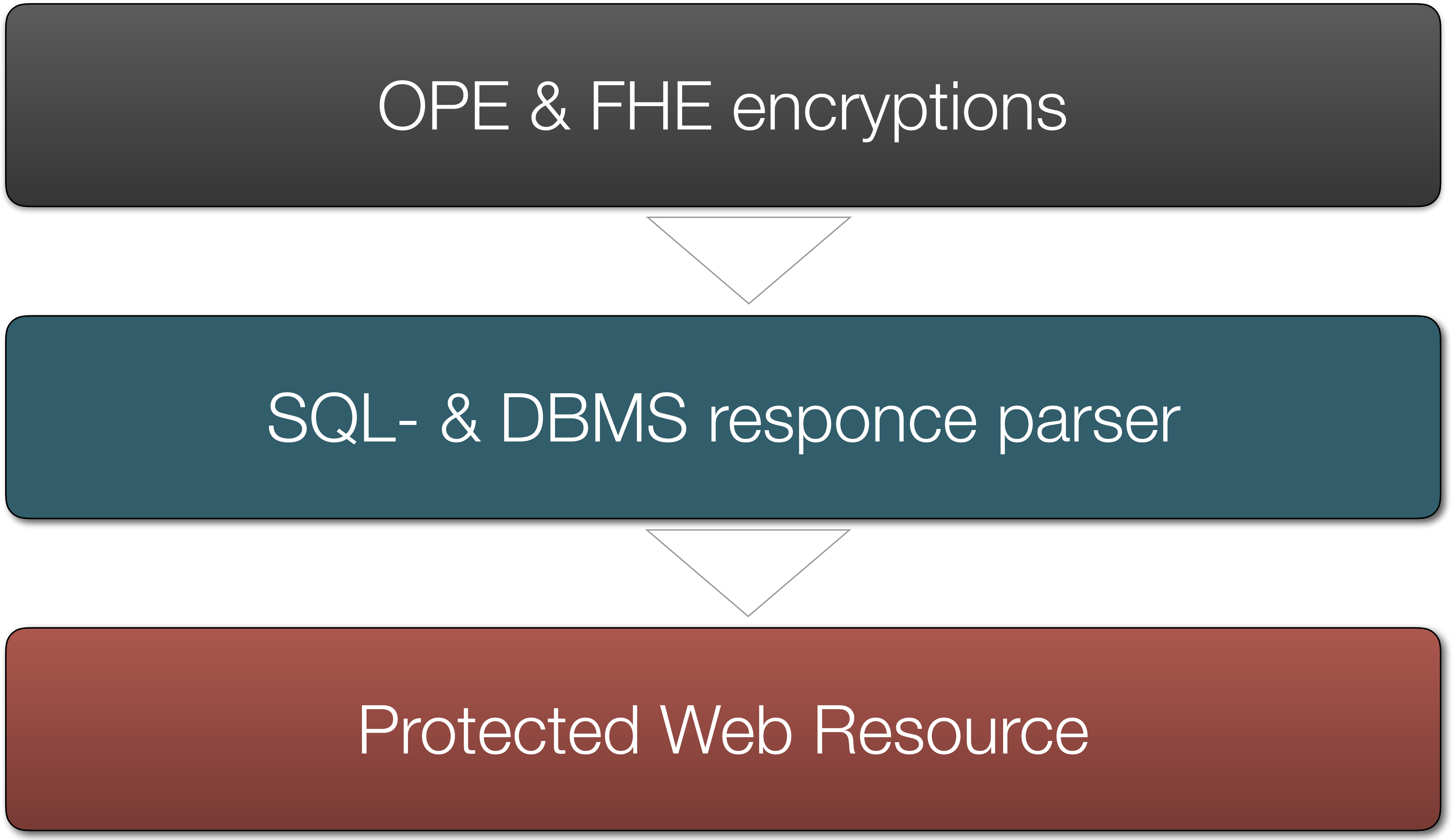
Securing Outsourced Database: Architecture for Protected Web Resource

KIRILL SHATILOV, SERGEY KRENDELEV, DIANA ANISUTINA, ARTEM SUMANEEV AND EVGENY OGURTSOV

2015
CEE-SEC(R)
Software Engineering
Conference in Russia

Current research is performed in Laboratory NSU-Parallels in Novosibirsk State University under financial support of the Ministry of education and science of Russia (contract № 02.G25.310054).
Science leader - Krendelev Sergey Fedorovich

