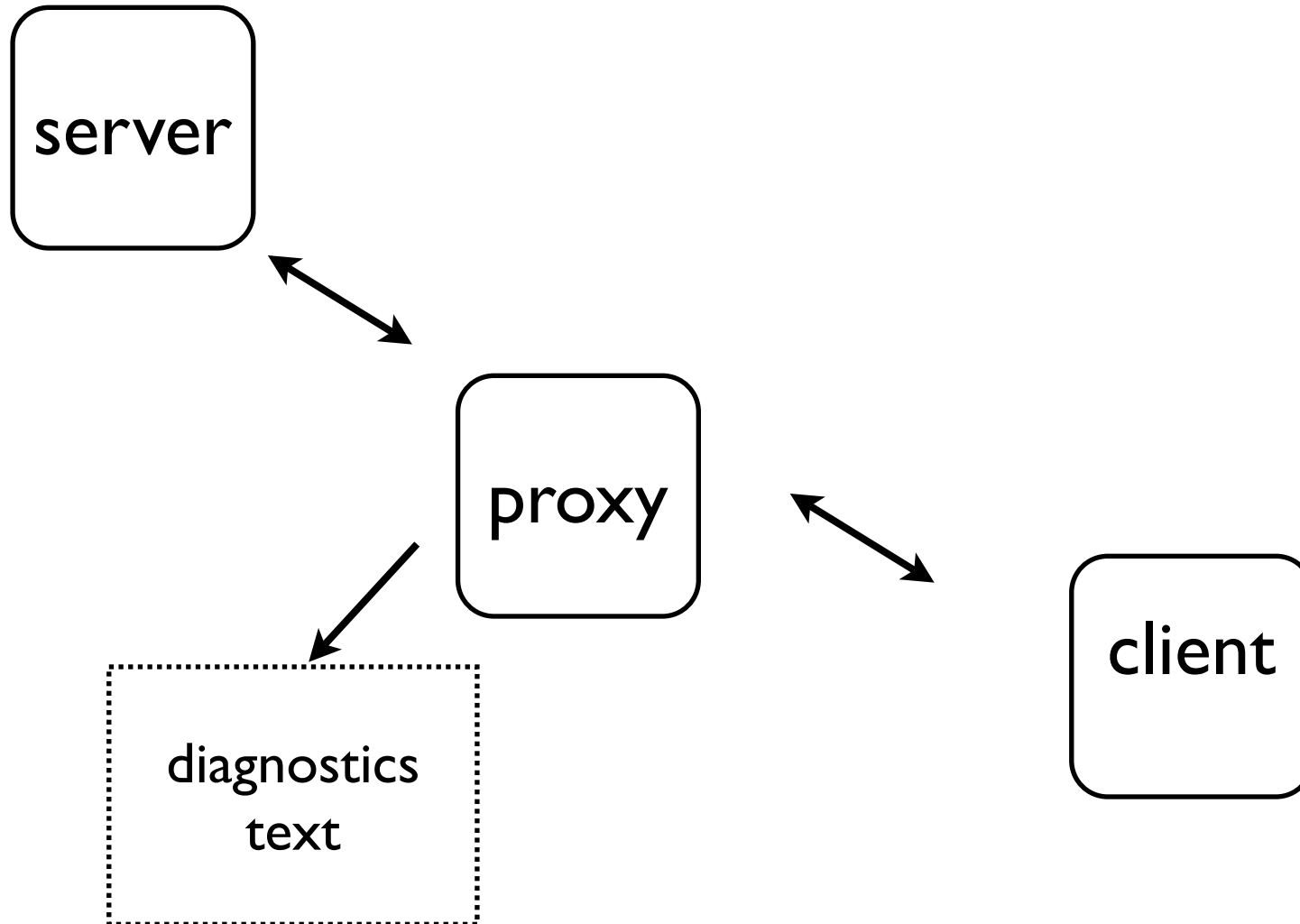


# read\_query and read\_query\_result

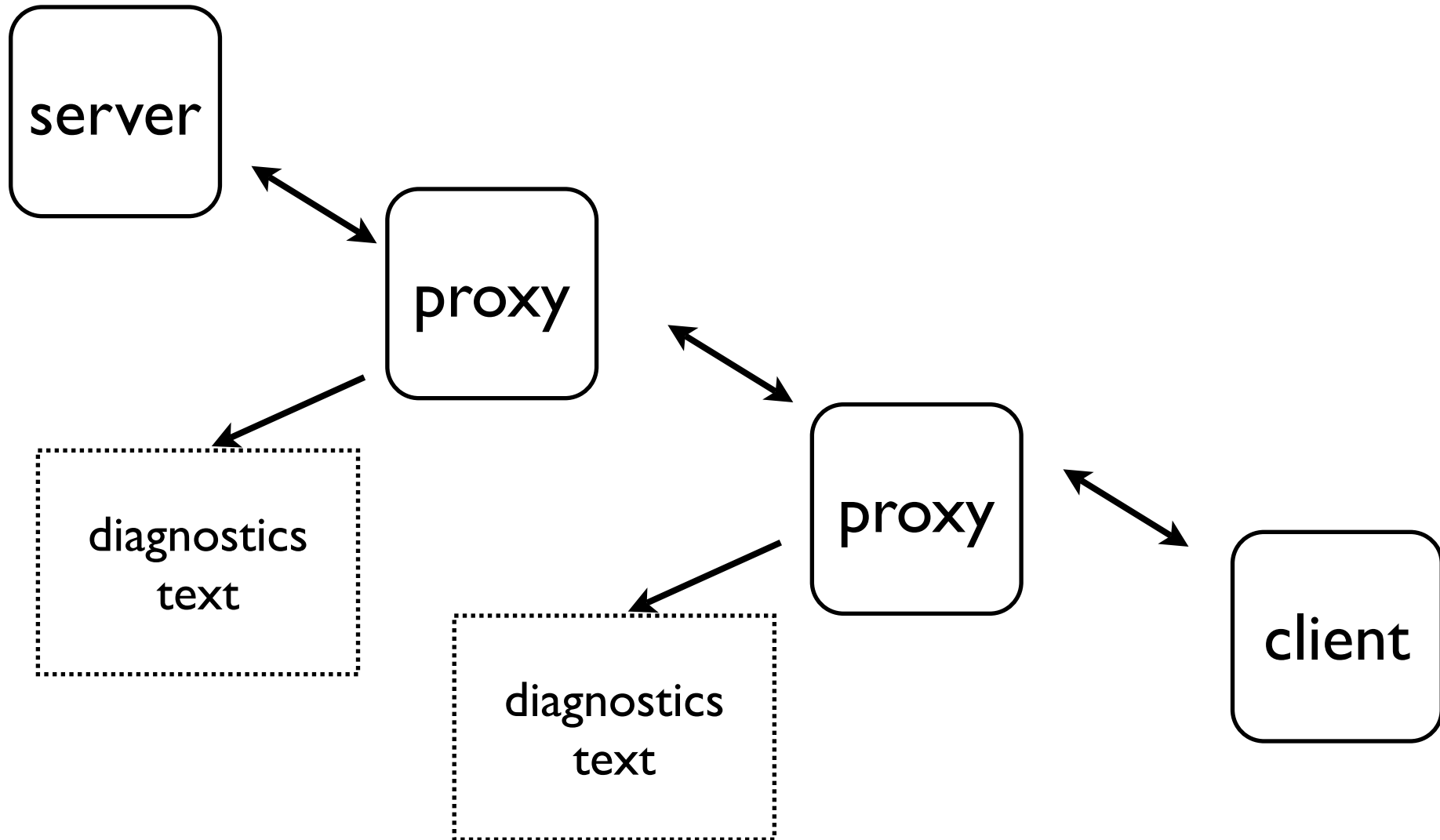


**only if** a query is added to the **query queue**, its result is evaluated by **read\_query\_result**

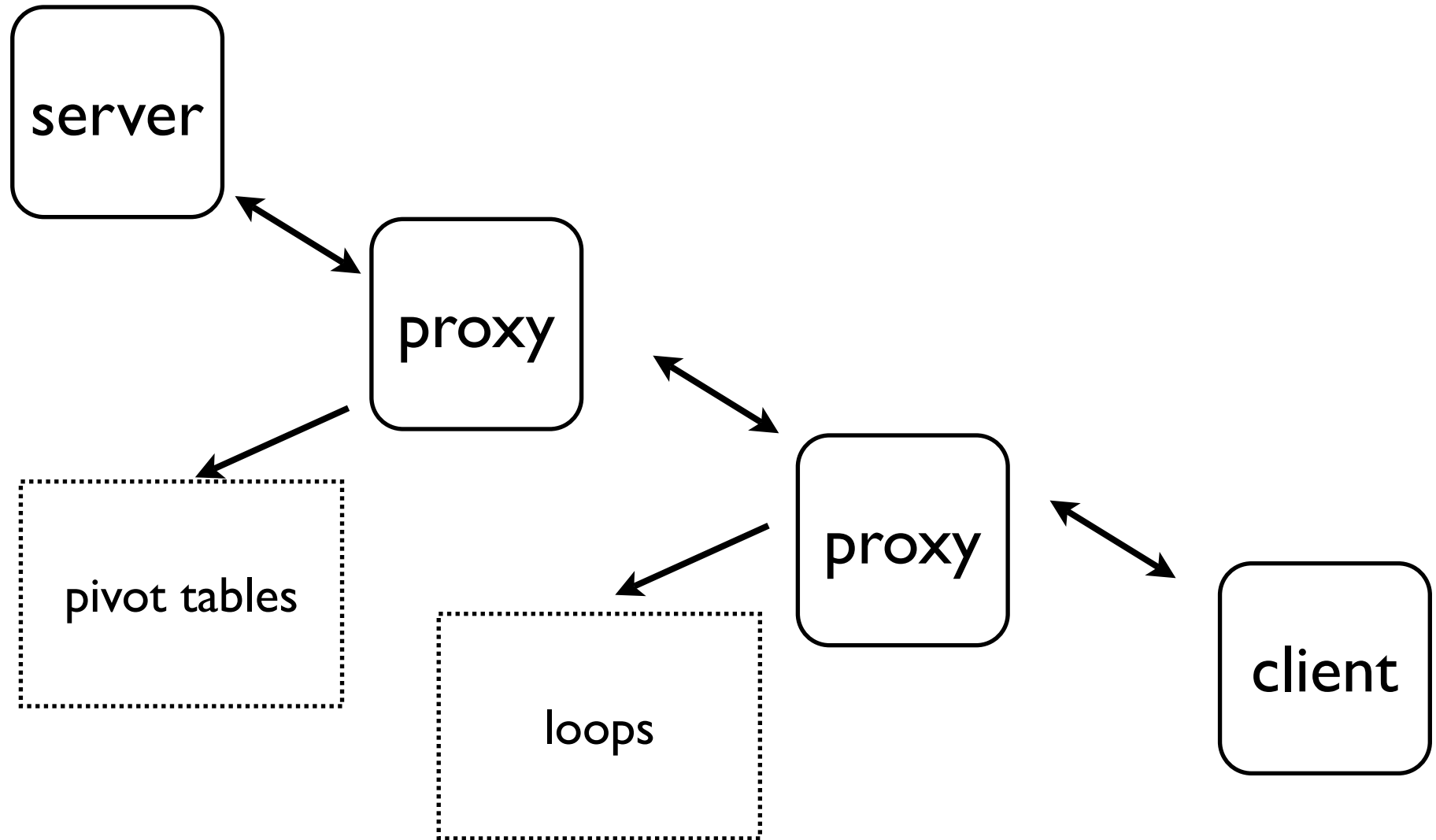
# debugging



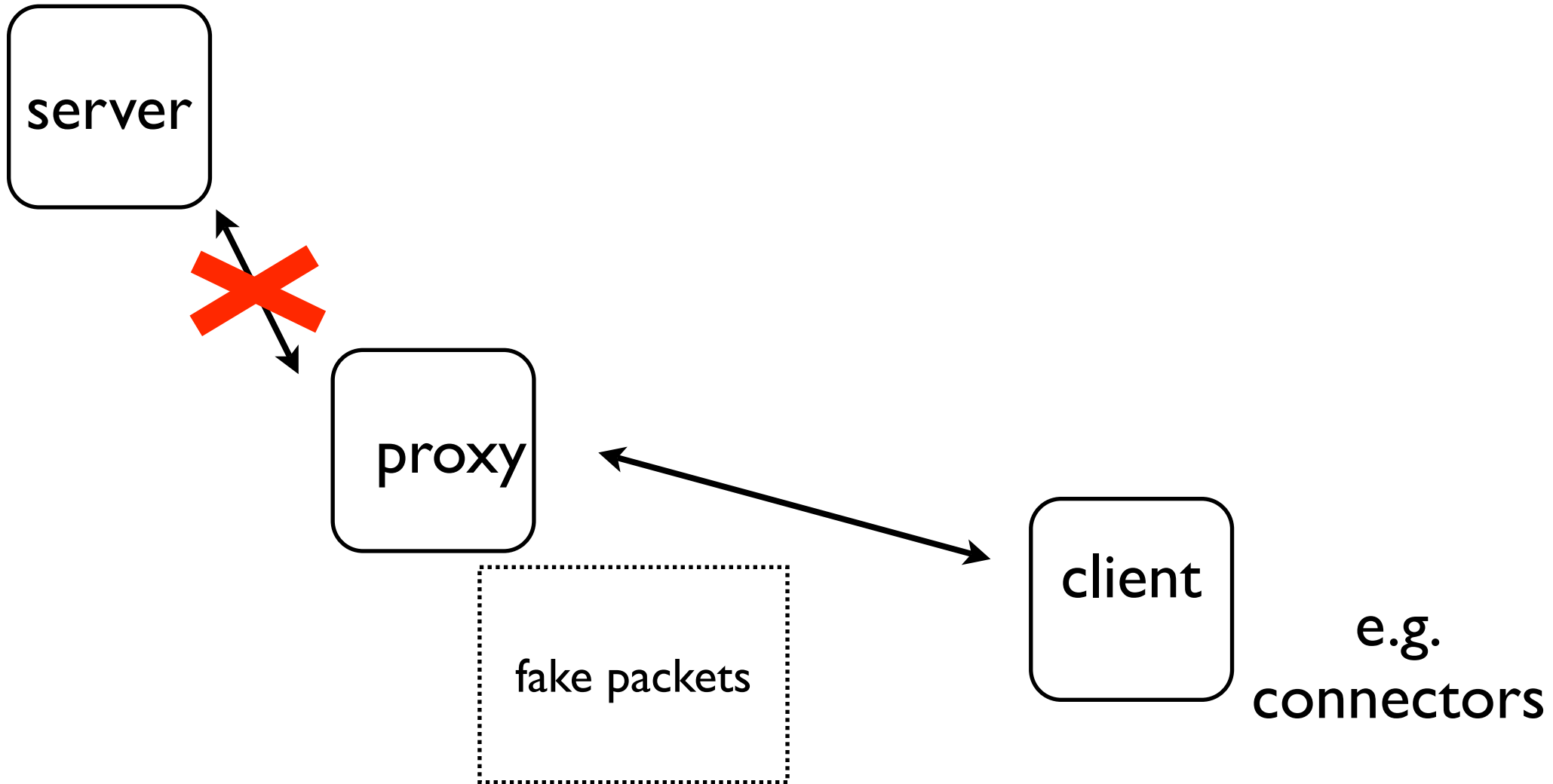
# debugging scripts



# chained proxy: double features



# testing



## An example: user quotas

- You want to limit user access
- When the user reaches a given amount of data, further queries should be rejected
- the quota must work across sessions (code online)

<http://forge.mysql.com/tools/tool.php?id=111>

## User quotas

-- some global variables

-- proxy.global.bandwidth

-- will sum up the used bytes

proxy.global.bandwidth

= proxy.global.bandwidth or {}

-- session\_user will identify the user

-- throughout the session

local session\_user

