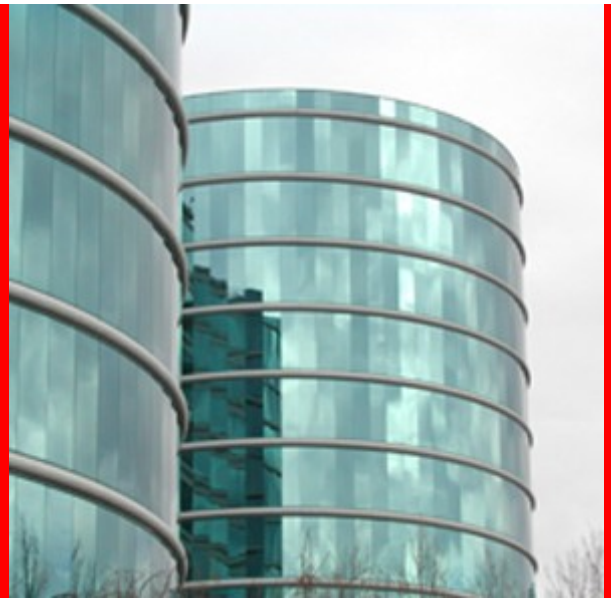

ORACLE®



ORACLE®

PHP and MySQL

Johannes Schlüter
MySQL Engineering Team

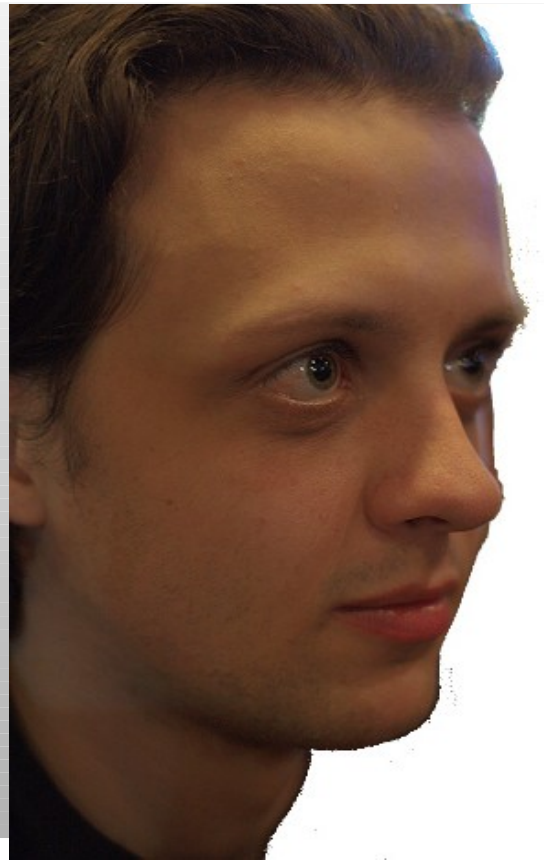
Program Agenda

- Deliver this Presentation
- Enjoy lunch

Program Agenda

- Introduction
- MySQL 5.6
- PHP's MySQL Support
- The PHP World Today
- mysqlnd plugins
- MySQL in the Future
- Q&A

Johannes Schlüter



MySQL User since 1998
PHP User since 1999

PHP Contributor since 2004
Release Master, PHP 5.3

MySQL Team Member since 2007

Second time at Dutch PHP

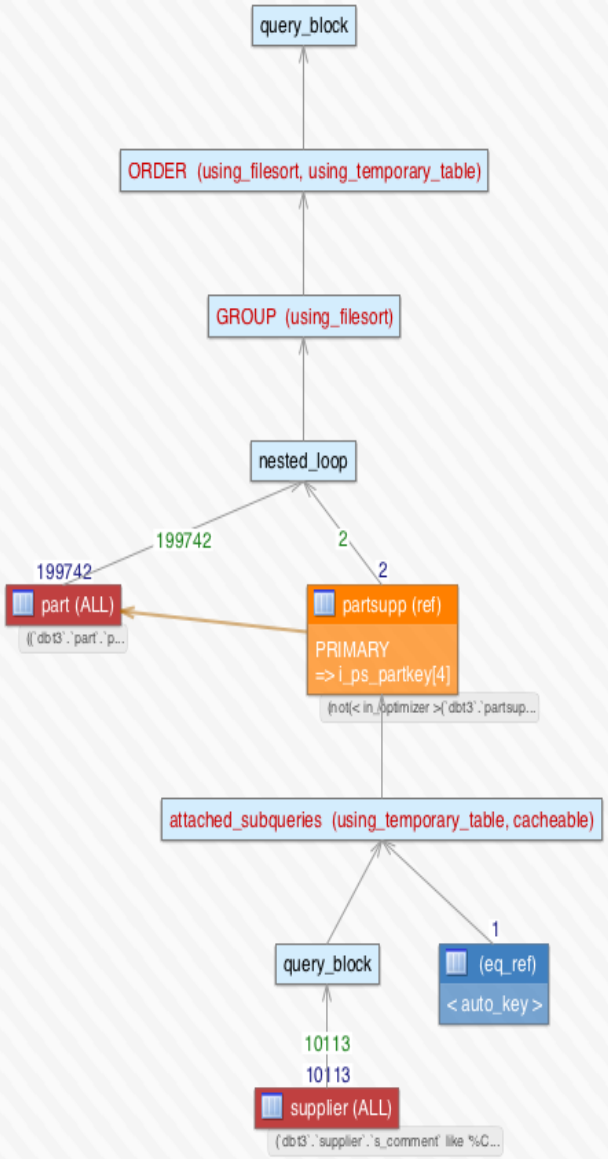
MySQL 5.6

MySQL Optimizer

- Subquery Optimizations
- File sort optimizations with small limit
  - 4X better execution time – 40s to 10s
- Index Condition Pushdown
  - 160X Better execution time – 15s to 90ms
- Postpone Materialization of views/subqueries in FROM
  - 240X better execution time for EXPLAIN – 8m to 2s
- Batched Key Access and Multi Range Read
  - 280X Better execution time – 2800s to 10s

MySQL Optimizer
Diagnostics and Debugging

- EXPLAIN
  - INSERT, UPDATE, and DELETE
  - JSON format for better readability
- Persistent Optimizer Statistics - InnoDB
- Optimizer Traces



Performance Schema Improvements

- Statements/Stages
  - Most resource intensive queries? Where do they spend time?
- Table/Index I/O, Table Locks
  - Which application tables/indexes cause the most load or contention?
- Users/Hosts/Accounts
  - Which application users/hosts/apps consume the most resources?
- Network I/O
  - Network loaded? How long do sessions idle?
- Summaries
  - Aggregate stats grouped by thread, user, host, account or object

InnoDB in MySQL 5.5

- Performance and Scalability
  - Multiple buffer pool instances
  - Multiple rollback segments
  - Improved purge scheduling
  - Extended change buffering with delete buffering and purge buffering
  - Native async I/O support on Linux
  - Improved log sys mutex
  - Separate flush list mutexWindows performance improvements
  - Performance schema for InnoDB

InnoDB in MySQL 5.6

- Performance and Scalability
  - Split the kernel mutex
  - Multi threaded purge
  - Use rw_locks for page_hash
  - Add 'page_cleaner' thread to flush dirty pages
  - Ibuf merge rate improvement
  - Configurable data dictionary cache
  - InnoDB persistent optimizer statistics
  - MRR/ICP support for InnoDB
  - Online ALTER TABLE

InnoDB in MySQL 5.6

- InnoDB FullText Search
  - Support all query types supported by MyISAM:
    - Natural language search
    - Query expansion
    - Boolean search
  - Plus
    - Proximity search
    - Create full-text index with parallel tokenization and parallel sort

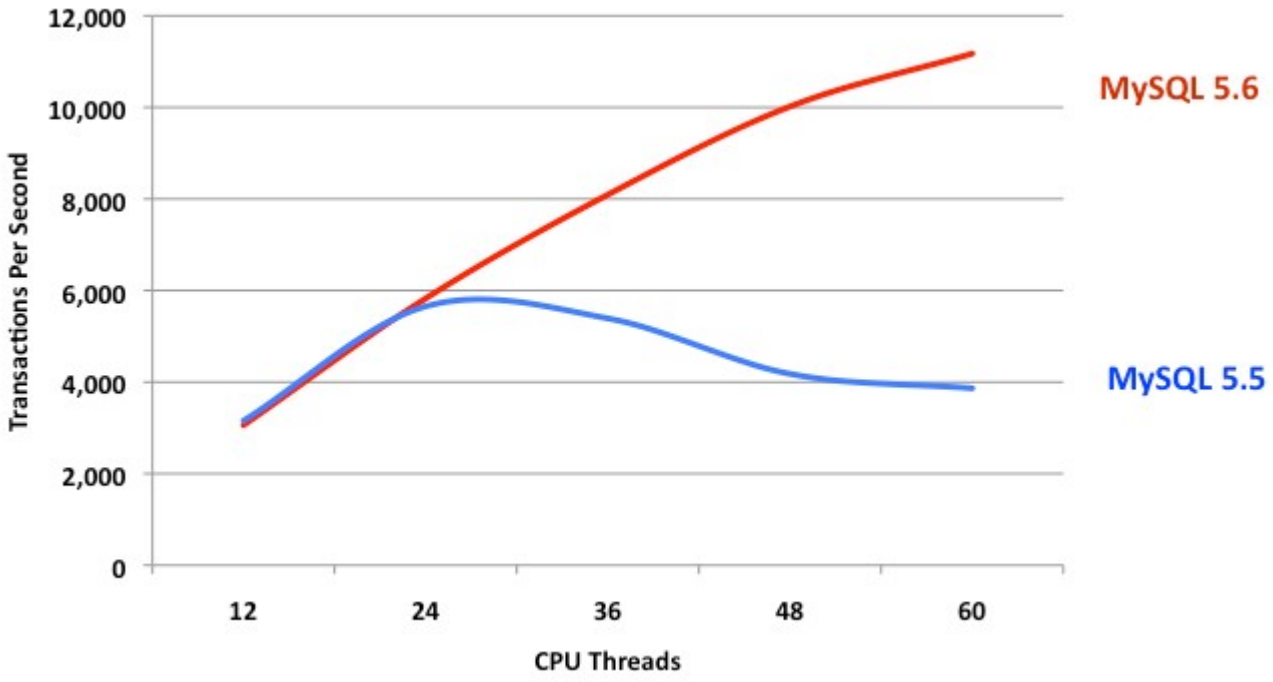
Replication Improvements

- Crash-safe master
  - Automatic binary log file trimming
  - If the master crashes, binary log will recover automatically from incomplete file flushes.
- Crash-safe slave
  - Slave Info Tables
- Global Transaction Ids
- Multi-Threaded Slaves