IN GENERAL



OPE & FHE encryptions

SQL- & DBMS responce parser

Protected Web Resource

OUTLINE

- Motivation
- Methodology and Design

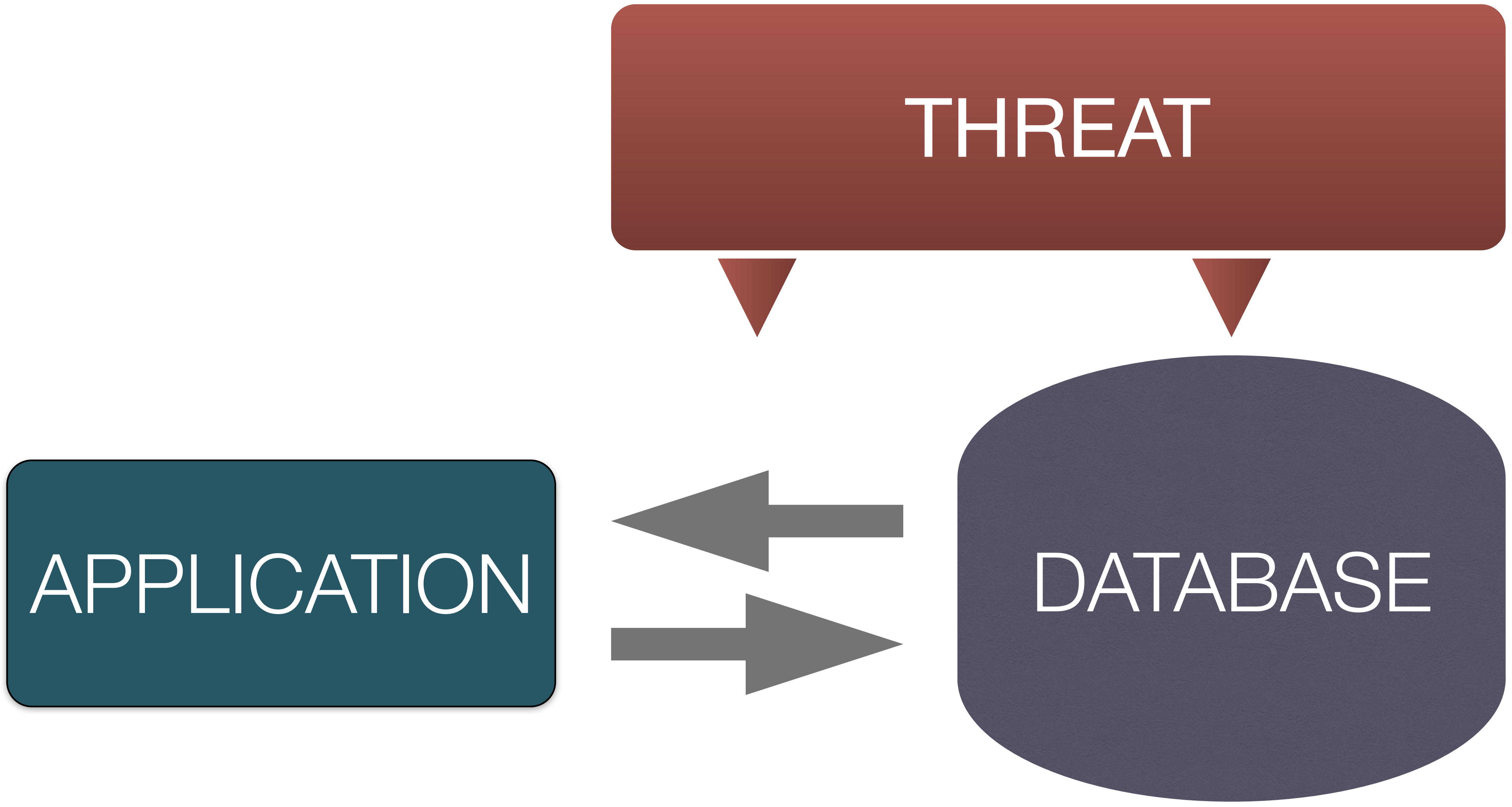
  - encryptions

  - syntax processor

  - components configuration
- Achieved results
- Future challenges
- Summary

MOTIVATION

SCENARIO	·	SQL DBMS backed Web resource
THREAT 1	·	Insider
THREAT 2	·	Adversary

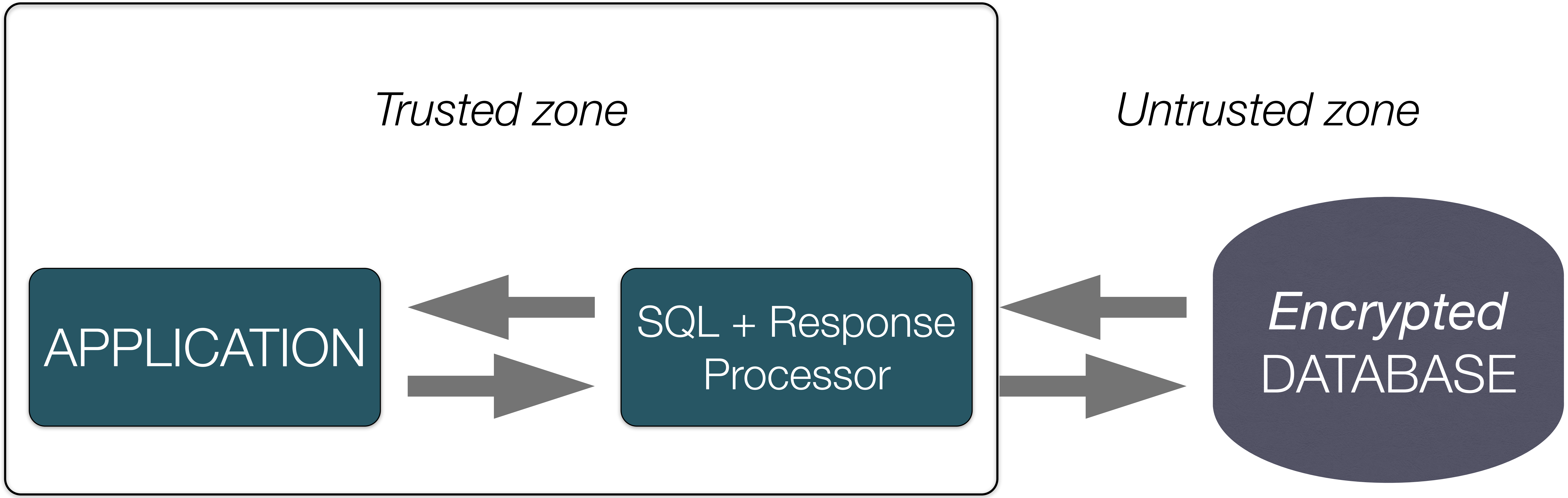


SOLUTION

POINT 1 · Encrypted database

POINT 2 · Intermediate processing components

POINT 3 · Trust zones



POINT 1. ENCRYPTION LIBRARY

