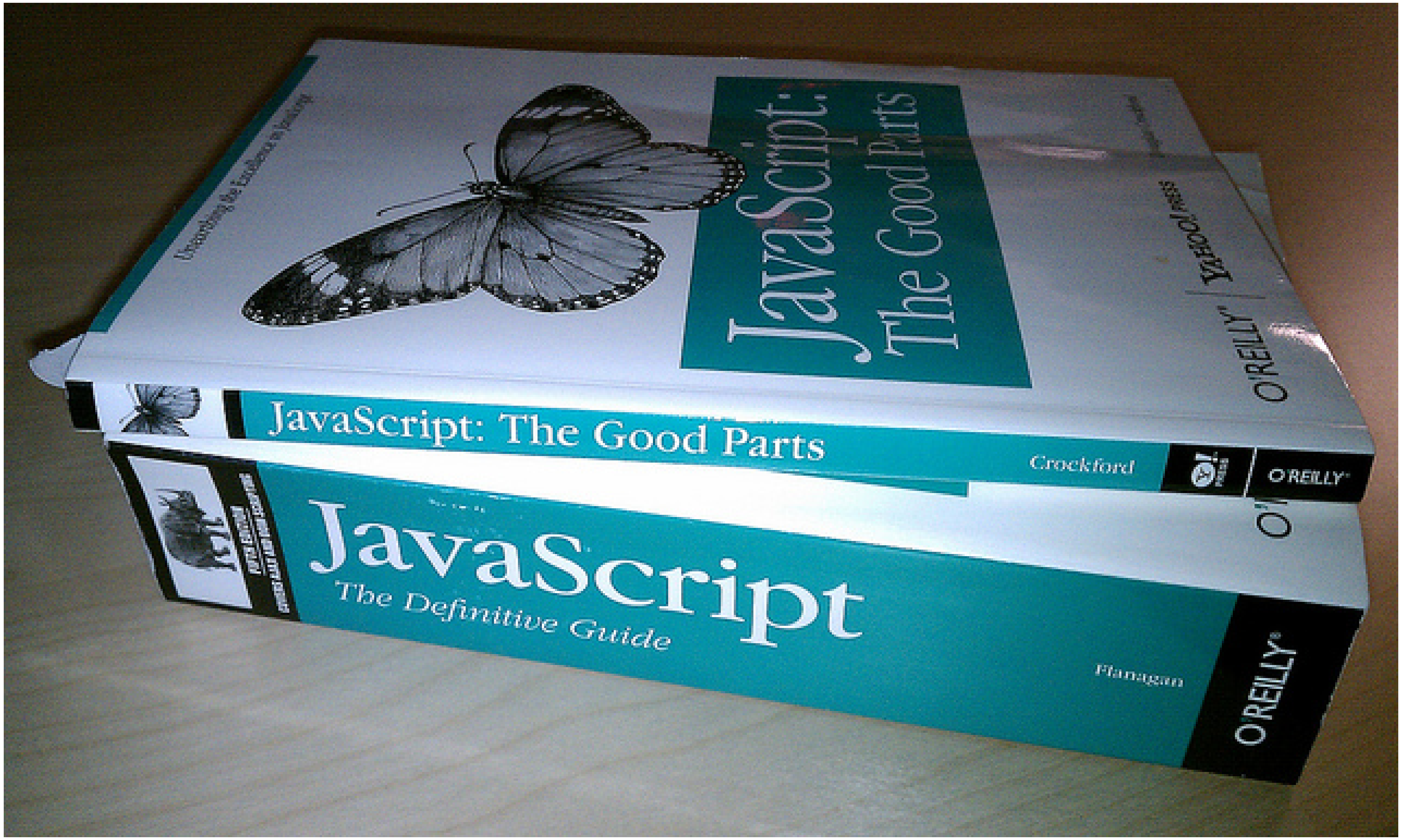
ZABBIX 2014 Conference

Zabbix 3.0+

Where do we go

Agenda


JavaScript: The Good Parts
Crockford
O'REILLY
YAHOO! PRESS
JavaScript
The Definitive Guide
Flanagan
O'REILLY

What we are building?

- Monitoring platform you can trust
    - Not limited to IT only
- Trust: reliable, stable, correct, predictable
- Suitable for environments of any size

Let's go back into 1998

Will we meet Perl there?

ZABBIX 2014 Conference

Toolbox from 1998

- Perl was used initially, then switched to:

  C language for all critical parts

  PHP language for the WEB interface

  SQL back-end

ZABBIX 2014 Conference

C language

Properties:

+ Low level language

+ Efficient code: fast

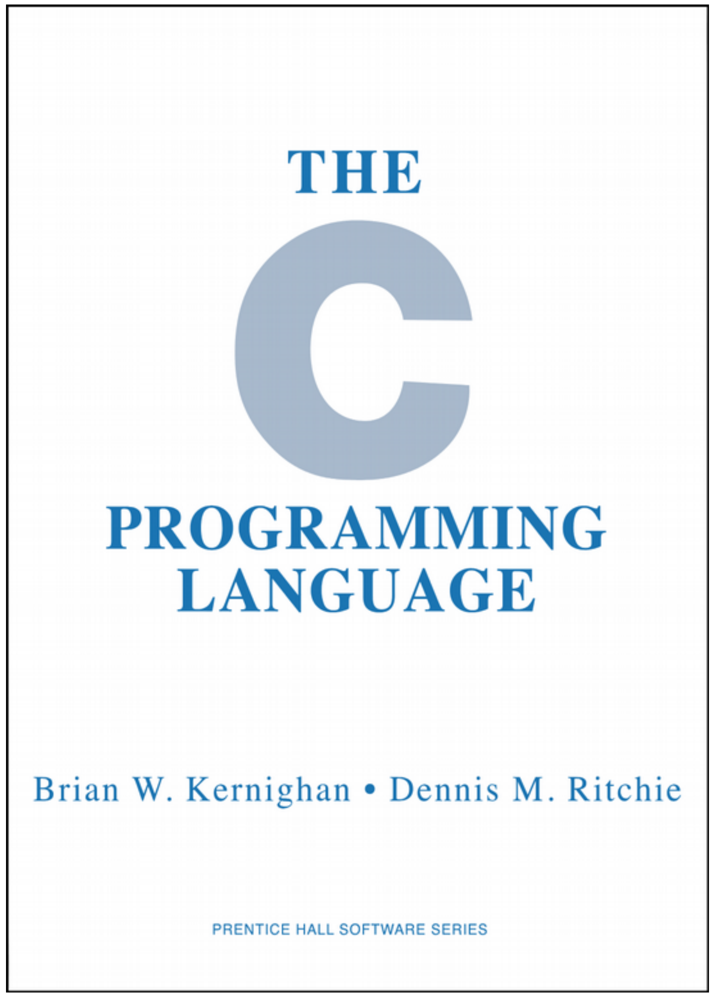
+ Lowest resource usage (CPU/mem)