## User quotas

- session\_user needs to be initialized
- when the user information is passed
- (during the authentication)

```
function read_auth( auth )  
    session_user = auth.username  
    proxy.global.bandwidth[session_user]  
    = proxy.global.bandwidth[session_user]  
    or 0  
end
```



## BREAK (a handy Lua idiom)

-- simple assignment

```
a = a or 0
```

-- corresponds to:

```
if a == nil
then
    a = 0
else
    a = a
end
```

```
if a ~= nil
then
    a = a
else
    a = 0
end
```

## BREAK (a handy Proxy Lua function )

-- returns an error to the client

```
function error_result (msg)
    proxy.response = {
        type          =
            proxy.MYSQLD_PACKET_ERR,
        errmsg        = msg,
        errcode        = 7777,
        sqlstate       = 'X7777',
    }
    return proxy.PROXY_SEND_RESULT
end
```

## User quotas

```
-- read_query (1)
-- checking if the quota has been
-- exceeded
function read_query (packet )
    if proxy.global.bandwidth[session_user]
        > 10000
        and session_user ~= 'root'
    then
        return error_result(
            'you have exceeded your query quota')
    end
-- ...
```

## User quotas

```
-- read_query (2)
-- adding to the totalizer

-- ...
    proxy.global.bandwidth[session_user ]
    =
    proxy.global.bandwidth[session_user]
    + packet:len()
    proxy.queries:append(1, packet )
    return proxy.PROXY_SEND_QUERY
end
```

## User quotas

```
-- read_query_result (1)
-- adding row headers to the totalizer

function read_query_result(inj)
  local fields = inj.resultset.fields
  local rows = inj.resultset.rows
  if fields then
    for i = 1, #fields do
      proxy.global.bandwidth[session_user]
      =
      proxy.global.bandwidth[session_user]
      +
      (fields[i] and fields[i].name:len() or 0)
    end
  end
end
```