Deterministic and Probabilistic encryptions

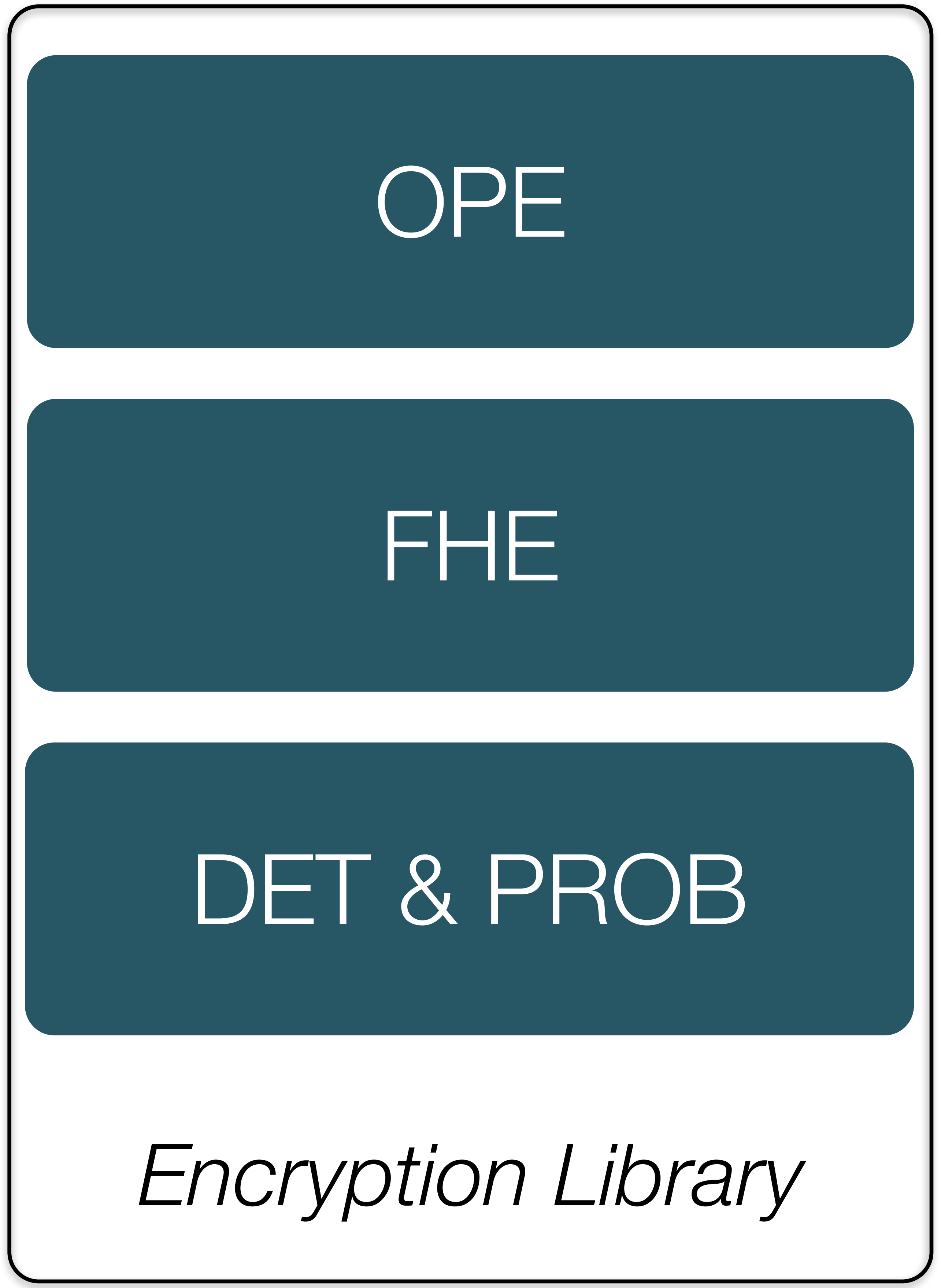
- Strong security
- Text Data

Order Preserving Encryption

- Order operations over ciphertexts
- Secure indexes, dates

Fully Homomorphic Encryption

- Multiplication & addition over ciphertexts
- Math & commerce



LINKS

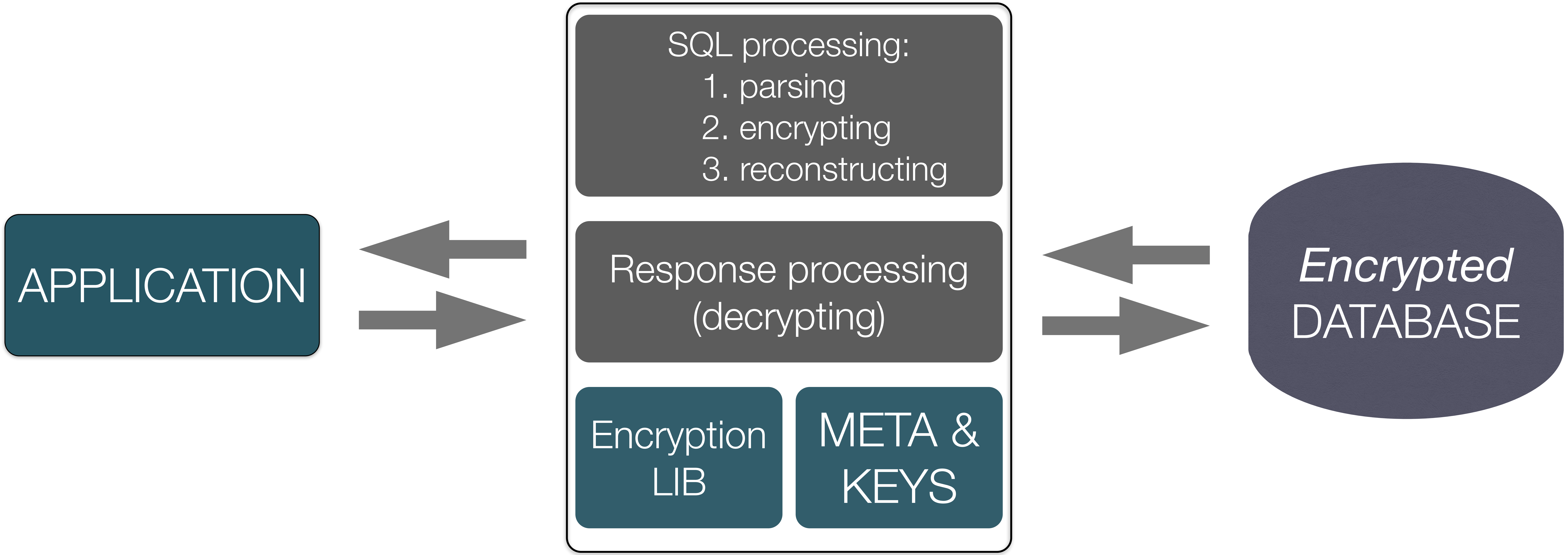
- FHE

Fully Homomorphic Encryption for Secure Computations in Protected Database
Darya Chechulina, Kirill Shatilov, Sergey Krendelev,
Position Papers of the 2015 Federated Conference on Computer Science and Information Systems, pp. 125-131