+ Almost no dependencies

+ Write once compile & run everywhere

- Slower development

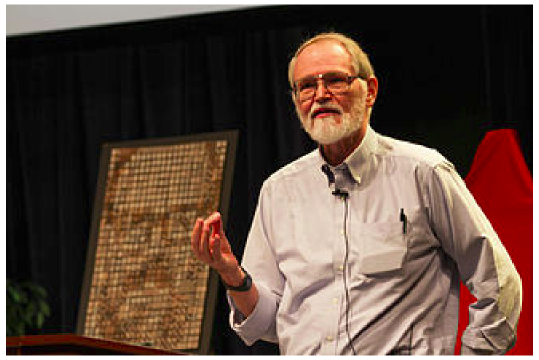
- Memory, lock, pointer related errors



THE C PROGRAMMING LANGUAGE

Brian W. Kernighan • Dennis M. Ritchie

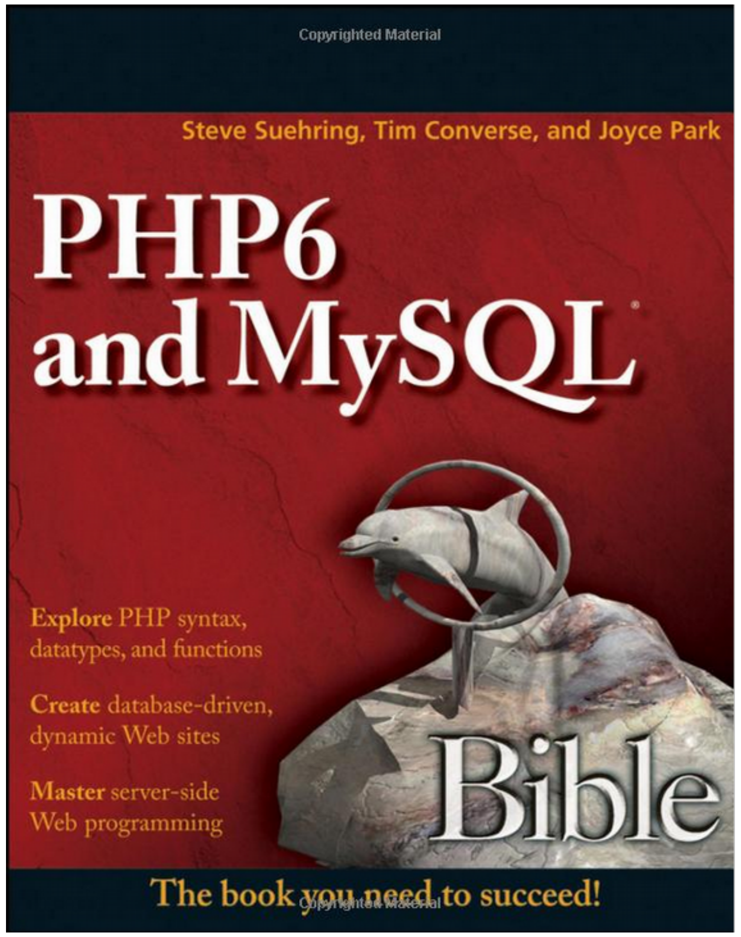
PRENTICE HALL SOFTWARE SERIES



ZABBIX 2014 Conference

PHP language

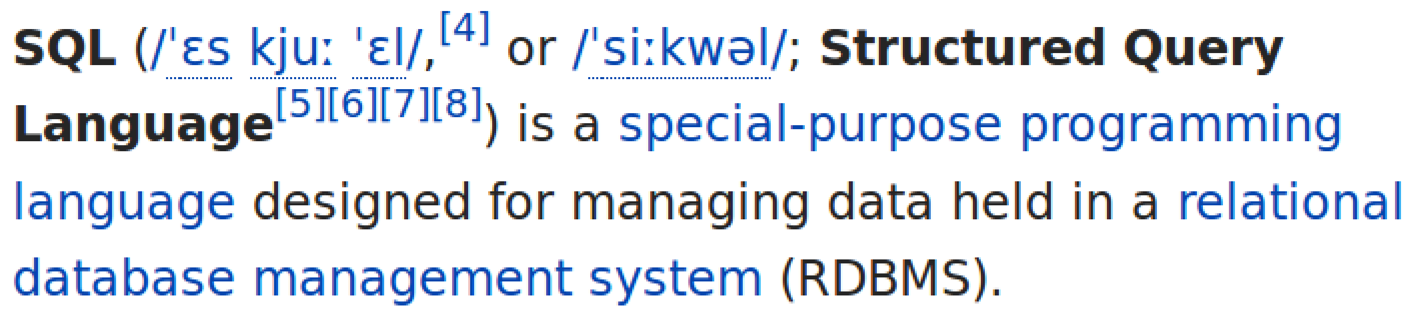
+ Fast learning curve
+ Available for all platforms
+ Very actively developed nowadays
- Dynamically typed
- Discipline is required for good code