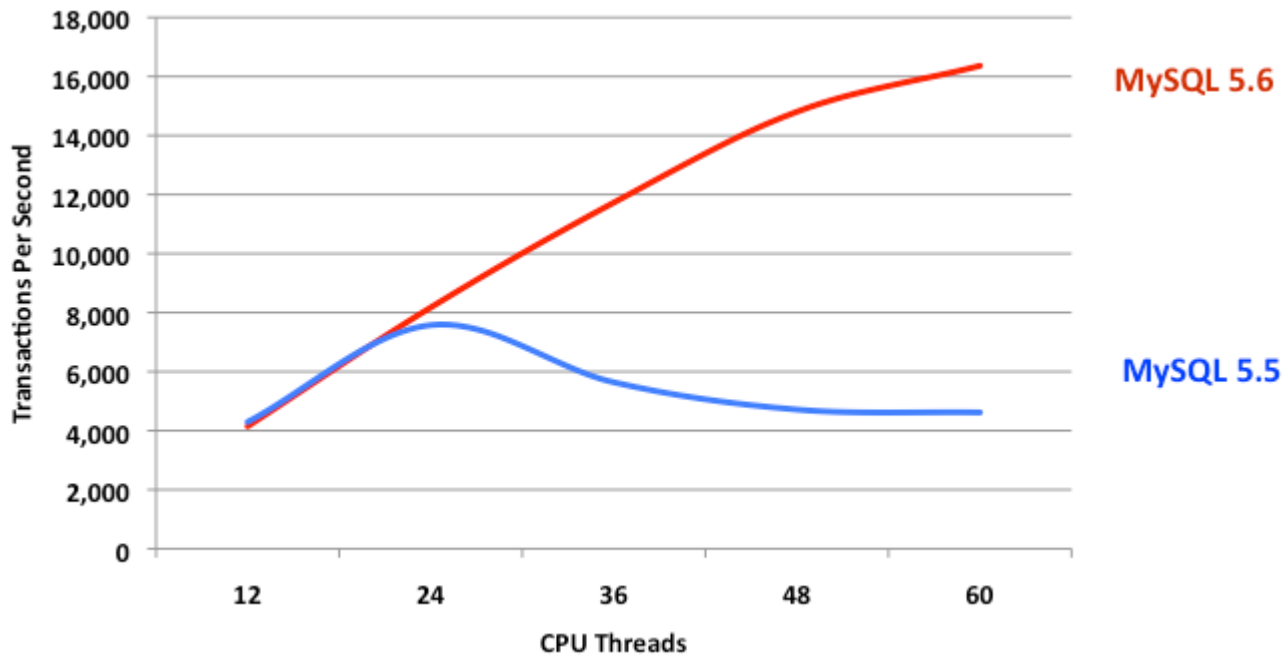
Performance Schema

- New Instrumentation
  - Statements/Stages
  - Table and Index IO
  - Table Locks
  - Users/Hosts/Accounts
  - Network IO
- New Features
  - Show Contents of Host Cache
  - New Summary Tables
  - Reduced Overhead
  - On By Default