- OPE

Order-preserving encryption schemes based on arithmetic coding and matrices
Maria Usoltseva, Sergey Krendelev, Mikhail Yakovlev,
Proceedings of the 2014 Federated Conference on Computer Science and Information Systems, pp 891-899

POINT 2. PROCESSING COMPONENTS



POINT 2. SYNTAX PROCESSING

Create statement processing:

1. Encryption’s keys are generated or chosen.

2. Determination of number, names, types and constraints of output columns.

3. Correct SQL string is created according to determined information.

4. Anonimisation of columns' and tables' names.

5. Modified statement is sent to DBMS.

DML statements processing:

1. Data's extraction

2. Data's encryption

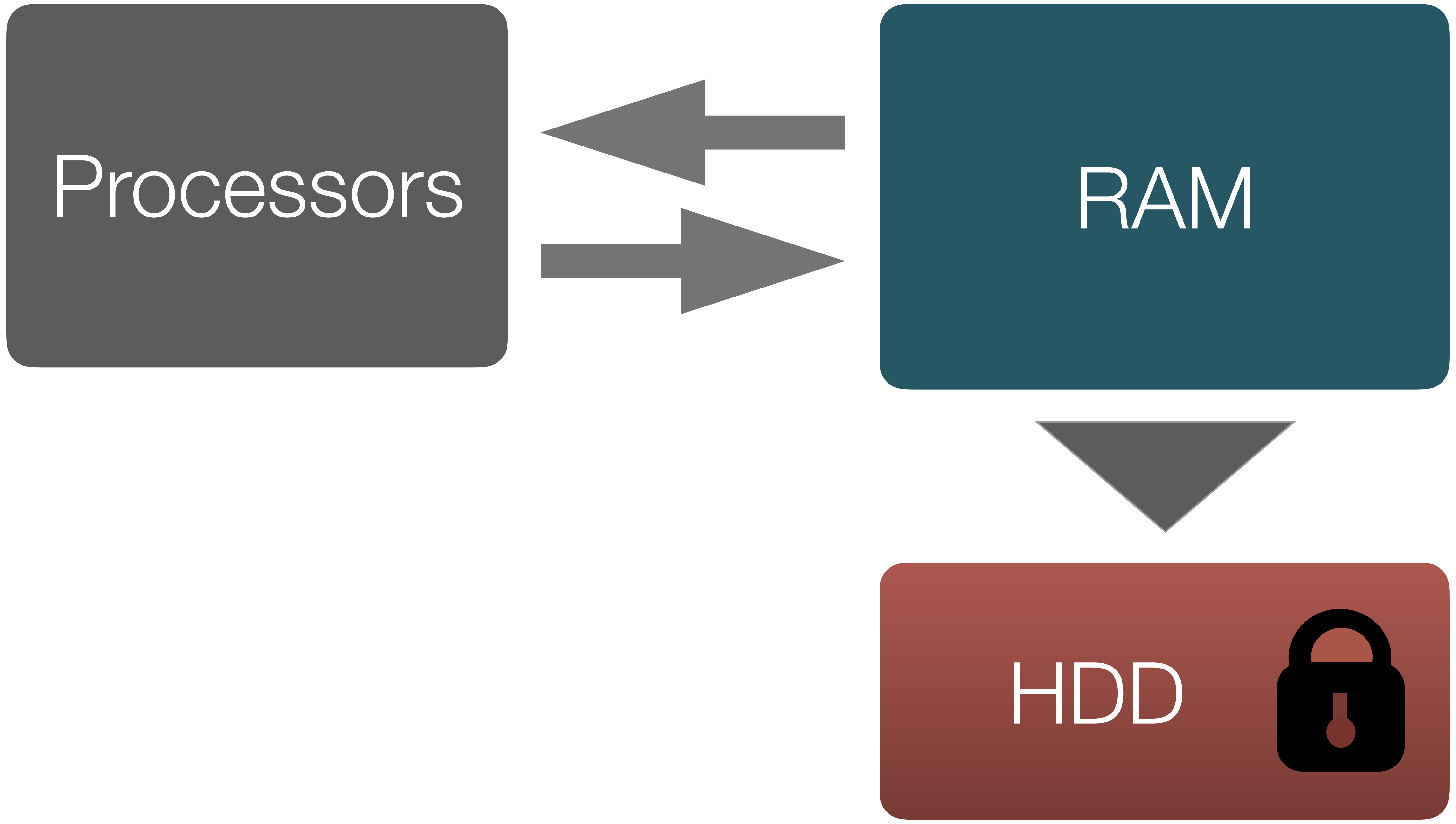
3. Columns' names synchronization

4. Math correction (in some cases)

5. Decryption of response (if needed)

METAFILE STORAGE

- In-memory database
- Constant backups
- Encrypted on HDD
- Storing:
  - Encryption keys
  - Initial column info
  - Output column format
  - JOIN groups info



POINT 2. PROCESSING CHALLENGES

- Multiple output columns

Encryption(value) = (a, b, c, ...)