- Interpreted: errors tend to come up at runtime


Copyrighted Material
Steve Suehring, Tim Converse, and Joyce Park
PHP6 and MySQL
Explore PHP syntax, datatypes, and functions
Create database-driven, dynamic Web sites
Master server-side Web programming
Bible
The book you need to succeed!

SQL back-end

MySQL, PostgreSQL, Oracle, DB2, SQLite

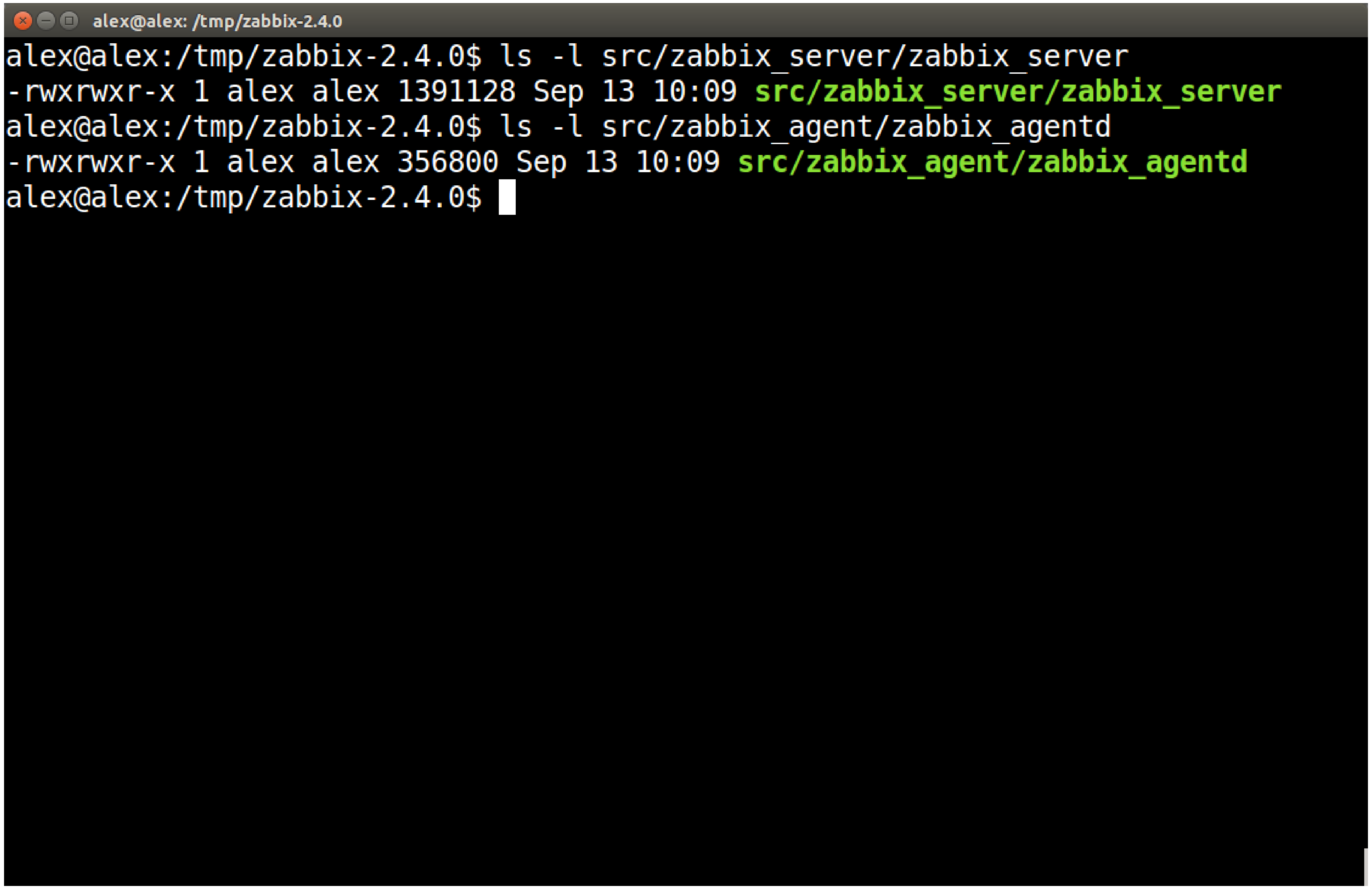
+ Transactional storage engine: consistency
+ Standard API: SQL
+ Easy to deploy
- Scalability
- High-availability