## User quotas

```
-- read_query_result (2)
-- adding rows contents to the totalizer

if rows then
    for row in rows do
        for i = 1, #fields do
            proxy.global.bandwidth[session_user]
            =
            proxy.global.bandwidth[session_user]
            +
            (row[i] and row[i]:len() or 0)
        end
    end
end
end
```

## User quotas

```
-- read_query_result (3)
-- displaying the current bandwidth

      print (session_user .. ' -> ' ..
            proxy.global.bandwidth[session_user])
end
```

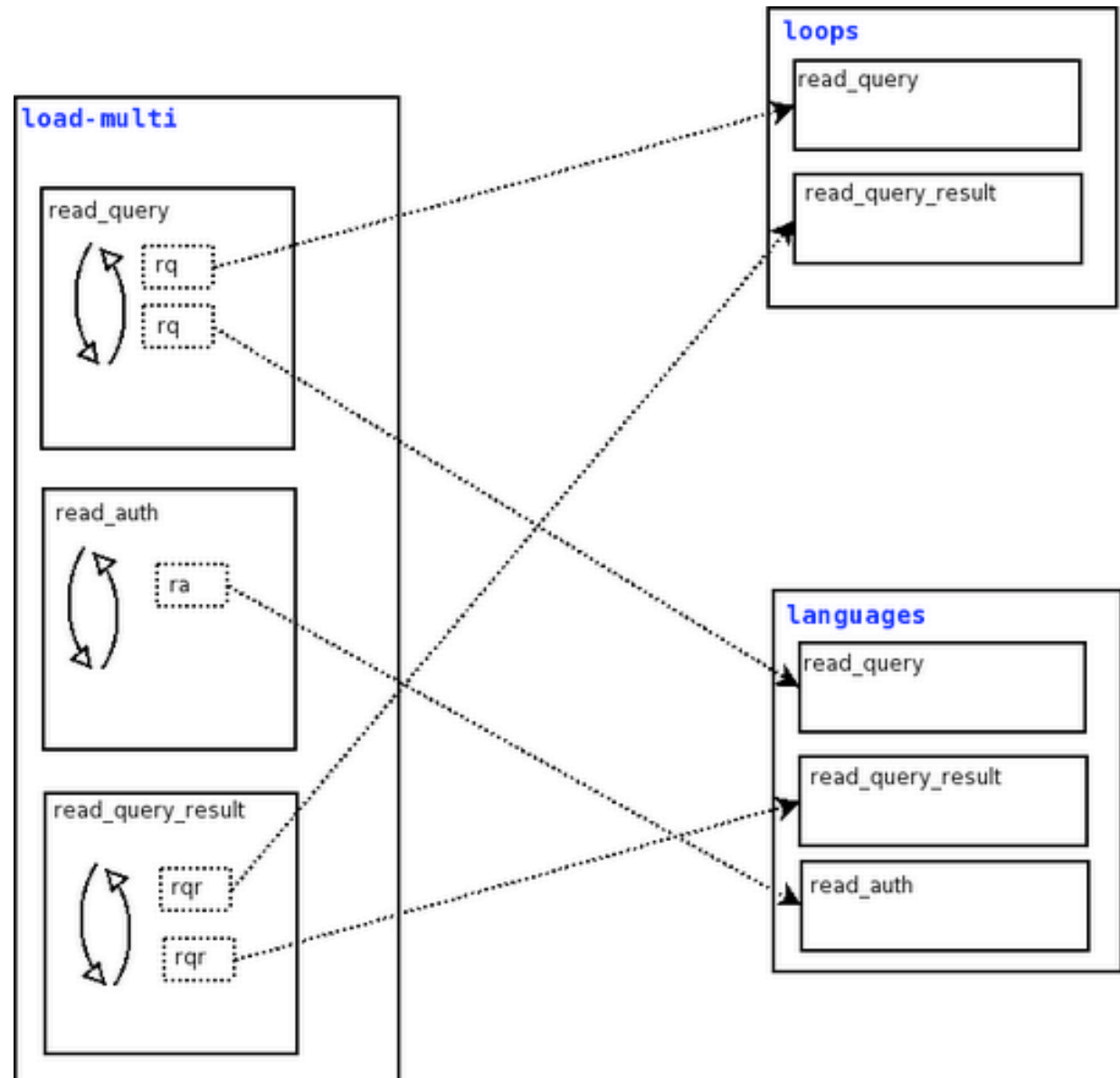
## User quotas (VERY advanced)

- you can create another module
- to be loaded at run time
- (this is really advanced)
- and in such module you define
- a SHOW QUOTAS command



# User quotas (VERY advanced)

-- load\_multi



## User quotas (VERY advanced)

```
mysql> pload show_quotas.lua;
```

```
+-----+
| info                                     |
+-----+
| module "show_quota.lua" loaded        |
+-----+
```

```
mysql> show quotas;
```

```
+-----+-----+
| name      | quota |
+-----+-----+
| simple    | 3578  |
| root      | 2111  |
| msandbox  | 102   |
+-----+-----+
```

## Uses in the field

- Analyzing Queries
- Better Instrumentation
- Connection Management

# Analyzing Queries

Currently

- EXPLAIN Plan
- Session Status
  - Get Before
  - Get After
  - Do diff
  - Remove act of measurement

# Analyzing Queries

With Proxy

```
$mysql-proxy --with-lua-script=analyze_query.lua
```

# Analyzing Queries

```
Norm_Query: "SELECT `device` FROM `schema` . `l_device` WHERE `id` = ? "  
Exec_time: 11.044 ms  
.. Bytes_received = 47  
.. Com_select = 1  
.. Handler_read_key = 1  
.. Handler_read_next = 1  
.. Key_read_requests = 3  
.. Last_query_cost = 1.199  
.. Questions = 1  
.. Table_locks_immediate = 1
```