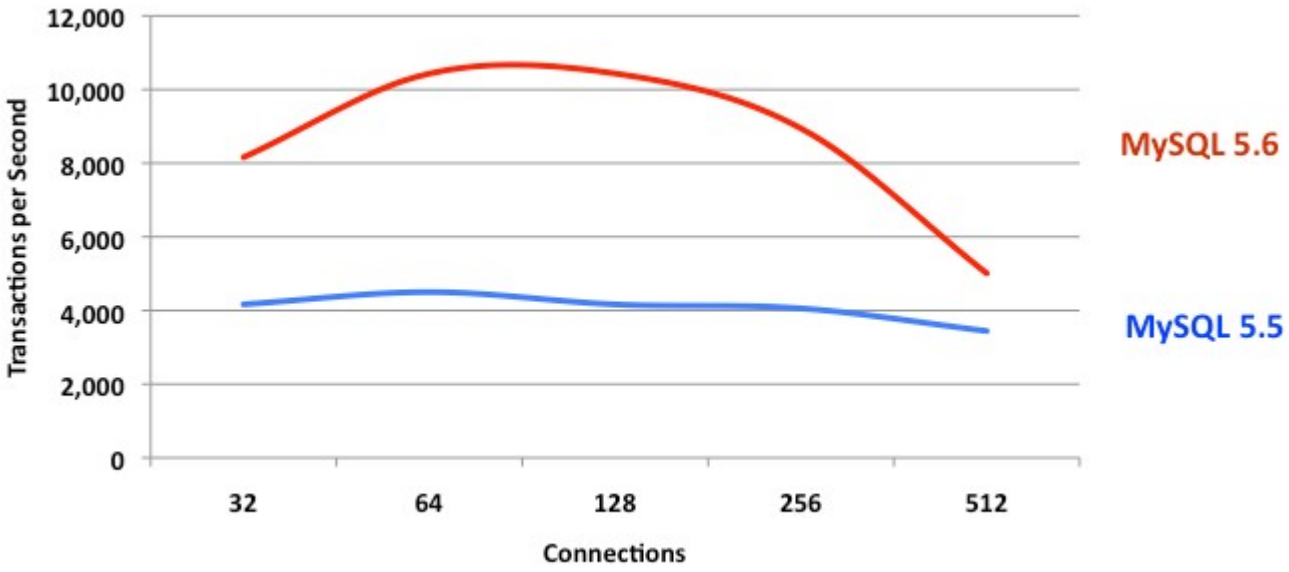
sysbench read/write Benchmark



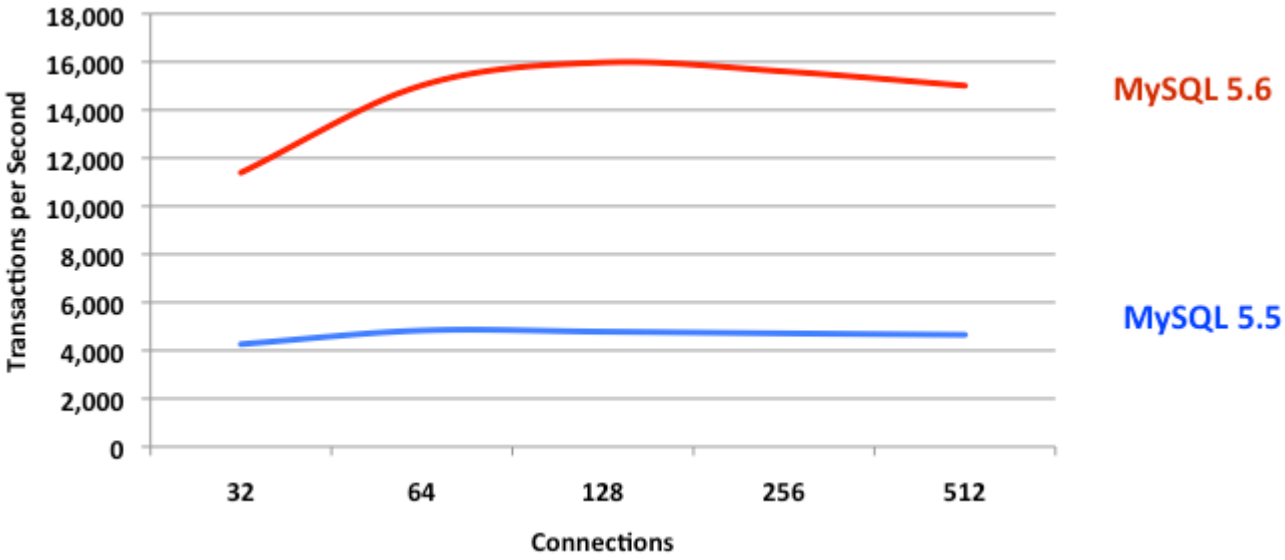
sysbench read-only Benchmark



sysbench read/write Benchmark



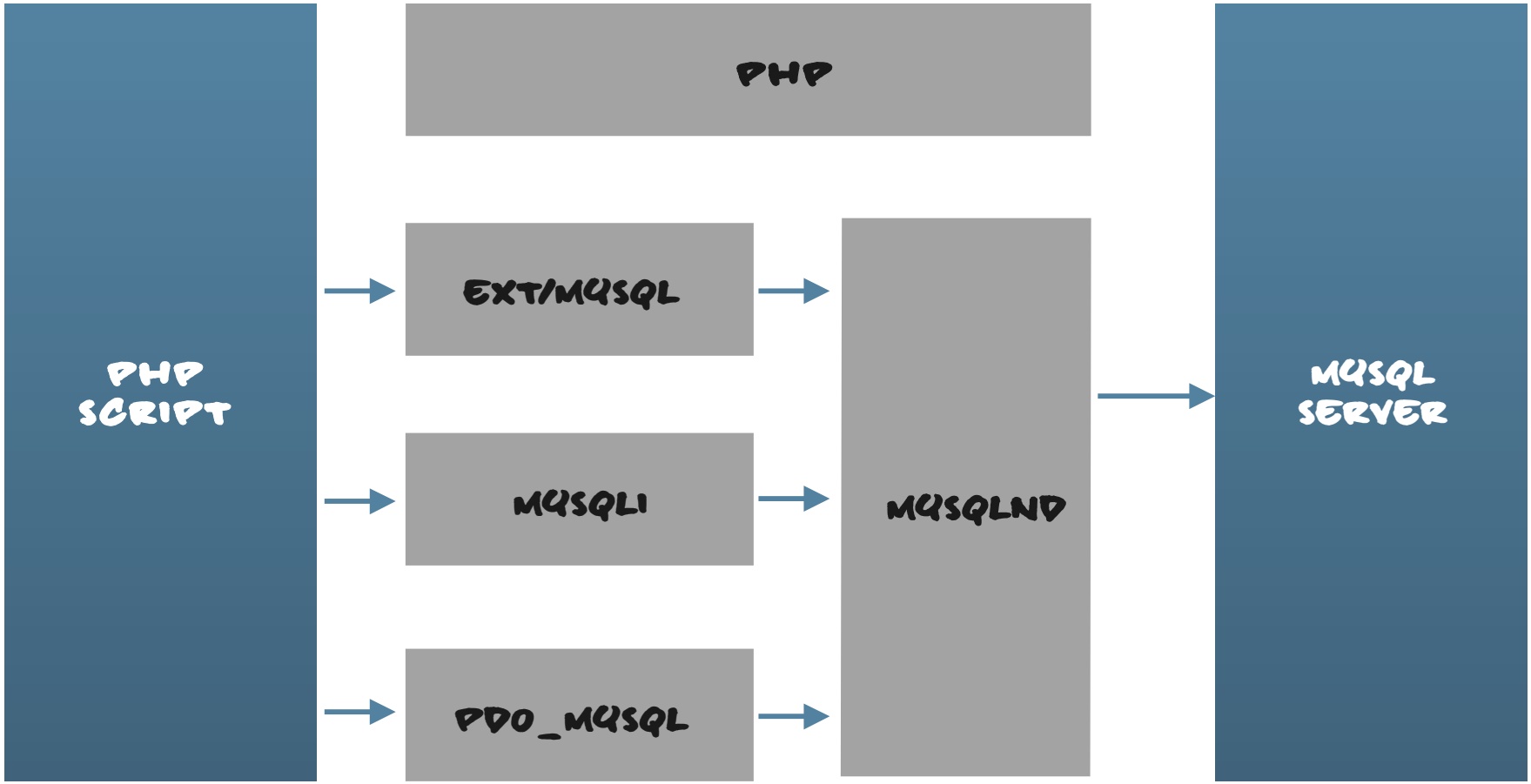
read-only Transactions



All benchmarks from
http://dev.mysql.com/tech-resources/articles/mysql-5.6.html

PHP and MySQL

PHP's MySQL Architecture



ext/mysql

The old and outdated

- One of the first PHP extensions ever
- Actively maintained with PHP 4
  - No new features in PHP 5
  - Exception: Added mysqlnd support with PHP 5.3
  - Deprecated with PHP 5.5
- Bug fixing only
- Best documented database extension
  - Tons of books, tutorials, ...
- Missing support for many MySQL features
  - Prepared statements, Queries with multiple result sets (stored procedures), compression, encryption, full charset support, ..

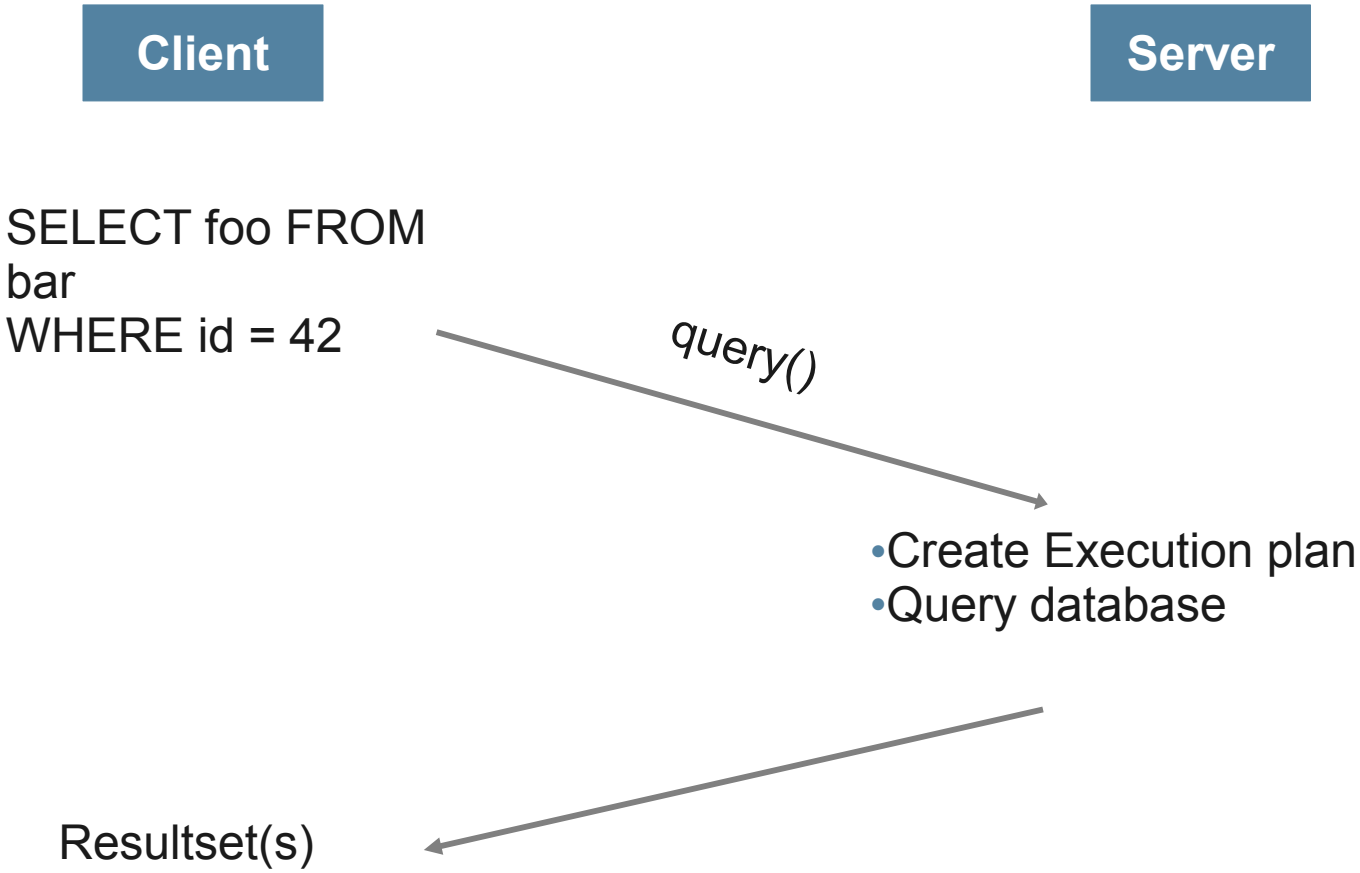
mysqli
The Improved MySQL Extension

- Full support for all MySQL features
  - Stored Procedures
  - Prepared Statements
  - Encryption (SSL)
  - Compression
  - Charsets
  - ...
- Actively developed, maintained and supported by Oracle