SQL (/ˈɛs kjuː ˈɛl/,[4] or /ˈsiːkwəl/; **Structured Query Language**[5][6][7][8]) is a special-purpose programming language designed for managing data held in a relational database management system (RDBMS).

How C, PHP, SQL affect Zabbix

- Zabbix is super compact



alex@alex: /tmp/zabbix-2.4.0

```
alex@alex:/tmp/zabbix-2.4.0$ ls -l src/zabbix_server/zabbix_server
-rwxrwxr-x 1 alex alex 1391128 Sep 13 10:09 src/zabbix_server/zabbix_server
alex@alex:/tmp/zabbix-2.4.0$ ls -l src/zabbix_agent/zabbix_agentd
-rwxrwxr-x 1 alex alex 356800 Sep 13 10:09 src/zabbix_agent/zabbix_agentd
alex@alex:/tmp/zabbix-2.4.0$
```

How C, PHP, SQL affect Zabbix

+ Almost no dependencies

+ Easy to maintain

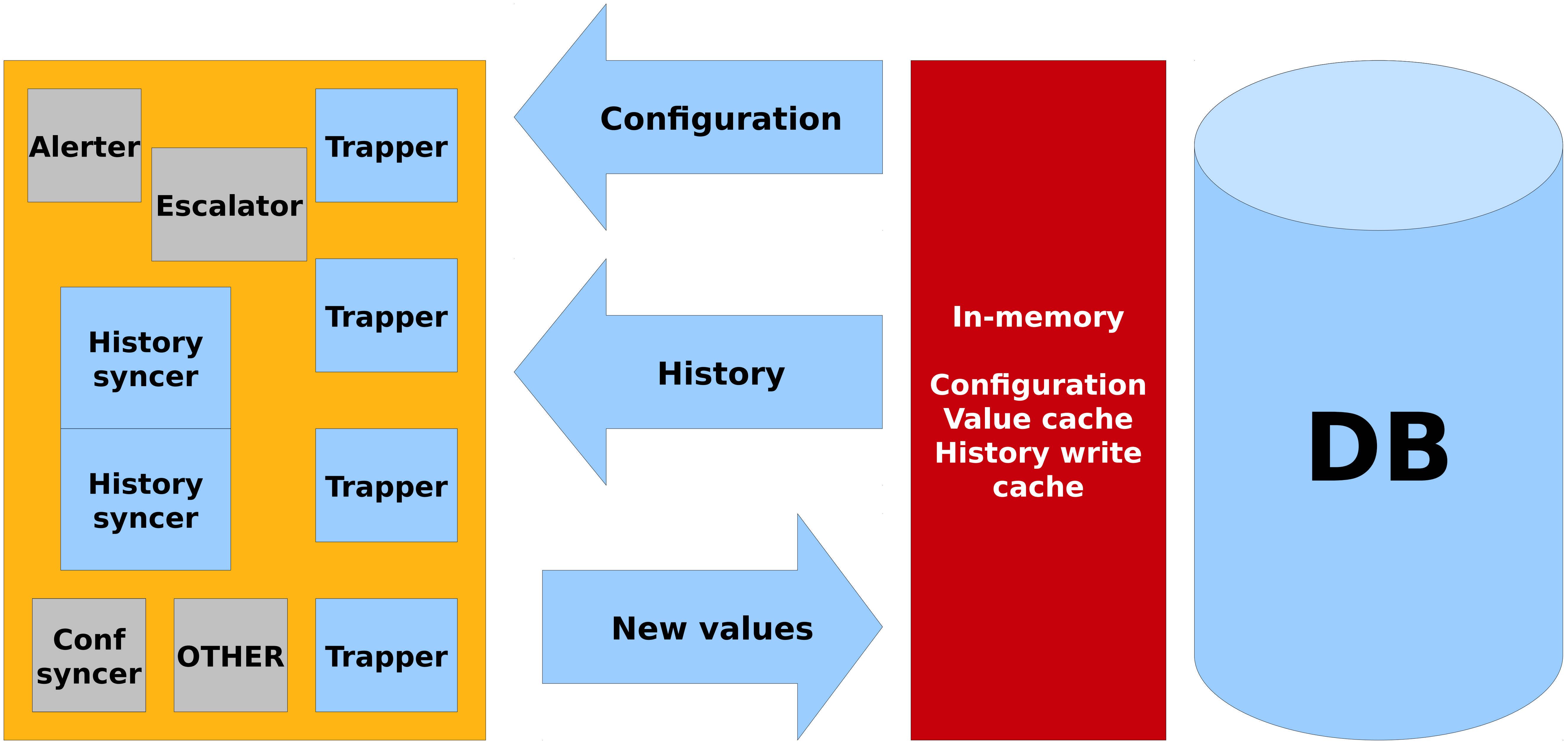
+ High-performance

+ Low resource usage

+ Supported on all Unix platforms

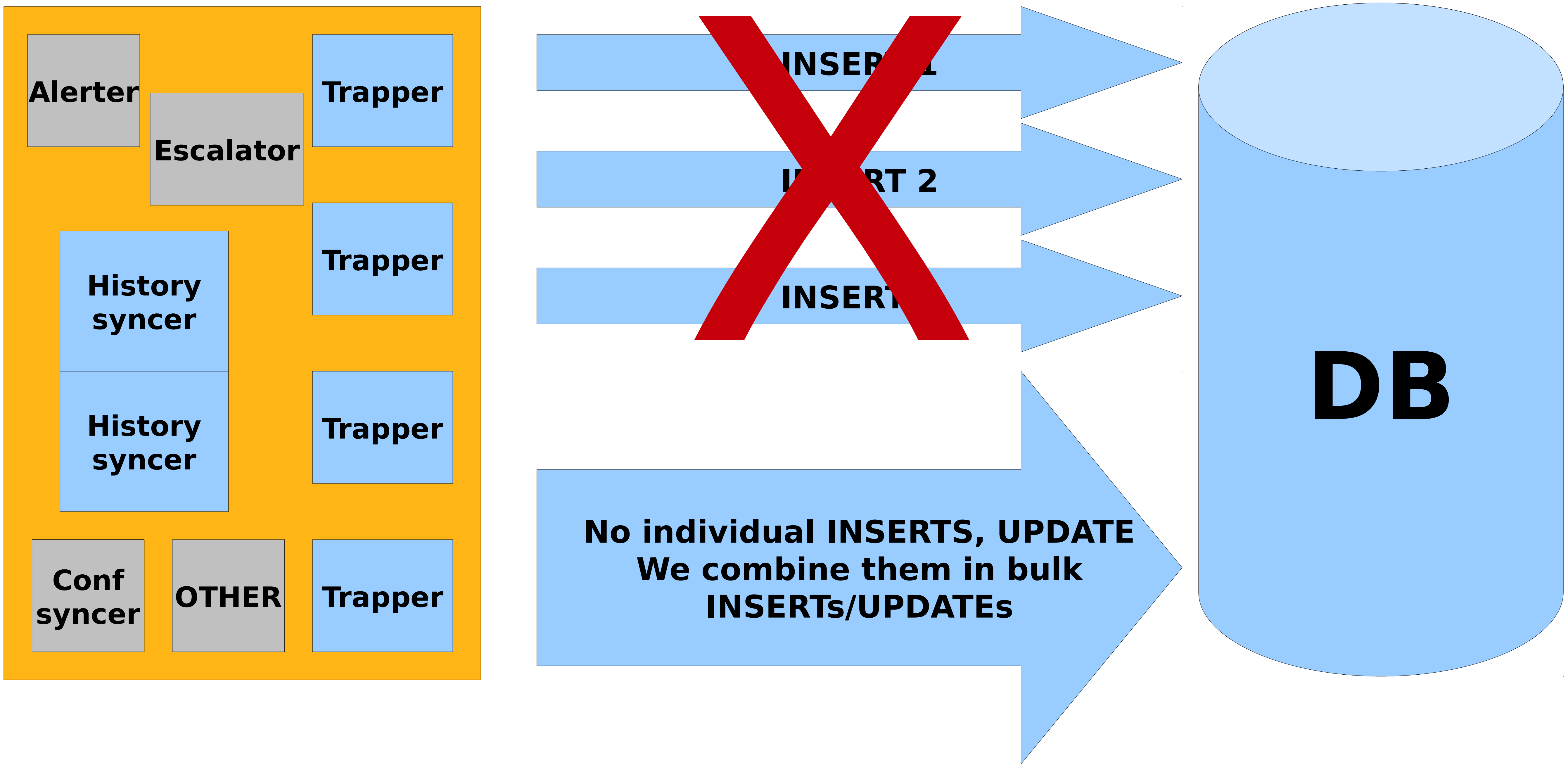
- Regressions and unfortunate issues (undefined variables)

Some techniques: cache



- used only for backend (Server, Proxy)

Some techniques: bulk operations



No individual INSERTS, UPDATE
We combine them in bulk INSERTs/UPDATEs

- used only for backend (Server, Proxy)