# Analyzing Queries

```
Norm_Query: "SELECT * FROM ( SELECT `i` . `iv` , `dl` . * FROM `schema` . `vri`  
`i` , `schema` . `vdl` `dl` WHERE `i` . `c_key` = `dl` . `c_key` AND `i` .  
`m_type` IN ( ? ) ORDER BY `i` . `index_value` DESC LIMIT ? ) `foo` ORDER BY  
RAND( ) LIMIT ? "
```

```
Exec_time: 19548 us
```

```
.. Bytes_received = 328
```

```
.. Bytes_sent = 96297
```

```
.. Com_select = 1
```

```
.. Created_tmp_tables = 2
```

```
.. Handler_read_key = 528
```

```
.. Handler_read_prev = 524
```

```
.. Handler_read_rnd = 262
```

```
.. Handler_read_rnd_next = 1047
```

```
.. Handler_write = 1045
```

```
.. Innodb_buffer_pool_read_requests = 817
```

```
.. Innodb_rows_read = 1050
```

```
.. Last_query_cost = 141.249
```

```
.. Qcache_not_cached = 2
```

```
.. Questions = 1
```

```
.. Select_scan = 2
```

```
.. Slow_queries = 1
```

```
.. Sort_rows = 262
```

```
.. Sort_scan = 1
```

```
.. Table_locks_immediate = 2
```

# Analyzing Queries

## Further Uses

- Determine key status's you wish to monitor and write SQL, EXPLAIN Plan, and Status output based on these rules
- e.g.
  - Select\_scan
  - Sort\_merge\_passes
  - Com\_alter\_table (Watching Admin Commands)



# Better Instrumentation

Currently

- SHOW GLOBAL STATUS
  - COM\_??? e.g. SELECT,INSERT,UPDATE etc
- `$ mysqladmin -r -i 1 -n 60 extended-status | grep -v " | 0 "`

# Better Instrumentation

With Proxy

```
$mysql-proxy --with-lua-script=histogram.lua
```

- Gives Read/Write breakdown per table
- Shows top executing queries by frequency
- Shows max and avg execution time

# Better Instrumentation

```
mysql> select * from histogram.tables;
```

table	reads	writes
schema1.dotCommLookup	991	0
schema2xxxxxx.dotCommClient	6	0
schema2xxxxxx.ReleaseNumbers	3	6
schema2xxxxxx.PAYCYCLE	1	0
schema1.dotCommClient	1505	0
schema1.Job	131	0
schema2xxxxxx.CondMod	0	28
schema2xxxxxx.SPINFO	3	0
schema1.Poll	1184	633
schema2xxxxxx.LaborSchedules	0	96
schema1.JobRequest	192	0
schema1.PollTasks	454	1048
schema1.Tasks	1048	0
schema1.auth	141	59
schema2xxxxxx.dotComm	2	0
schema2xxxxxx.MenuItem	0	2446
schema1.Location	19	118
schema2xxxxxx.CONCASH	2	0
schema1.dotComm	156	0
schema1.Client	47	0

```
20 rows in set (0.00 sec)
```

# Better Instrumentation

```
mysql> select * from histogram.queries;
***** 8. row *****
      query: SELECT `type` , COUNT( * ) AS `count` FROM `schema1` . `table1`
`a` WHERE `clientID` = ? AND `bocDate` = ? AND `locationID` IN( ? ) GROUP BY ?
ORDER BY ?
      count: 8
max_query_time: 12932
avg_query_time: 4154.125
***** 35. row *****
      query: SELECT `empl_first_nam` , `empl_last_nam` , `empl_mid_initial`
FROM `schema2` . `table2` WHERE `clientID` = ? AND `empl_ssn` = ? AND
`locationID` = ? LIMIT ?
      count: 84
max_query_time: 61974
avg_query_time: 2501.3095238095
***** 305. row *****
      query: INSERT INTO `schema3` . `table3` SET `eventTime` = NOW( ) ,
`event` = ? , `userID` = ? , `details` = ?
      count: 59
max_query_time: 433610
avg_query_time: 38760.983050847
***** 302. row *****
      query: SELECT * FROM `table4` WHERE `jobID` = ? AND `taskID` = ?
      count: 1056
max_query_time: 17564
avg_query_time: 672.67708333333
```

## Proxy and multiple backends ...

For many of you, this is probably the main reason for attending this talk

## Proxy and multiple backends ...

You may be disappointed.

There is no silver bullet (yet).  
Let us explain

## Proxy options for multiple backends

**--proxy-address**  
=**<host:port>**

**--proxy-read-only-backend-addresses**  
=**<host:port>**

**--proxy-backend-addresses**  
=**<host:port>**

## Proxy functions for multiple backends

`connect_server()`  
(backend host)

`read_auth()`  
(username and password)

`read_auth_result()`  
(accepted/rejected)



## Proxy sample scripts for multiple backends

**rw-splitting.lua**

(masks master/slave behind a proxy)

**auditing.lua**

(captures the info passing)

**tutorial-packets.lua**

(shows really all the info passing  
between client and server)

# Why rw-splitting.lua is a challenge

## PRO

- split queries automatically
- writes go to the master
- reads go to the slaves
- transactions go to the master

## Why rw-splitting.lua is a challenge

### CON

- Just a proof-of-concept
- not stable
- user variables aren't handled nicely
- functions with updates aren't handled


```
function xxx () returns int
BEGIN
    INSERT INTO x VALUES (1) ;
    return 0 ;
END
```

# Community

Tools :: Search Results

[http://forge.mysql.com/tools/search.php?t=tag&k=mysqlproxy](#)

[NN](#)
[mw](#)
[PD](#)
[bm](#)
[PM](#)
[gmax](#)
[books](#)
[progr](#)
[groups](#)
[db](#)
[perl](#)
[mail](#)
[C++](#)
[tech](#)
[util](#)
[news](#)
[m](#)
[e](#)
[events](#)


**MySQL Forge**  
Resources for the MySQL Community

[home](#)
[about](#)
[people](#)
[projects](#)
[tools & code](#)
[software previews](#)
[worklog](#)
[contribute](#)
[wiki](#)
[login](#)
[create account](#)

[- home -](#)
[- browse tools -](#)
[- top rated -](#)
[- newest -](#)

Displaying **1** to **20** of **24** total results

Filter for:

Name/Title	Type	Added On	Author	Rating
MySQL-Proxy connection pooler	code snippet	Jul 9, 2008 @ 00:34	John Loehrer	★★★★★
MySQL Proxy tutorial - Crosstabs	code snippet	Jul 3, 2008 @ 12:49	Giuseppe Maxia	★★★★★
quick wrapper for MySQL Proxy	code snippet	May 27, 2008 @ 18:56	Giuseppe Maxia	★★★★★
Run two chained Proxies in protected environments	code snippet	May 27, 2008 @ 17:20	Giuseppe Maxia	★★★★★
MySQL Proxy - Improved CREATE TABLE ... SELECT	code snippet	May 1, 2008 @ 03:10	Diego Medina	★★★★★
MySQL Proxy - Log warnings and errors	code snippet	Apr 12, 2008 @ 03:45	Diego Medina	★★★★★
MySQL Proxy Tutorial - Return a full dataset	code snippet	Mar 29, 2008 @ 21:47	Giuseppe Maxia	★★★★★
MySQL Proxy Tutorial - Return a simple dataset	code snippet	Mar 29, 2008 @ 21:43	Giuseppe Maxia	★★★★★
MySQL Proxy Tutorial - Return an error	code snippet	Mar 29, 2008 @ 21:40	Giuseppe Maxia	★★★★★
MySQL Proxy tutorial - tracking transaction status	code snippet	Dec 2, 2007 @ 11:11	Giuseppe Maxia	★★★★★
MySQL Proxy tutorial - Measuring bandwidth by user	code snippet	Dec 2, 2007 @ 11:01	Giuseppe Maxia	★★★★★
MySQL Proxy tutorial - Measuring bandwidth by session	code snippet	Dec 2, 2007 @ 10:55	Giuseppe Maxia	★★★★★
MySQL Proxy tutorial - Show all hooks	code snippet	Dec 2, 2007 @ 10:51	Giuseppe Maxia	★★★★★
MySQL Proxy tutorial - Blocking unwanted queries	code snippet	Nov 19, 2007 @ 14:10	Giuseppe Maxia	★★★★★
mysql-proxy tutorial - loops	code snippet	Aug 12, 2007 @ 11:42	Giuseppe Maxia	★★★★★
mysql-proxy - Query statistics	code snippet	Jul 2, 2007 @ 20:42	Stephane Varoqui	Not rated
Using mysql-proxy without changing port on Linux	code snippet	Jun 30, 2007 @ 12:20	Giuseppe Maxia	★★★★★
mysql-proxy tutorial - query and basic result logging	code snippet	Jun 29, 2007 @ 09:21	Giuseppe Maxia	★★★★★

## Community Contribution - Gaia Online

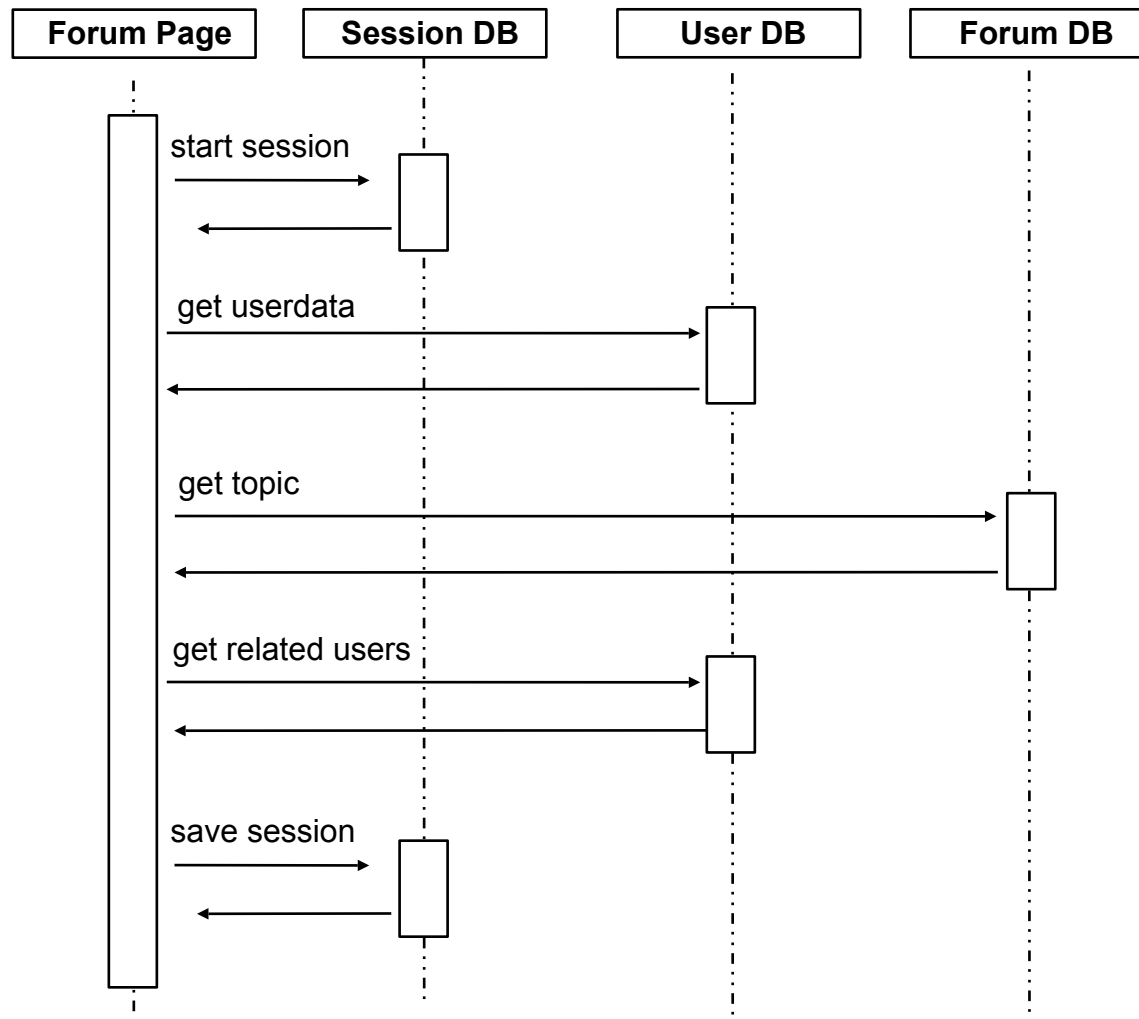
- online hangout for teens
- 5 million unique visitor
- 1.3 billion forum entries