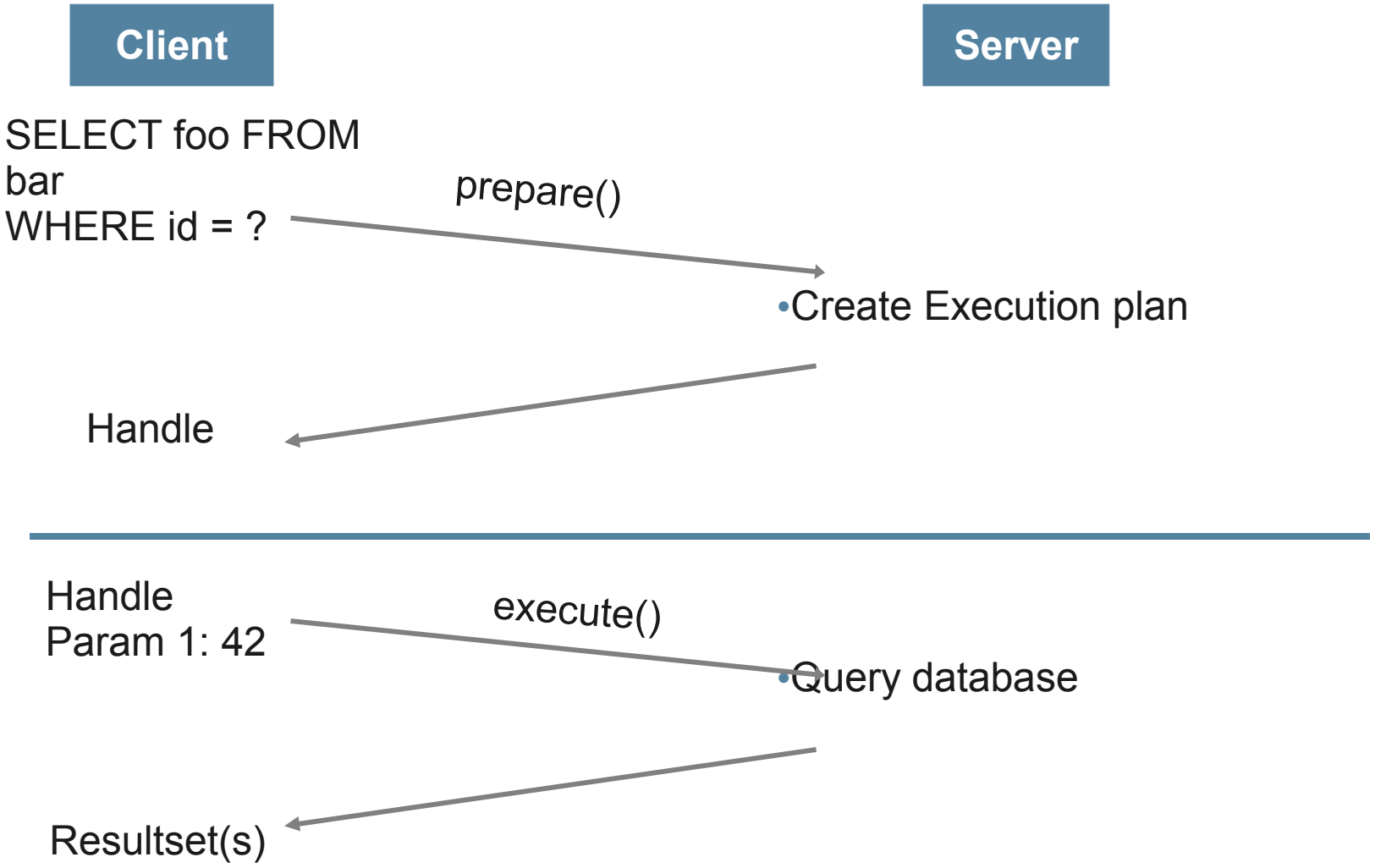
pdo_mysql
The MySQL PDO Driver

- “The PHP Data Objects (PDO) extension defines a lightweight, consistent interface for accessing databases in PHP.” http://php.net/intro.pdo
- Lowest common denominator
- PHPish API

Intermezzo: Prepared Statements



Intermezzo: Prepared Statements



PDO – Broken by Design

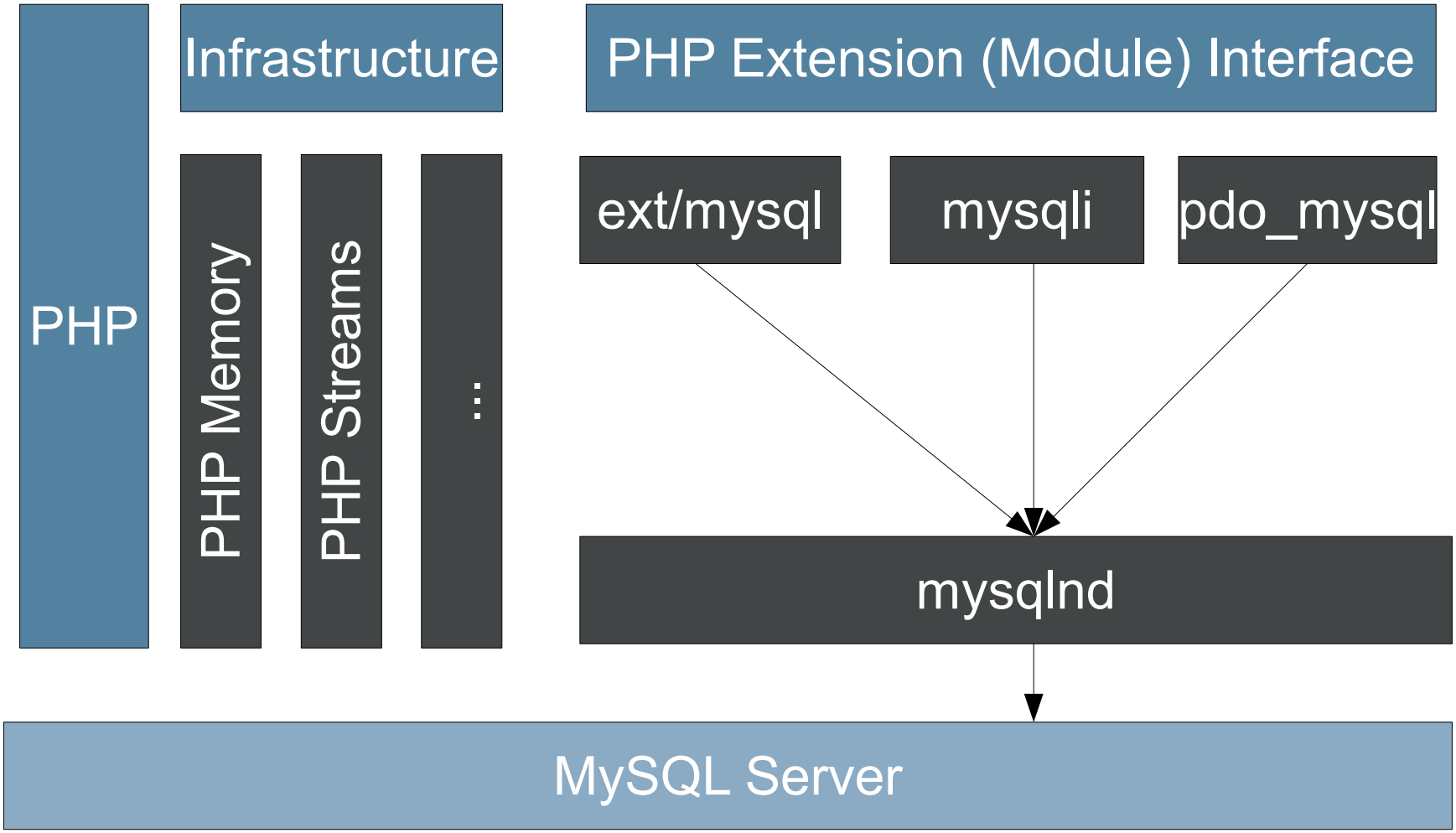
```
<?php
$pdo = new PDO(“mysql:host=localhost;dbname=test”,
        “user”, “password”);

$query = $pdo->prepare(
                    “SELECT id FROM table LIMT ?, ?”);
$query->bindValue(1, $_GET[“offset”]);
$query->bindValue(2, $_GET[“limit”]);

$query->execute();
?>
```

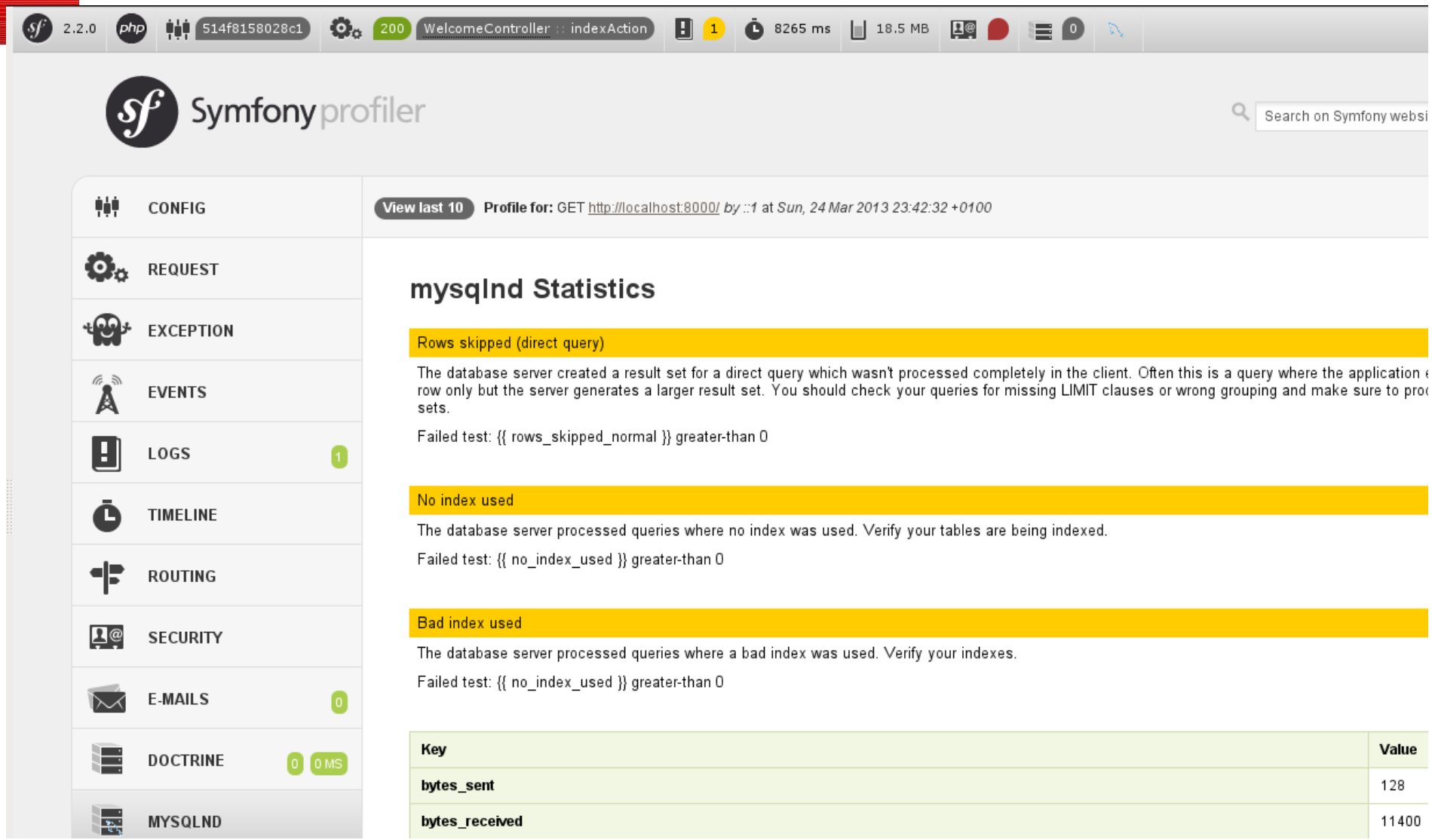
1064 You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version for the right syntax to use near "1', '2"