SELECT *value* ***FROM*** *table_name* ▶ ***SELECT*** *a, b, c, ..* ***FROM*** *table_name*

POINT 2. PROCESSING CHALLENGES

- Encryption specific math

FHEncryption(value) = (a)(b)

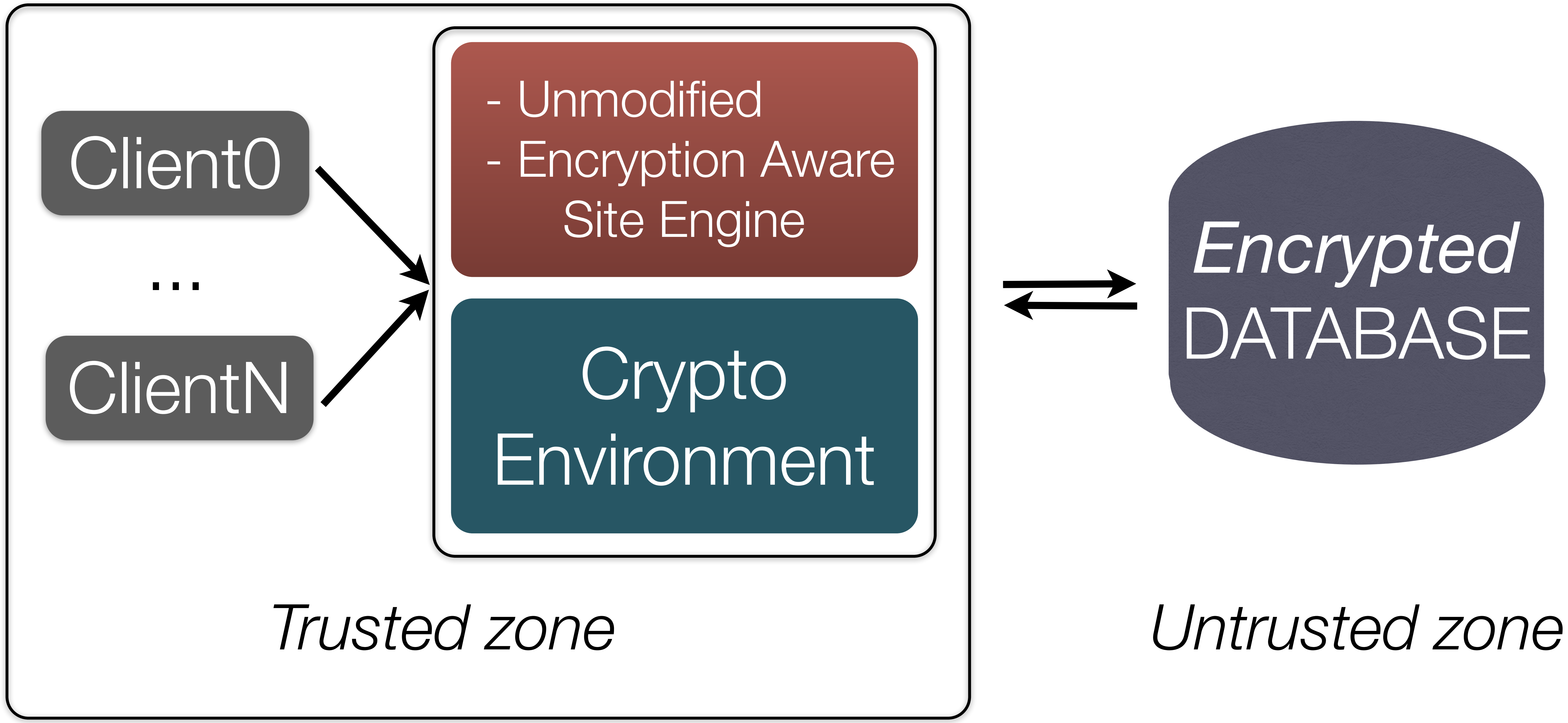
ciphertext + ciphertext = **UDF** (a1, b1, a2, b2, **Multiplication Table**)

Multiplication Table ~ 5000 values

*SELECT **SUM**(values) FROM table_name* ▶ *SELECT **UDF_SUM(...)** FROM table_name*