## Community Contribution - Gaia Online

### Existing Architecture

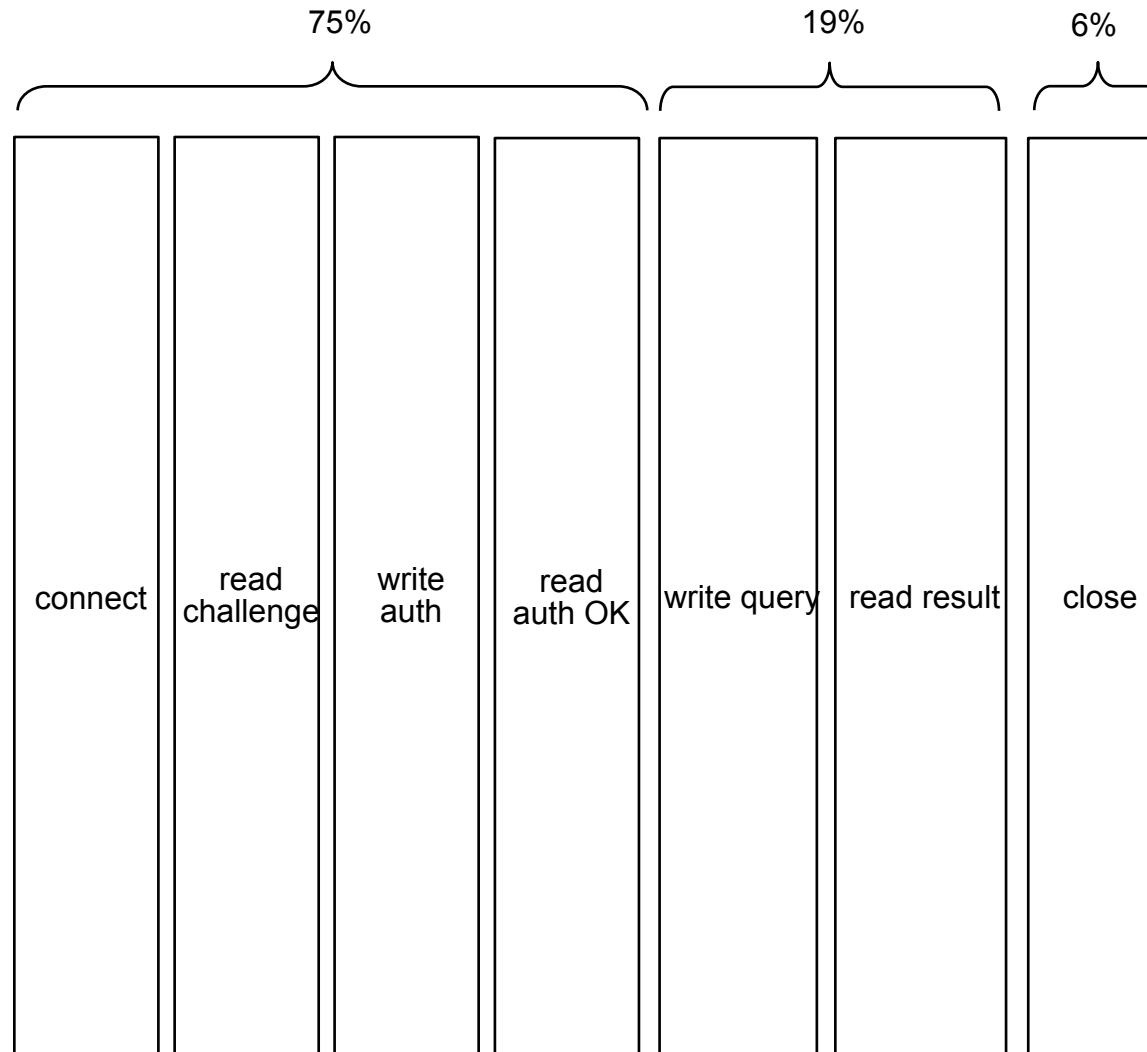
- Data is sharded
- large number of different db server connections per page
- Analysis shows 75% of time in authorization of connection