API Choice

- **mysqli**
  - Support for all MySQL features
  - Best support / stability
  - Integration with existing applications / environments
- **PDO_MYSQL**
  - Simple applications supporting multiple databases
    - API-compatibility is often not enough, though
  - Integration with existing applications / environments

mysIqnd – The MySQL native driver





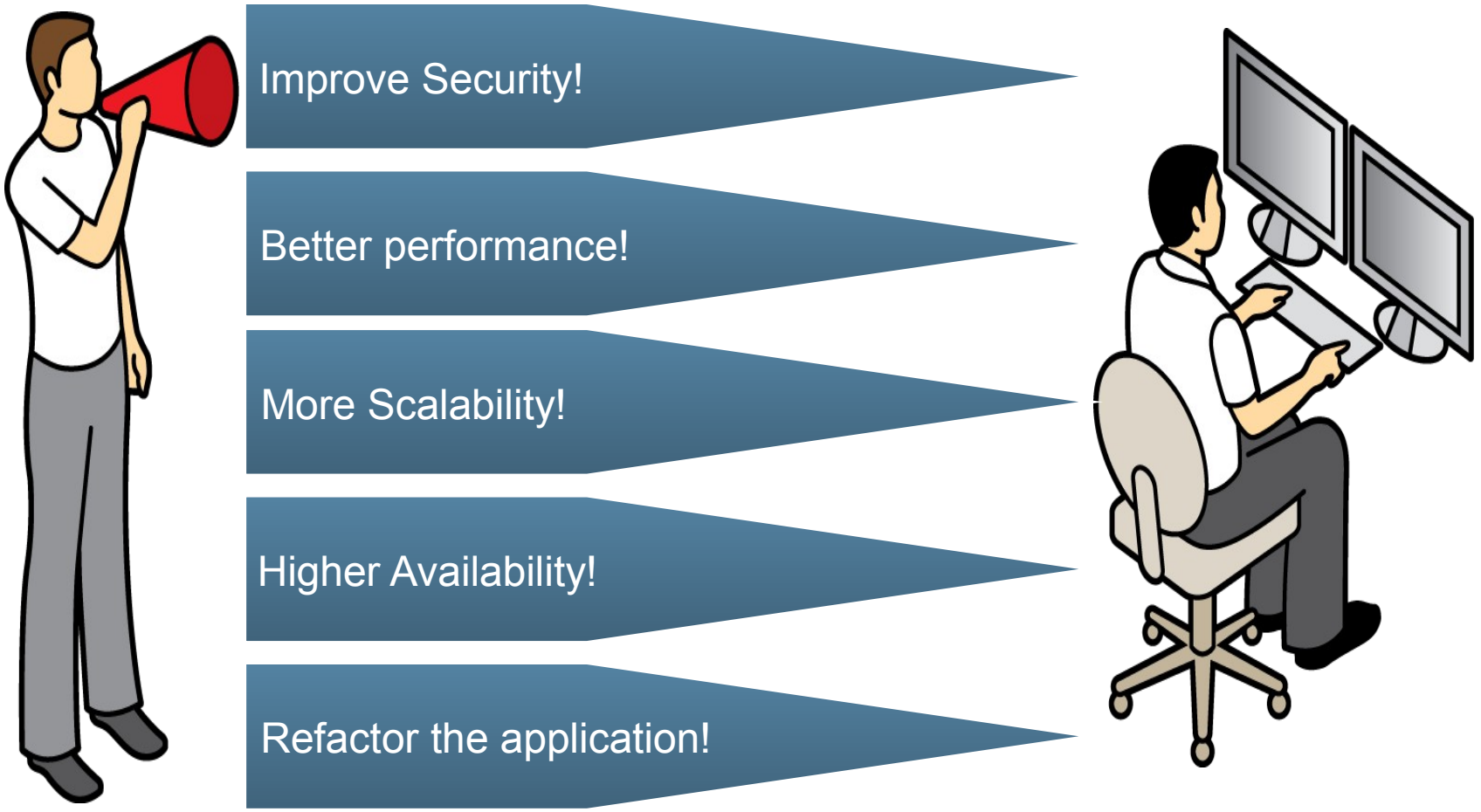
https://github.com/johannes/JSMysqlndAnalytics
https://github.com/johannes/JSMysqlndBundle

PHP Today

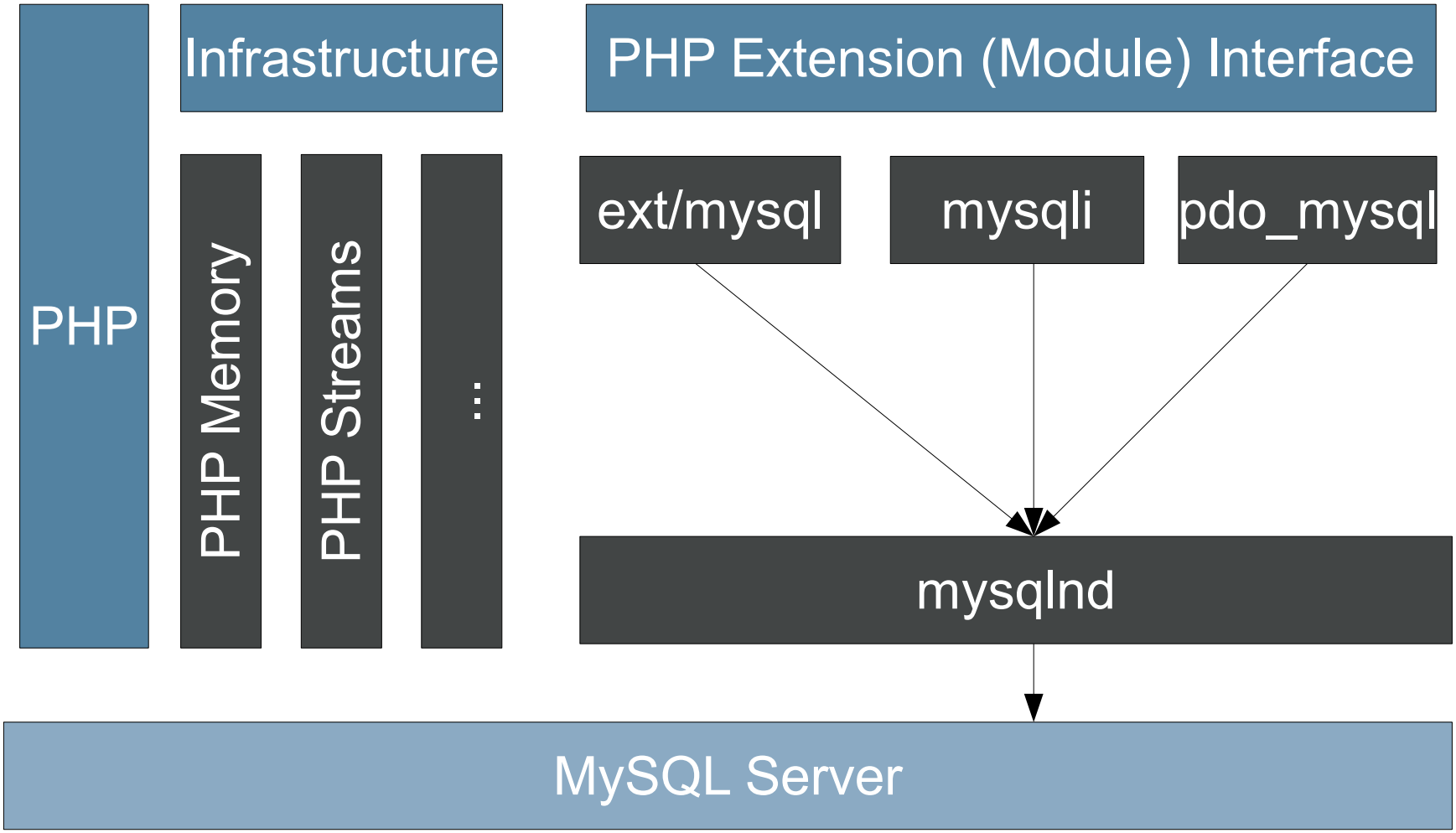
Three kinds of PHP Users

Users of existing Applications	Quick'n'Dirty Applications	Framework-based Applications
Take existing applications (i.e. Wordpress, phpBB, moodle, ...) Little customization only Often hard to scale	Ad-hoc developed applications Code often mixture of PHP with embedded SQL etc. Hard to scale/adapt to new situations	Custom applications built on i.e. Symfony, Zend Framework or CakePHP Built on DB abstraction Layers Abstractions hide full power of MysQL