ZABBIX 2014 Conference

Architecture: the good parts

+ Good separation of logic: data collection, problem detection, alerting, visualization, API, etc

+ Multi process application: scales to all available CPU cores

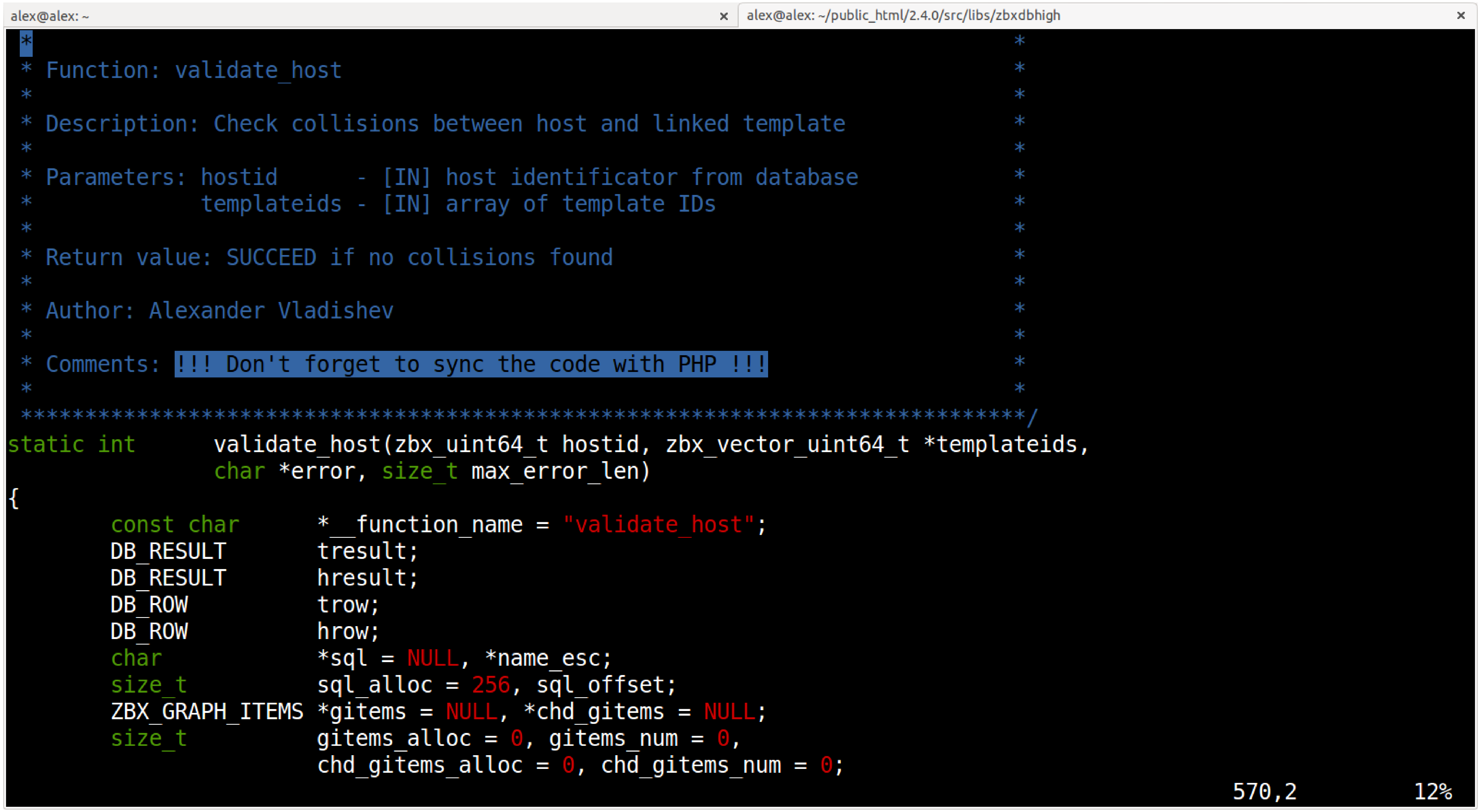
+ Data is always in consistent state

It was a good foundation in 1998

Is it now?

Present challenges

Code duplication (back-end, front-end)



alex@alex: ~ | alex@alex: ~/public_html/2.4.0/src/libs/zbxdbhigh

```
 *                                                                             *
 * Function: validate_host                                                     *
 *                                                                             *
 * Description: Check collisions between host and linked template              *
 *                                                                             *
 * Parameters: hostid      - [IN] host identificator from database             *
 *             templateids - [IN] array of template IDs                        *
 *                                                                             *
 * Return value: SUCCEED if no collisions found                                *
 *                                                                             *
 * Author: Alexander Vladishev                                                 *
 *                                                                             *
 * Comments: !!! Don't forget to sync the code with PHP !!!                    *
 *                                                                             *
 ******************************************************************************/
static int      validate_host(zbx_uint64_t hostid, zbx_vector_uint64_t *templateids,
                char *error, size_t max_error_len)
{
        const char      *__function_name = "validate_host";
        DB_RESULT       tresult;
        DB_RESULT       hresult;
        DB_ROW          trow;
        DB_ROW          hrow;
        char            *sql = NULL, *name_esc;
        size_t          sql_alloc = 256, sql_offset;
        ZBX_GRAPH_ITEMS *gitems = NULL, *chd_gitems = NULL;
        size_t          gitems_alloc = 0, gitems_num = 0,
                        chd_gitems_alloc = 0, chd_gitems_num = 0;
                                                                  570,2         12%
```