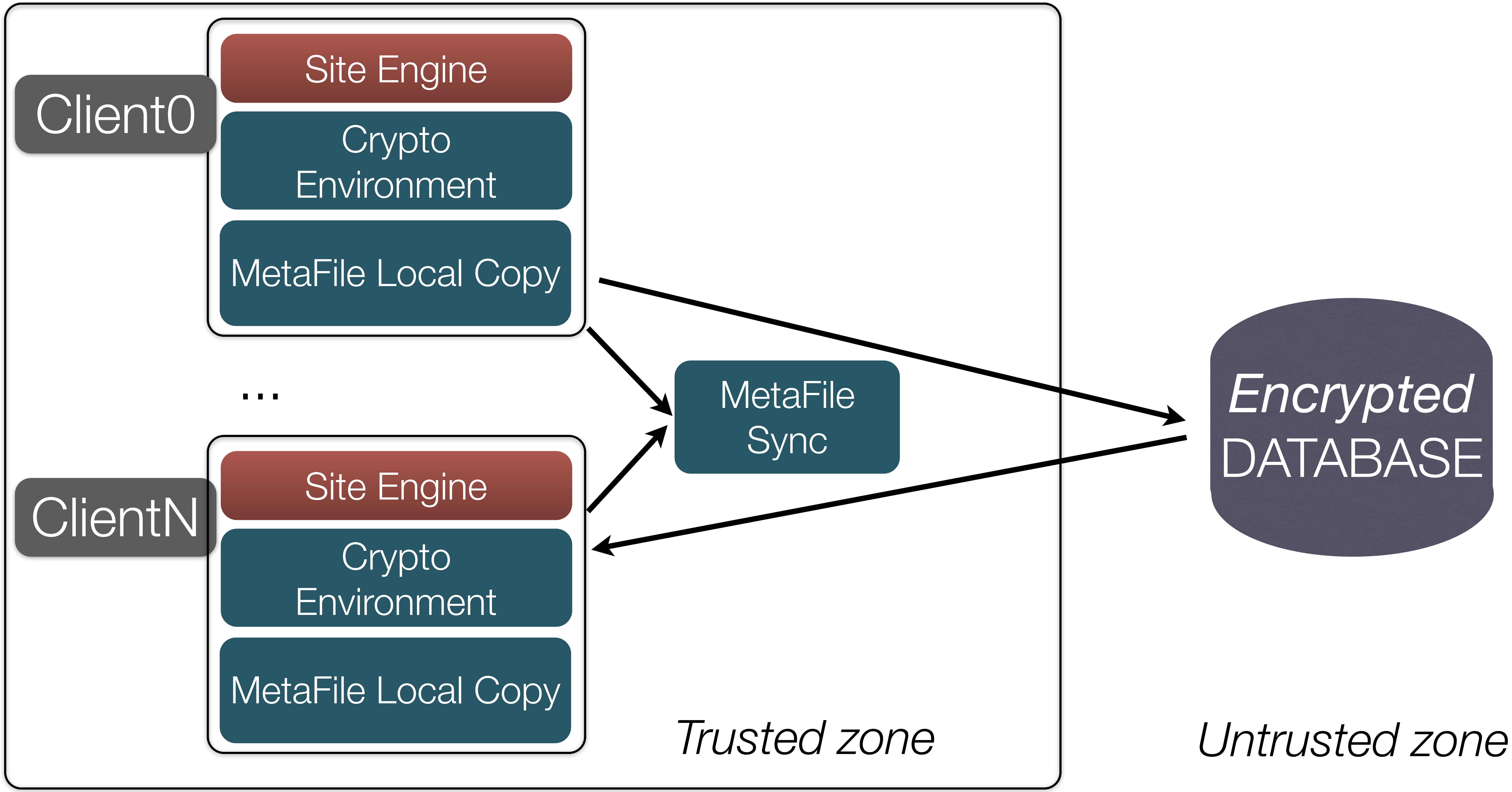
POINT 3. ZONING & CONFIGURATION

1. Centralized

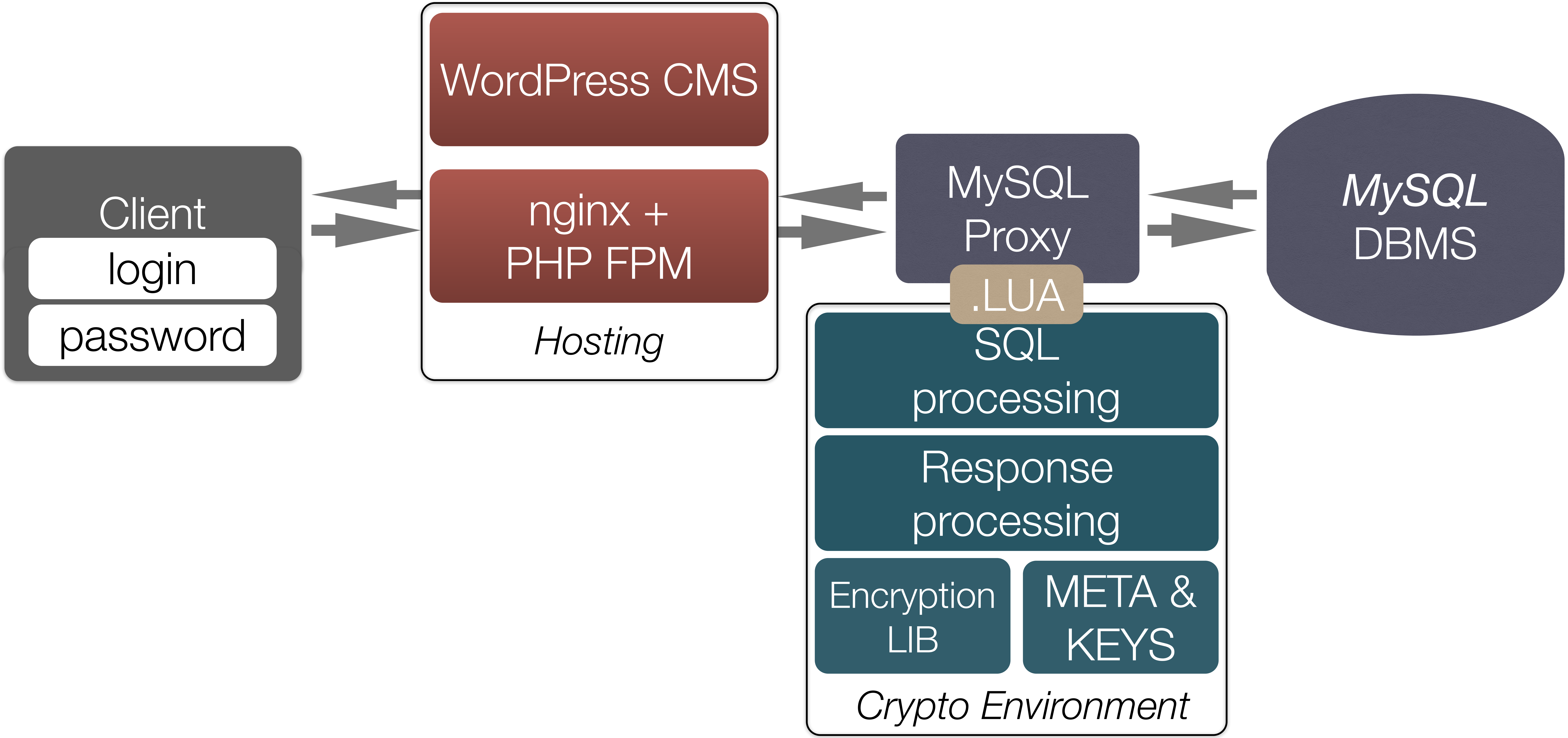


POINT 3. ZONING & CONFIGURATION

2. Distributed



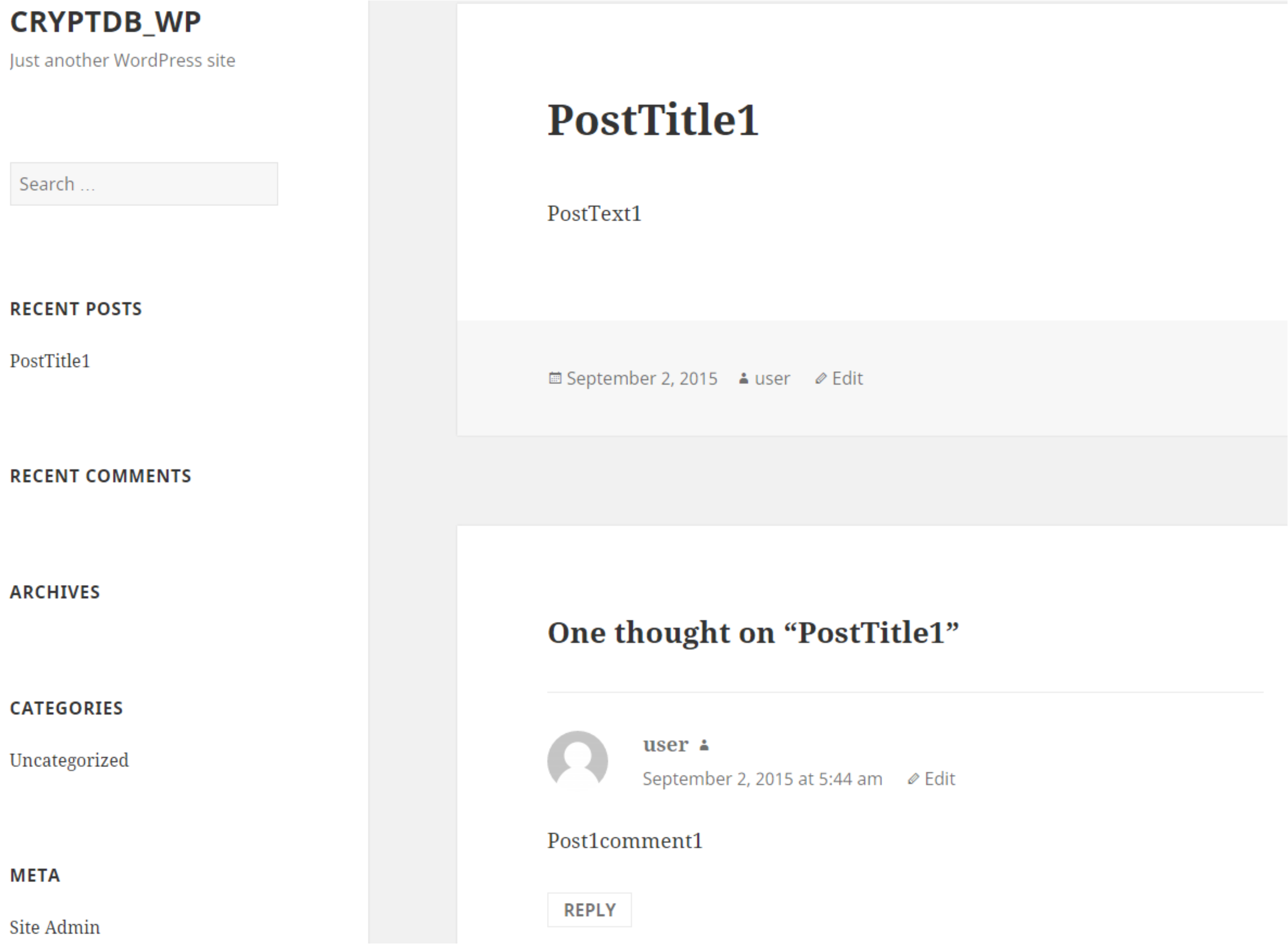
RESULTS. PRACTICAL IMPLEMENTATION



RESULTS. APPLIED ENCRYPTION

Field	Type	Encryption
tags, headers	text	deterministic
post, comments text	long text	probabilistic
post, comments, events date	date	OPE
user email, name	text	deterministic
user password	text	deterministic
ratings, order terms	integer	OPE

RESULTS. SHOWCASE



CRYPTDB_WP

Just another WordPress site

Search …

RECENT POSTS

PostTitle1

RECENT COMMENTS

ARCHIVES

CATEGORIES

Uncategorized

META

Site Admin

PostTitle1

PostText1

September 2, 2015 user Edit

One thought on “PostTitle1”