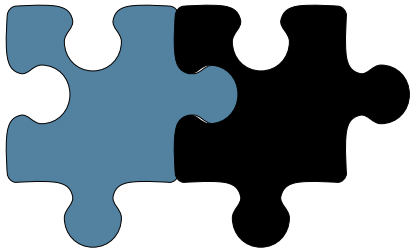
On-Going Demands



Solving these requests



mysqlnd in depth



mysqlnd_result:fetch_row

mysqlnd:query

mysqlnd:prepare

mysqlnd

mnd::allocate

mysqlnd_net:send

mysqlnd_net:read_result

mysqlnd plugins

myslqnd Plugins

- Loaded as regular PHP extension
  - Written in C
- Hook into mysqlnd in any level
  - From high-level APIs (query, prepare fetch_row, etc)
  - To low-level (network IO, memory allocation)
- Can be 100% transparent
- Can provide userspace APIs which can work with ext/mysql, mysqli and PDO_mysql instances

Writing mysqlnd Plugins

1. Create a PHP extension stub (i.e. Using ext_skel)
2. In MINIT register to mysqlnd
3. Identify APIs to hook
4. Register hooks in MINIT
5. Code away

- Documentation available: http://php.net/mysqlnd.plugin

mysqlnd_uh
If you don't like writing C

uh = user handler

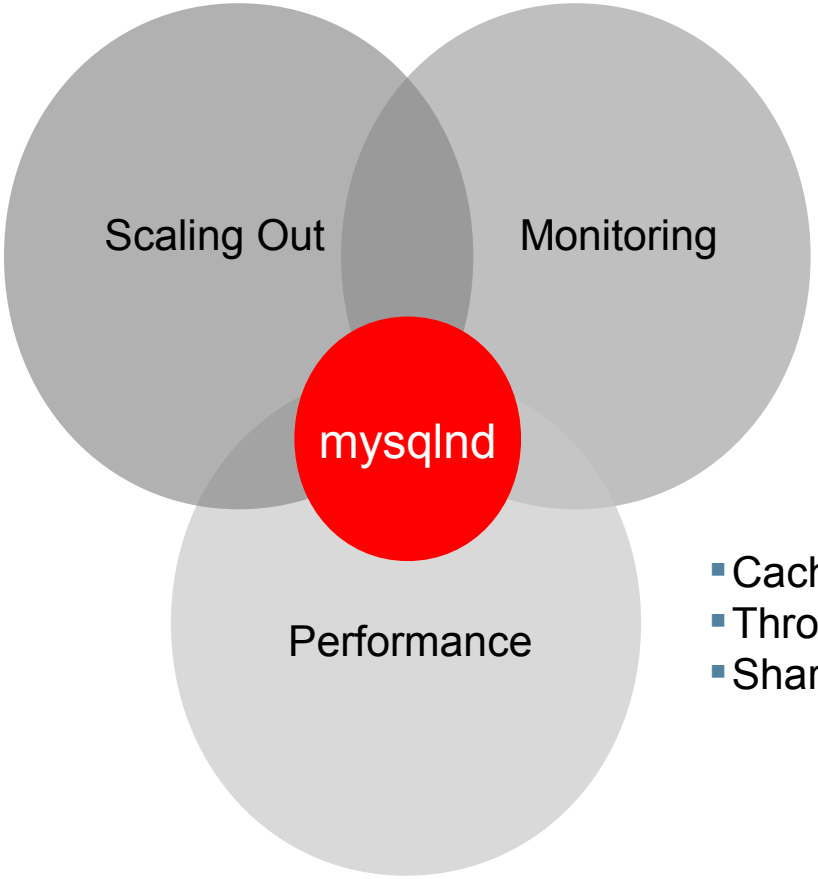
```
<?php
class conn_proxy extends MysqlndUhConnection {

    public function query($res, $query) {

        debug_print_backtrace();
        return parent::query($res, $query);
    }
}

mysqlnd_uh_set_connection_proxy(new conn_proxy());
?>
```

mysqlnd_uh

- Exports many of the C-level APIs to PHP userland
- Developed by Community (David Soria Parra / Mayflower GmH) with some help from our team
- Allows to go quite deep, mistakes can have some bad effects
- PECL site: http://pecl.phpnet/mysqlnd_uh
- Documentation: http://php.net/mysqlnd_uh

What can Plugins be used for?

- Read/Write Splitting
- Failover
- Round-Robin



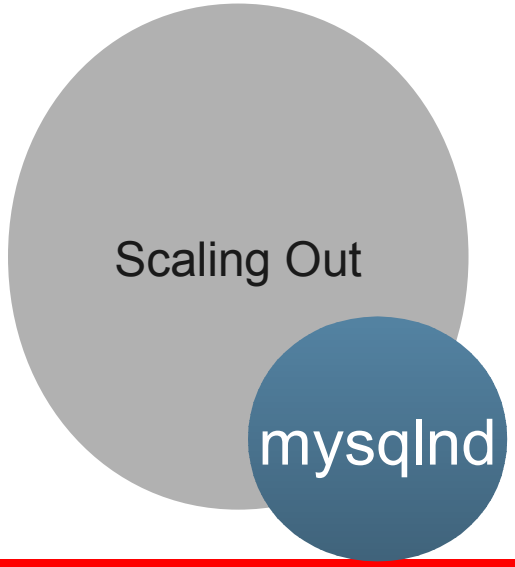
- Query Logging
- Query Analysis
- Query Auditing

- Caching
- Throttling
- Sharding

mysqlnd_ms
mysqlnd replication and load balancing plugin

ms = master/slave

- Fully transparent
- Load Balancing
  - Distribute queries based on a user-defined strategy to different MySQL servers
  - i.e. round-robin, random, user callback defined
- Read/Write-Splitting
  - Automatically detect reads and writes
  - Reads go to slaves
  - Can be disabled in multi-master environments (MySQL Cluster)



mysqlnd_ms
mysqlnd replication and load balancing plugin

ms = master/slave

- Global Transaction ID
  - Picks up-to-date slave for reads
  - Benefits from MySQL 5.6 GTID Features
  - Can provide GTID emulation
- Service level
  - Applications can request eventual consistency

mysqlnd_ms
Getting Started - Configuration

php.ini:

```
extension=mysqlnd_ms.so
mysqlnd_ms.enable=1
mysqlnd_ms.config_file=/path/to/mysqlnd_ms_plugin.json
```

mysqlnd_ms_plugin.json:

```
{
  "myapp": {
    "master": { "master_0": {
       "host": "192.168.2.27", “port”:”3306”
    } },
    "slave": {  "slave_0": {
       "host": "localhost", "socket": "/tmp/mysql.sock"
    } }
  }
}
```

mysqlnd_ms
Getting Started - Usage

```
$mysqli = new mysqli("myapp", "username", "pwd", "db");

$pdo = new PDO('mysql:host=myapp;dbname=db', 'u', 'pwd');

$mysql = mysql_connect("myapp", "username", "password");
```

mysqlnd_ms
More Information

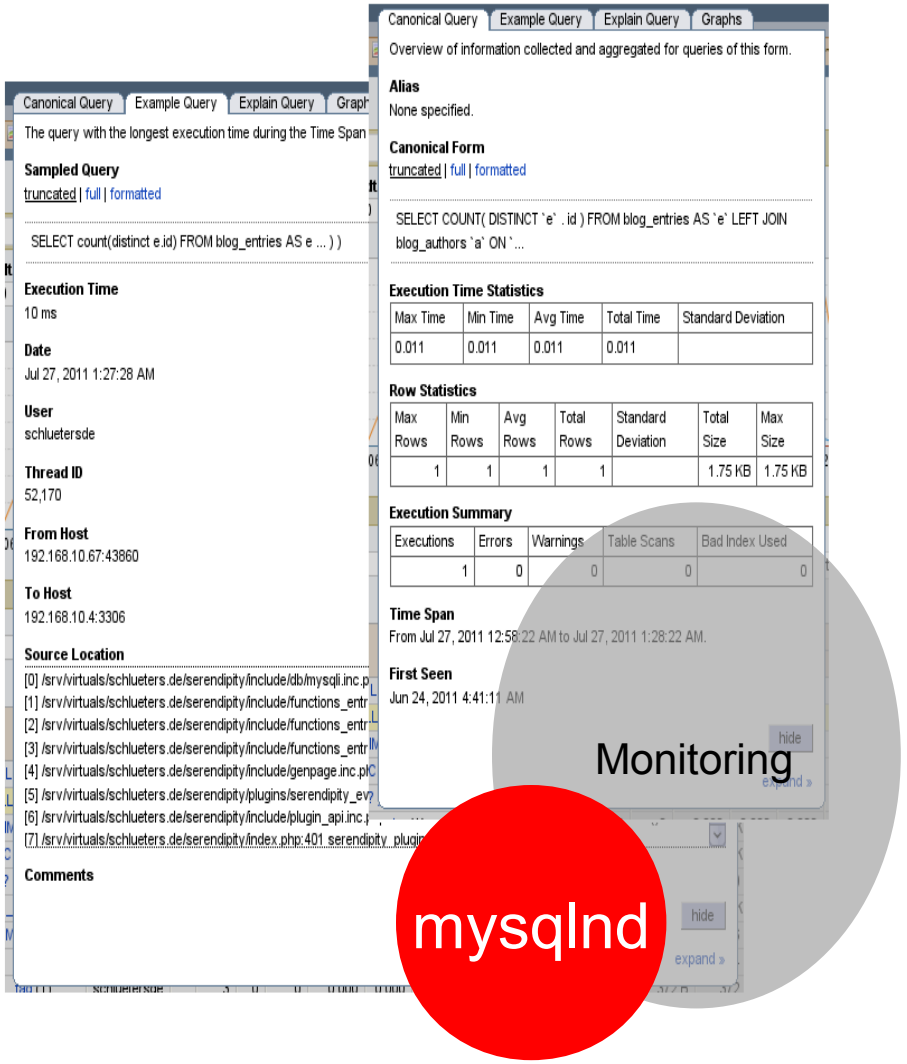
- PECL Website: http://pecl.php.net/mysqlnd_ms
- Documentation: http://php.net/mysqlnd_ms
- Ulf's Blog: http://blog.ulf-wendel.de/