ZABBIX 2014 Conference

Present challenges

Different technology: back-end and front-end

Speed of development

Regressions and quality

Maintaining high quality: tests vs better tools

Historical PHP code

Performance

5 things I'd like to improve in Zabbix

... to start with

WEB interface: facts

- Navigation is not efficient: menu!
- Too many clicks for basic work-flow

  "Click until die", Lukas, Zabbix Conference 2014

- Disconnected information
- Monitoring/administration is strictly separated
- Drop downs: memory usage, performance, usability

ZABBIX 2014 Conference

WEB interface: UX

- Improve usability
- Navigation: be object-centric
  - Selected host → All information about the host is one-click away
- Information should be interconnected
- Make it faster (also related to API performance)

API: some facts

- Can be extremely slow
- Generates too much SQL queries
- No strict validation
- Weak error reporting

API: faster and more reliable

- Make it 10-100x faster
  - More efficient algorithms
  - Bulk operations
- Make it 1st class citizen: possibly move to Zabbix Server side
  - That's where we have the most efficient code
- Implement strict validation
- Error reporting
- Composability

ZABBIX 2014 Conference

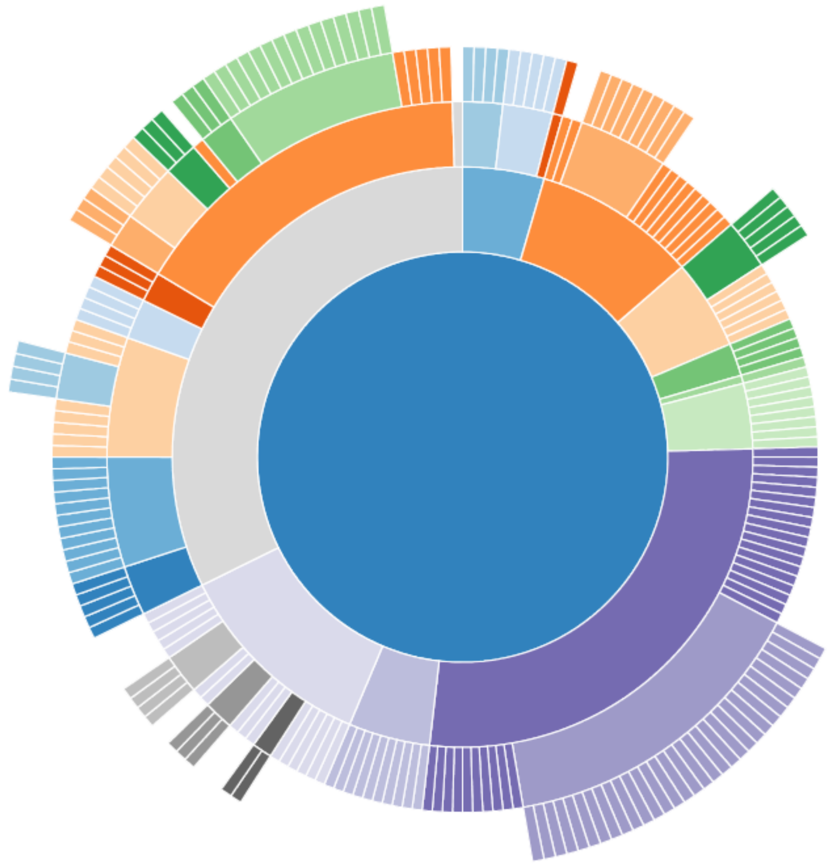
Reporting: facts

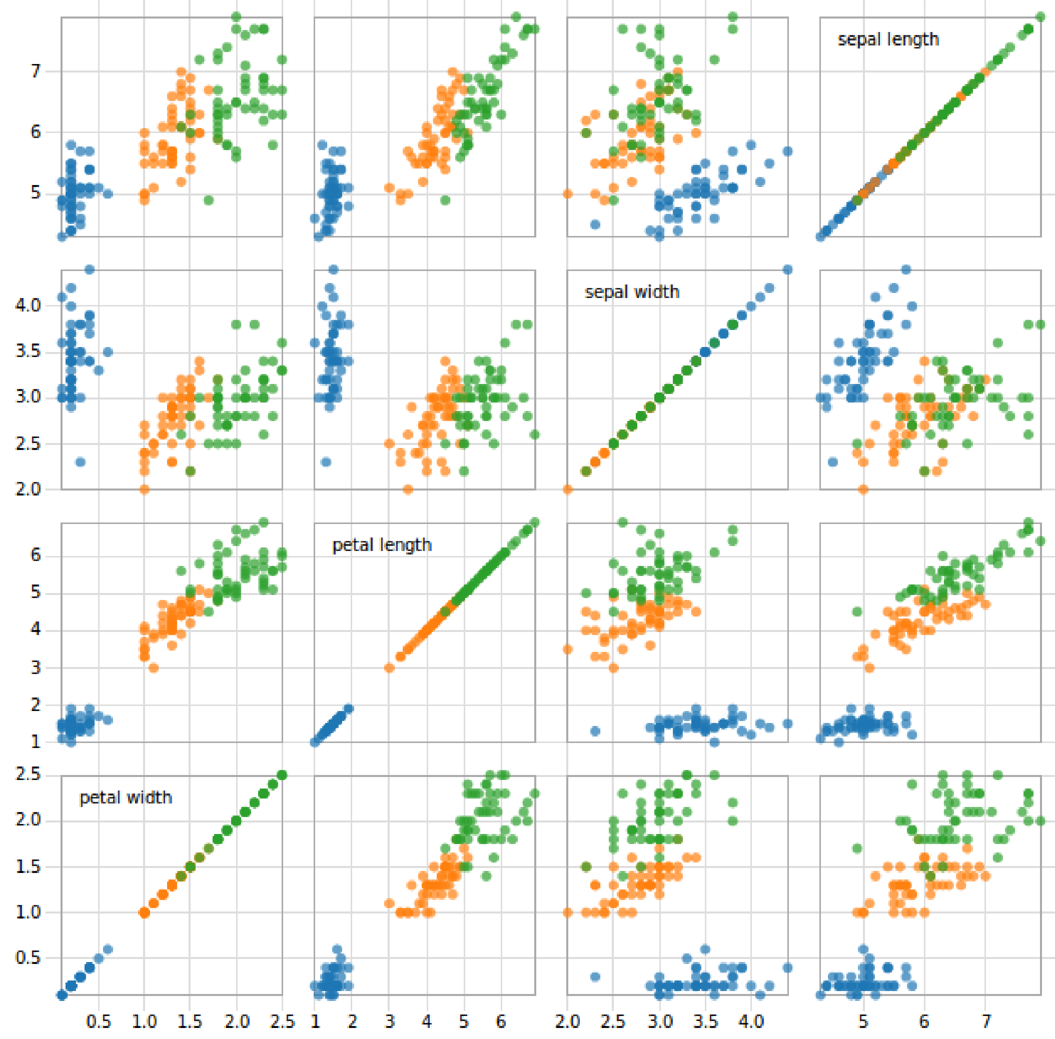
So much valuable data in the database but:

- Quite limited reporting capabilities
- No analytics
- No way to create ad-hoc reports
- No way to memorize parameters of executed reports

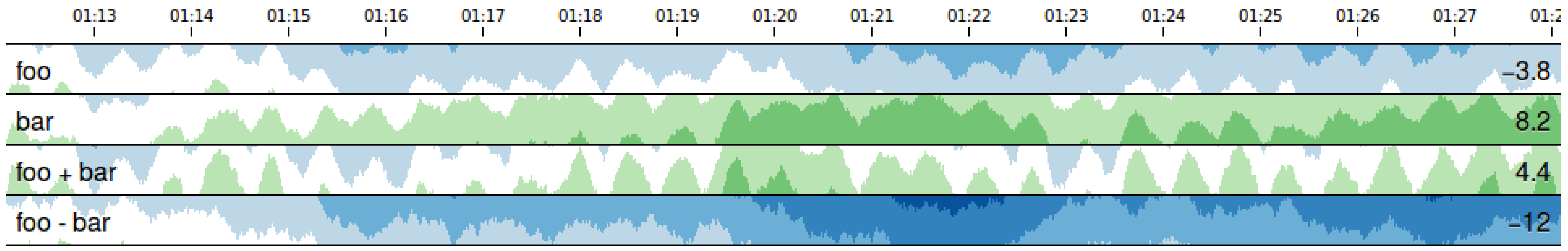
ZABBIX 2014 Conference

Reporting: visualization





- Real-time and analytics
- Important: response time and throughput



ZABBIX 2014 Conference

Scalability: facts

- Zabbix becomes (much) slower as data volume grows

- Requires special techniques to make it scalable

(database partitioning)

- Not easy to deliver HA/redundancy

Scalability: terabytes!

- Horizontal scalability of storage engine
  - Standard SQL engines do not deliver it
- Separate storage for historical data
- New distributed monitoring
- Front-end performance is important, sub-sec response

Encryption: facts

- Encryption and authentication are not supported out-of-the box

- Can be implemented using 3rd party tools (stunnel, OpenVPN, etc)

ZABBIX 2014 Conference

Encryption & authentication

- Must be part of the product
- Must be easy to enable and maintain
- Note quite sure about strong (SSL/TLS) encryption for Agents

Am I satisfied with existing tools and design decisions?

Am I satisfied with existing tools and design decisions?

Not quite.

What changes we may expect in Zabbix 3.0?

What changes we may expect in Zabbix 3.0?

I don't know.

ZABBIX 2014 Conference

Any questions?

:)