# Community Contribution - Gaia Online





# Community Contribution - Gaia Online

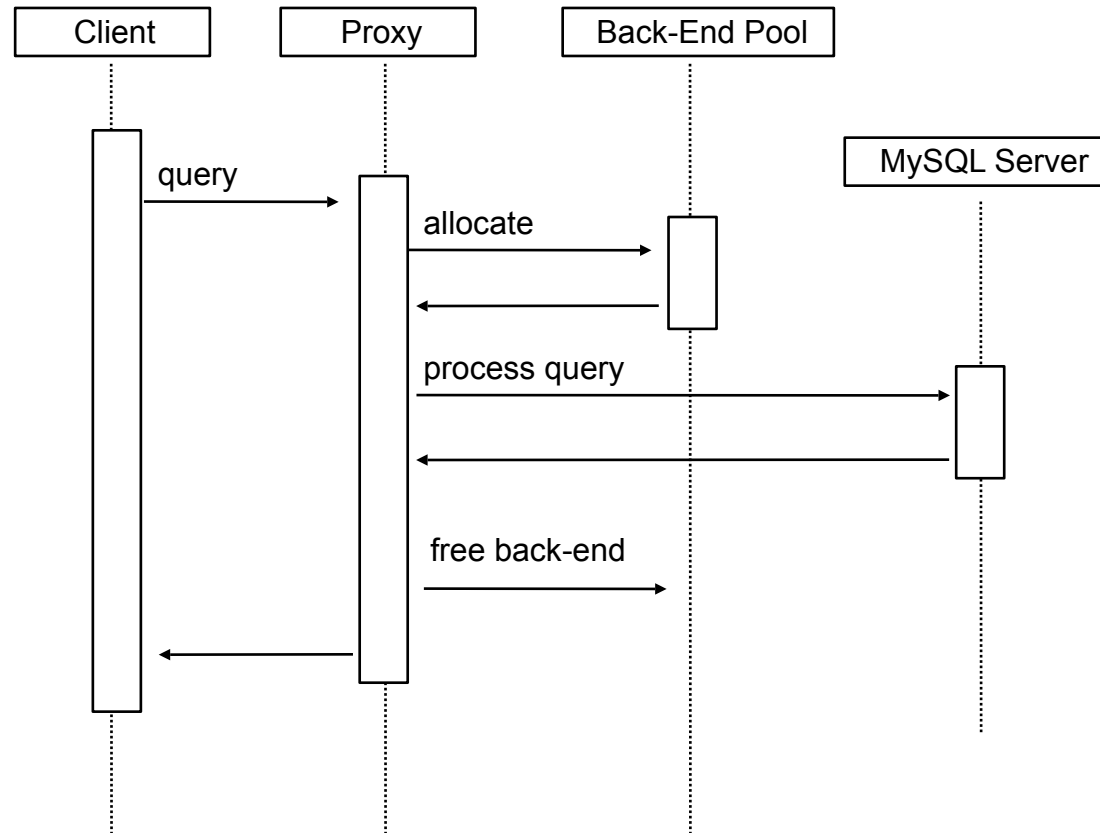


## Community Contribution - Gaia Online

### Use of Proxy

- Connection multiplexer
- recycling back-end connections
- app connections are transparent
- per connection authentication is removed from normal connections

# Community Contribution - Gaia Online



## Community Contribution - Gaia Online

### Use of Proxy

- turning 7,500 connections
- into 100 connections (in pools)
- large reduction in per query time

# The future of MySQL Proxy

## MySQL Load Balancer

*MySQL software which can route database queries to the appropriate database in a scale-out environment.*

- Intelligently route reads across slaves
  - Use database least behind, least loaded
  - Remove latent slaves from read rotation
  - Other distribution algorithms
- Improves/scales throughput of reads for online applications
  - Helps customers reduce cost of adding slaves

## MySQL Query Analyzer

*MySQL Enterprise Monitor feature that allows users to trace, monitor, and analyze MySQL query activity for specific servers, users, and applications.*

- Adaptive “Evil” query collection/tracing
- Historical browsing/analysis
  - “Needle in a haystack” identification of worst queries
  - Worst execution times, # of execs, etc.

SQL code is the #2 cause of performance issues  
97% of those surveyed will use this

# MySQL Query Analyzer

Monitor

Advisors

Query Analysis

Graphs

Replication

Settings

[Manage Watches](#) | [Browse Queries](#) | [View Running](#)

default-server:3306 Running Queries

By User

Limit

All

50

filter

reset

Id	Query	User	Session	Time	Ave Time
942	SELECT hotels.*,hotelinfo.*,continents.Continent,countries.Country,countries.CountryPhone,states.State,cities.City,...	root	121	56 ms	33 ms
941	SELECT CountryName FROM countries WHERE CountryID='213'	root	121	1 ms	1 ms
940	SELECT hotels.*,continents.Continent,countries.Country,states.State,cities.City,hotelchains.Chain,locations.Location...	root	121	45 ms	32 ms
939	SELECT hotels.*,continents.Continent,countries.Country,states.State,cities.City,hotelchains.Chain,locations...				
938	SELECT hotels.*,continents.Continent,countries.Country,states.State,cities.City,hotelchains.Chain,locations...				
937	SELECT hotels.*,continents.Continent,countries.Country,states.State,cities.City,hotelchains.Chain,locations...				

Browse Queries

Query Search

Time Display

Hours

Minutes

View

Query Ty

SELECT\*

Interval

06

00

Server

SELECT

Query	Total Execution	Num
<a href="#">+ UPDATE test3 SET field2...field5=? WHERE field1=? (3)</a>	61,712 ms	203
<a href="#">- SELECT user0_userName ... user0_userName LIKE ? (3)</a>	14,761 ms	509
127.0.0.1:13306	14,761 ms	509
127.0.0.1:3307	14,761 ms	509
127.0.0.1:3308	14,761 ms	509
<a href="#">+ UPDATE testBug28469 SET field1=? (3)</a>	45 ms	9
<a href="#">+ No-Op (3)</a>	9 ms	3

Summary Details

Example Details

Example Explain

29 ms

215,...

Alias

None specified.

Canonical Query Text

[truncated](#) | [full](#) | [formatted](#)  
 SELECT user0\_userName AS userName0\_0\_, user0\_ivalue AS ivalue0\_0\_, user0\_CNAME AS CNAME...

Execution Time Statistics

Max Time	Min Time	Avg Time	Total Time	Standard Deviation
37 ms	21 ms	29 ms	14,761 ms	3 ms

Row Statistics

Max Rows	Min Rows	Avg Rows	Total Rows	Standard Deviation
593	254	423	215,561	-67

Number of Executions

509

Summary Time Span

From Mar 13, 2008 3:01:14 PM to Mar 13, 2008 3:01:14 PM.

hide



## Next Steps

### ***MySQL Proxy Forum***

<http://forums.mysql.com/list.php?146>

### ***Documentation – MySQL Proxy***

<http://dev.mysql.com/doc/refman/5.1/en/mysql-proxy.html>

### ***Documentation – MySQL Load Balancer***

<http://dev.mysql.com/doc/refman/5.1/en/load-balancer.html>

### ***Beta Testing MySQL Proxy-Enabled Products***

[enterprise-beta@mysql.com](mailto:enterprise-beta@mysql.com)

### ***MySQL Proxy Online Poll***

<http://dev.mysql.com/tech-resources/quickpolls/mysql-proxy.html>

THANKS

**Let's talk!**