MySQL Enterprise Query Analyzer

The MySQL Enterprise Monitor is the tool for monitoring MySQL

Query Analyzer provides you with in-depth monitoring of query performance

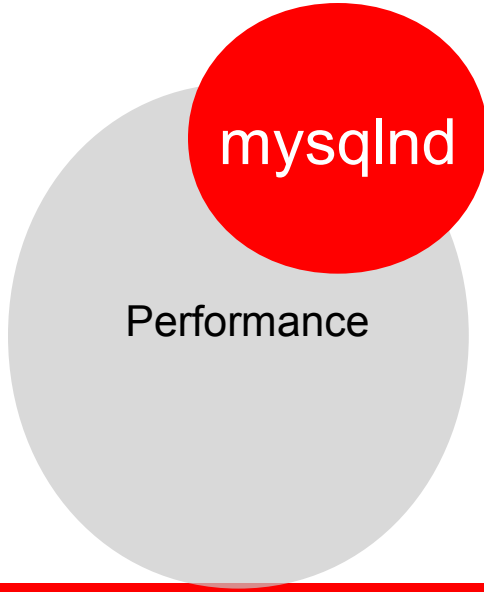
http://mysql.com/enterprise



mysqlnd_qc
Client-Side Query Result Caching

qc = query cache

- Transparent
- Flexible invalidation
  - Time-to-live
  - User-defined
- Flexibility with various backends
- Built-In Slam-Defense

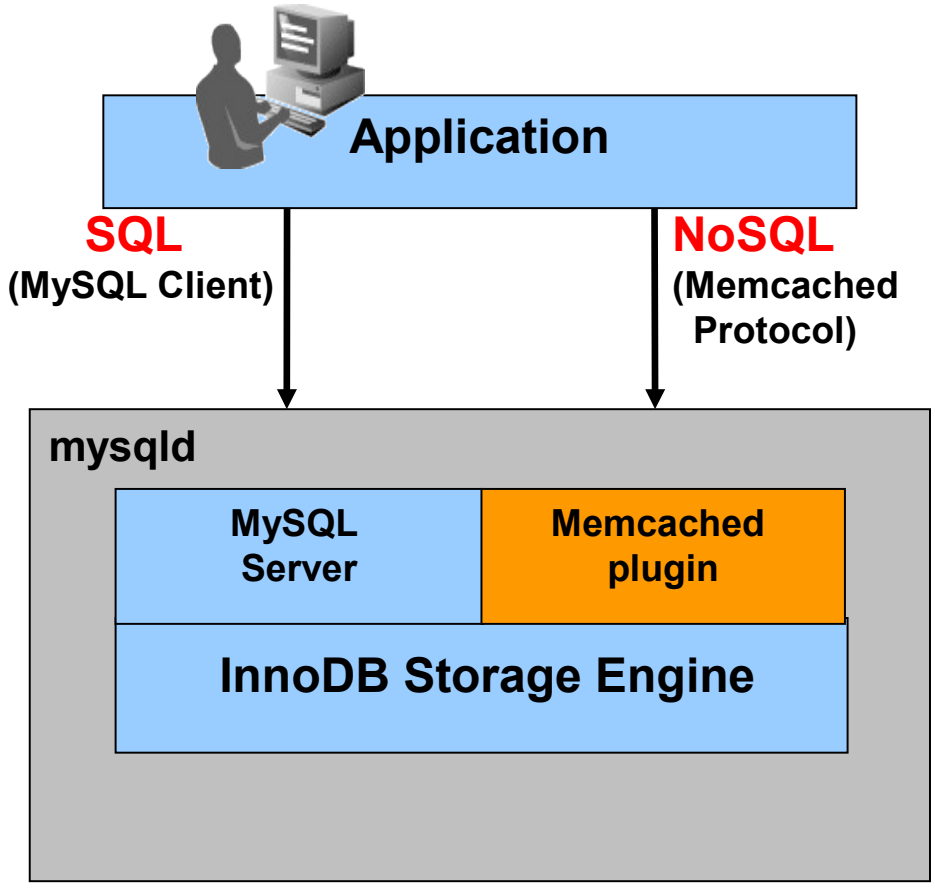


mysqlnd_qc
Storage Backends

mysqlnd_qc
More Information

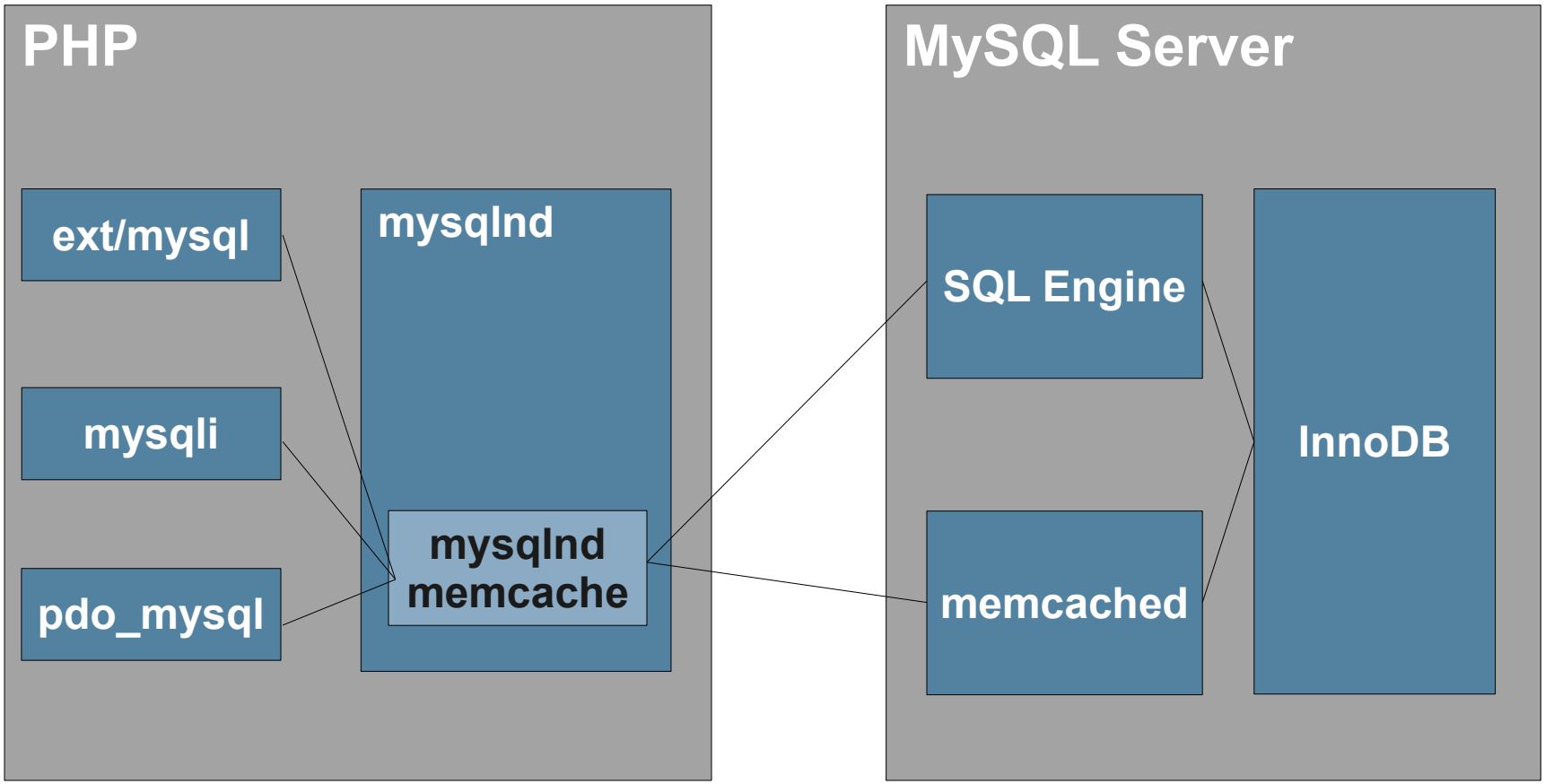
- PECL Site: http://pecl.php.net/mysqlnd_qc
- Documentation: http://php.net/mysqnd_qc
- Ulf's Blog: http://blog.ulf-wendel.de
- Webinar Recording: http://www.mysql.com/news-and-events/on-demand-webinars/display-od-668.html

MySQL 5.6
NotOnlySQL: Memcached API



- Fast, simple access to InnoDB
  - Accessed via Memcached API
  - Use existing Memcached clients
  - Bypasses SQL transformations
- NotOnlySQL access
  - Memcached for key-value operations
  - SQL for rich queries, JOINs, foreign keys, etc.
- Implementation
  - Memcached daemon plug-in to mysqld
  - Memcached protocol mapped to the native InnoDB API
  - Shared process space for ultra-low latency
  - Additional implementations in future DMs

mysqlnd Memcache plugin mysqlnd_memcache



mysqlnd Memcache plugin Example

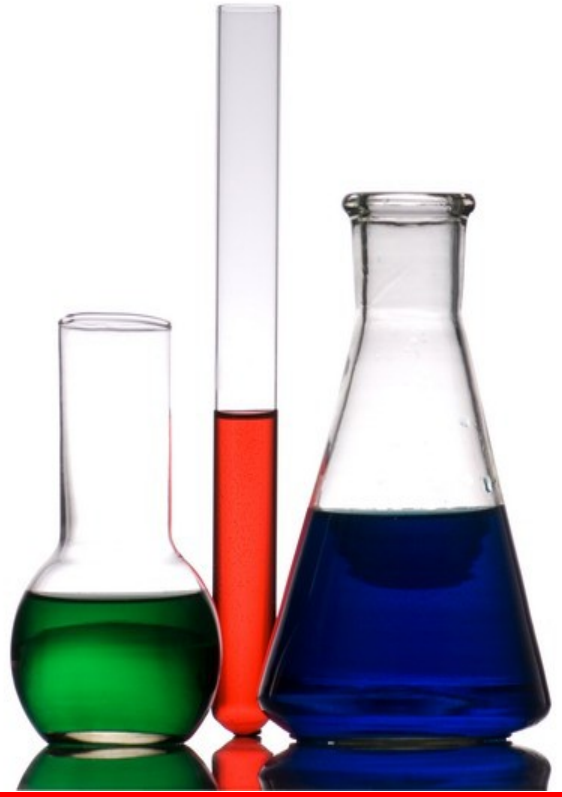
```
$mysql = new mysqli(“localhost”, “user”, “pass”, “test”);
$memcache = new memcached();
$memcache->addServer(“localhost”, 11211);

mysqlnd_memcache_set($mysql, $memcache);

$mysql->query(“SELECT name FROM table WHERE id = 1”);
```

mysqlnd_memcache
More Information

- PECL Site: http://pecl.php.net/mysqlnd_memcache
- Documentation: http://php.net/mysqlnd_memcache
- My Blog: http://schlueters.de/blog/

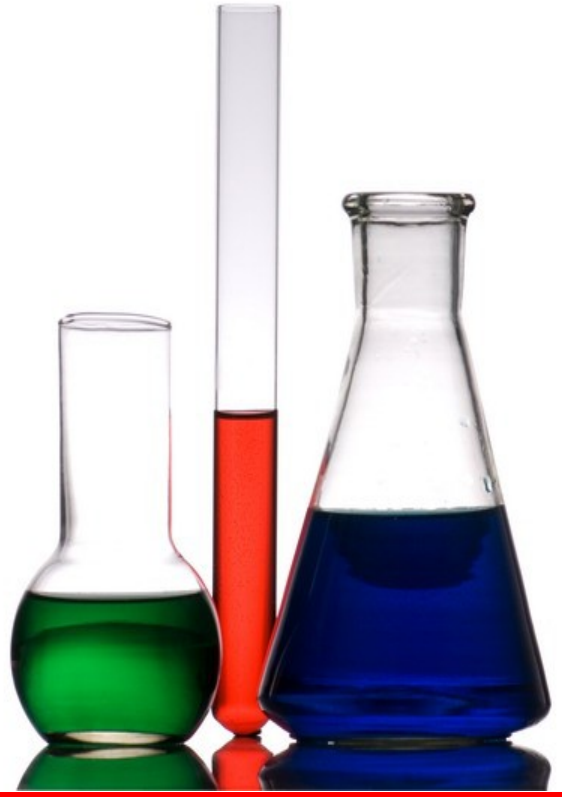
One more thing ...

- All previously mentioned plugins are available as stable/GA plugins for PHP
- But we have more!



Experimental Plugins

- mysqlnd_mc
  - Multi Connect, fetch data from multiple servers at once
- mysqlnd_pscache
  - Prepared statement handle cache for persistent connections
- mysqlnd_sip
  - SQL injection detection
- mysqlnd_mux
  - Multiplexing connections



… and now a quick look into the future ...

MySQL 5.7.1 DMR

- Support for stacked diagnostic areas according to SQL standard
- Online `RENAME INDEX`
- Online enlarge `VARCHAR` size
- Improved InnoDB temporary tables
- Introduced tablespace for temporary tables
- Non-blocking `SHOW SLAVE STATUS`
- Option to ignore errors with RBR
- `mysqlbinlog --rewrite-db` option

→ Fetch it from dev.mysql.com

MySQL Cluster 7.3 DMR 2

- Foreign Key Support
  - Works with SQL …
  - … and NoSQL APIs (C++, Java, JPA, HTTP/REST, Node.js AP, Memcached)
- Auto-Installer
  - Graphical setup and configuration of MySQL Cluster
- Based on MySQL 5.6
  - Improved SQL optimizer
  - Improved Replication
  - ...
- NoSQL Node.js API

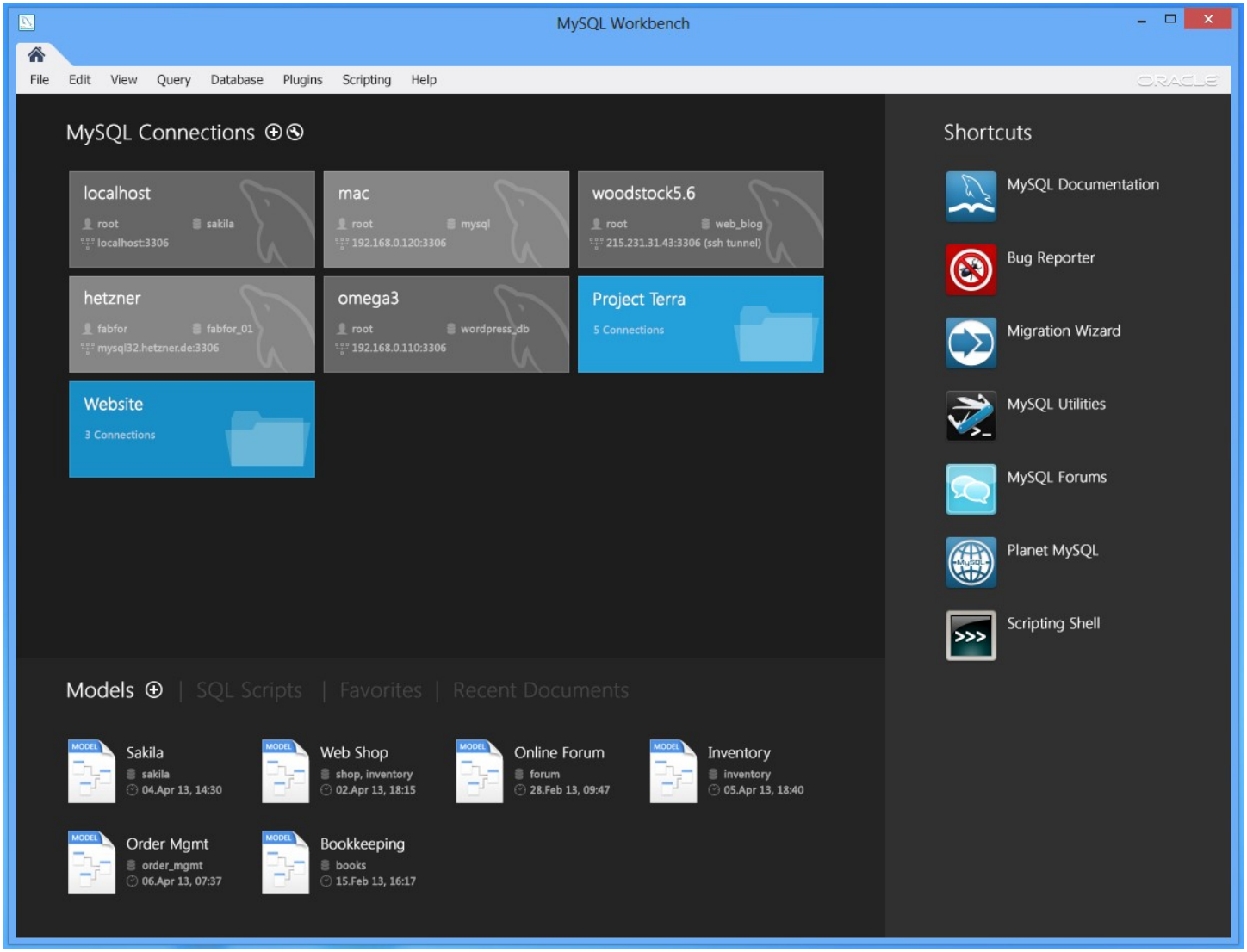
MySQL Hadoop Applier

- Real-Time replication of events from MySQL to Hadoop
- Connects to binary log, writes events to HDFS
- Each database mapped to a hive data warehouse directory

- See dev.mysql.com

- Fetch it from labs.mysql.com

MySQL Workbench 6.0



Learn More

- Tomas Ulin's (VP MySQL Engineering) Keynote at Percona Live: http://www.youtube.com/watch?v=OpHTV59I1gs (46:23, Apr 23rd)
- Tomas' Blog: (Monthly summary of developments) http://insidemysql.com/
- MySQL Development Zone: http://dev.mysql.com
- MySQL Connect Conference: (September 21-23, San Francisco) http://www.oracle.com/mysqlconnect/index.html

Thanks for Your attention!

Questions?

Johannes Schlüter
johannes.schlueter@oracle.com
Twitter: @phperror
http://schlueters.de/blog/

http://joind.in/talk/view/8441

Hardware and Software

ORACLE

Engineered to Work Together


ORACLE®