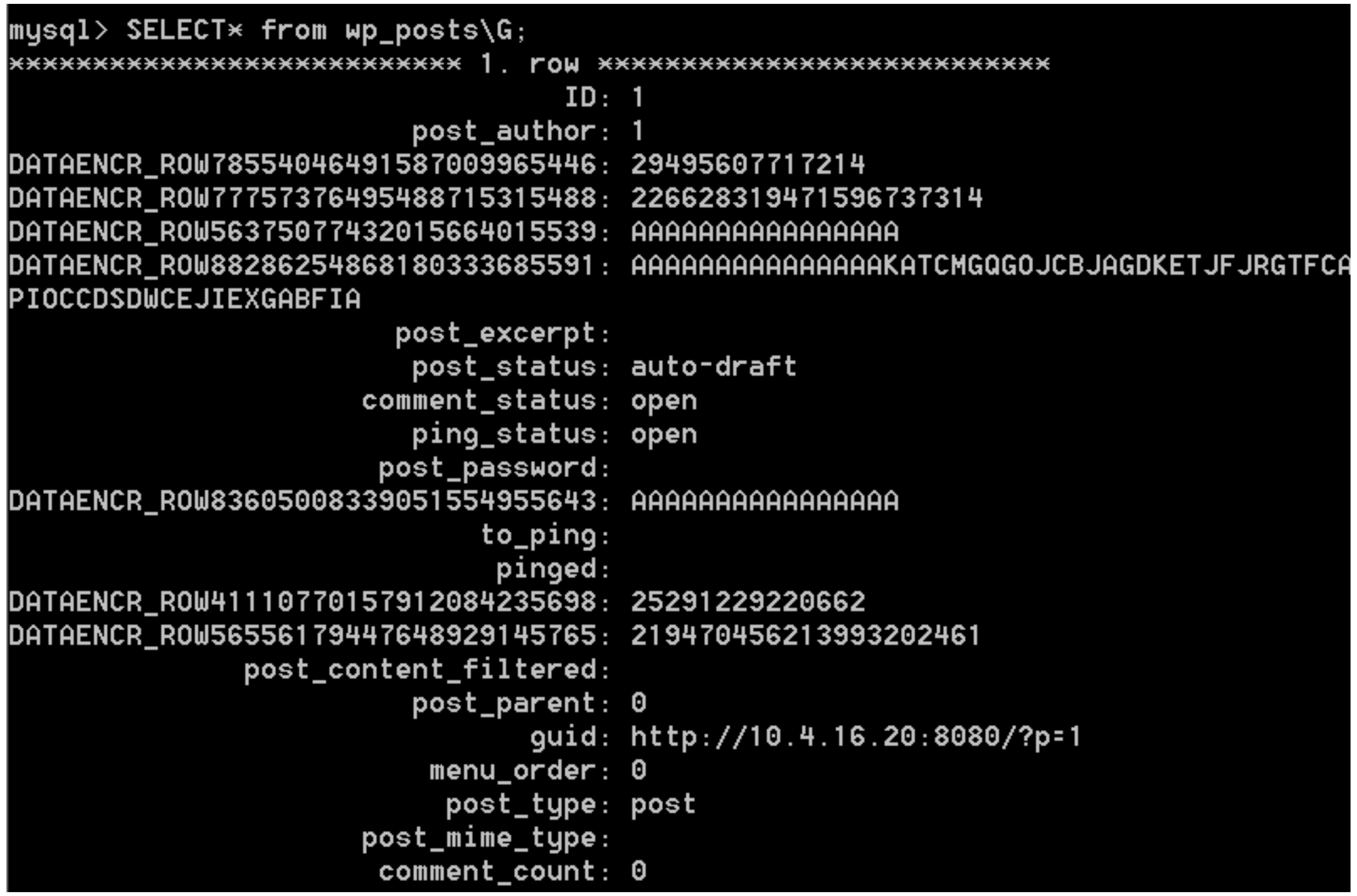
user

September 2, 2015 at 5:44 am Edit

Post1comment1

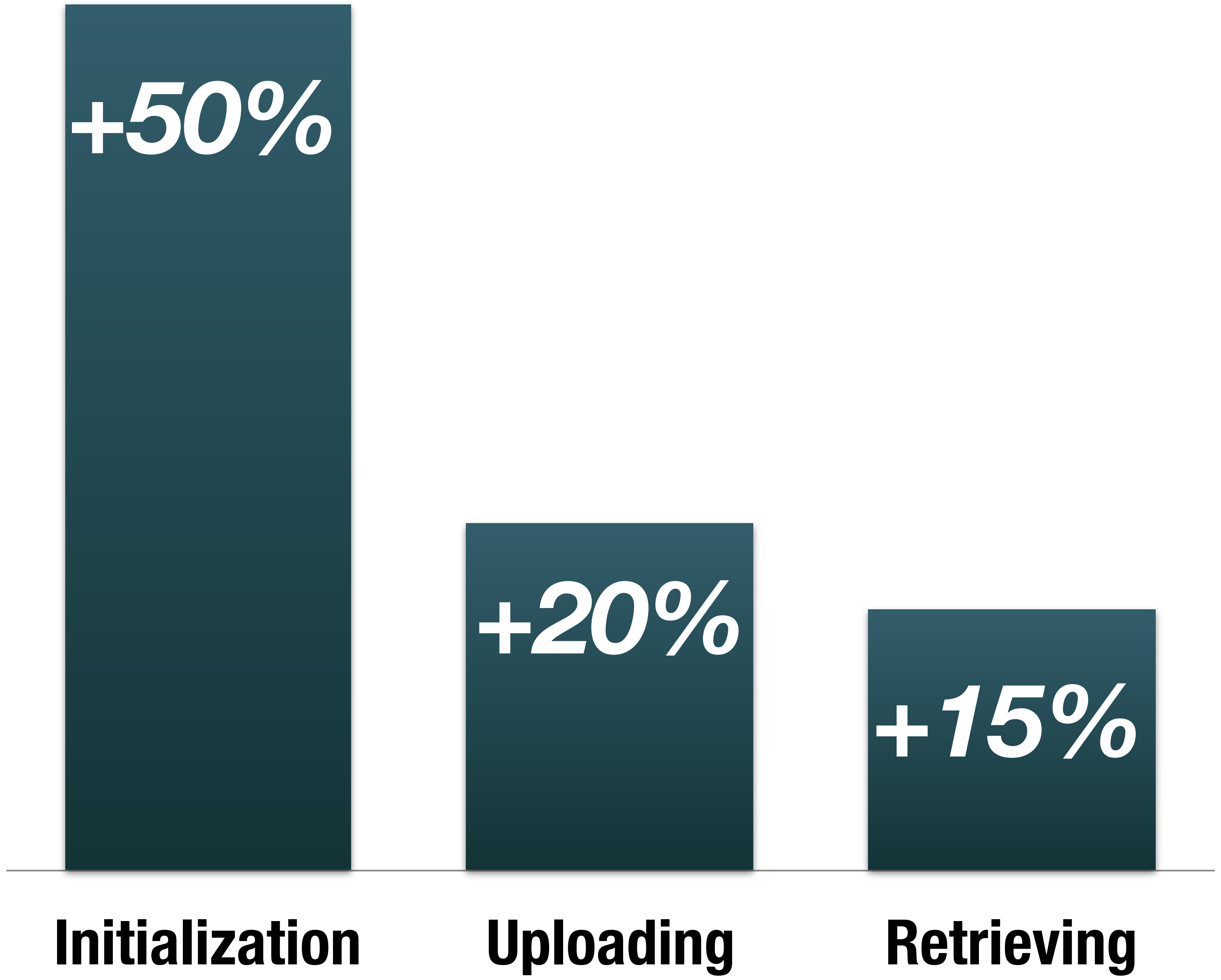
REPLY

*mysql>***SELECT** * **FROM** *wp_posts***\G**

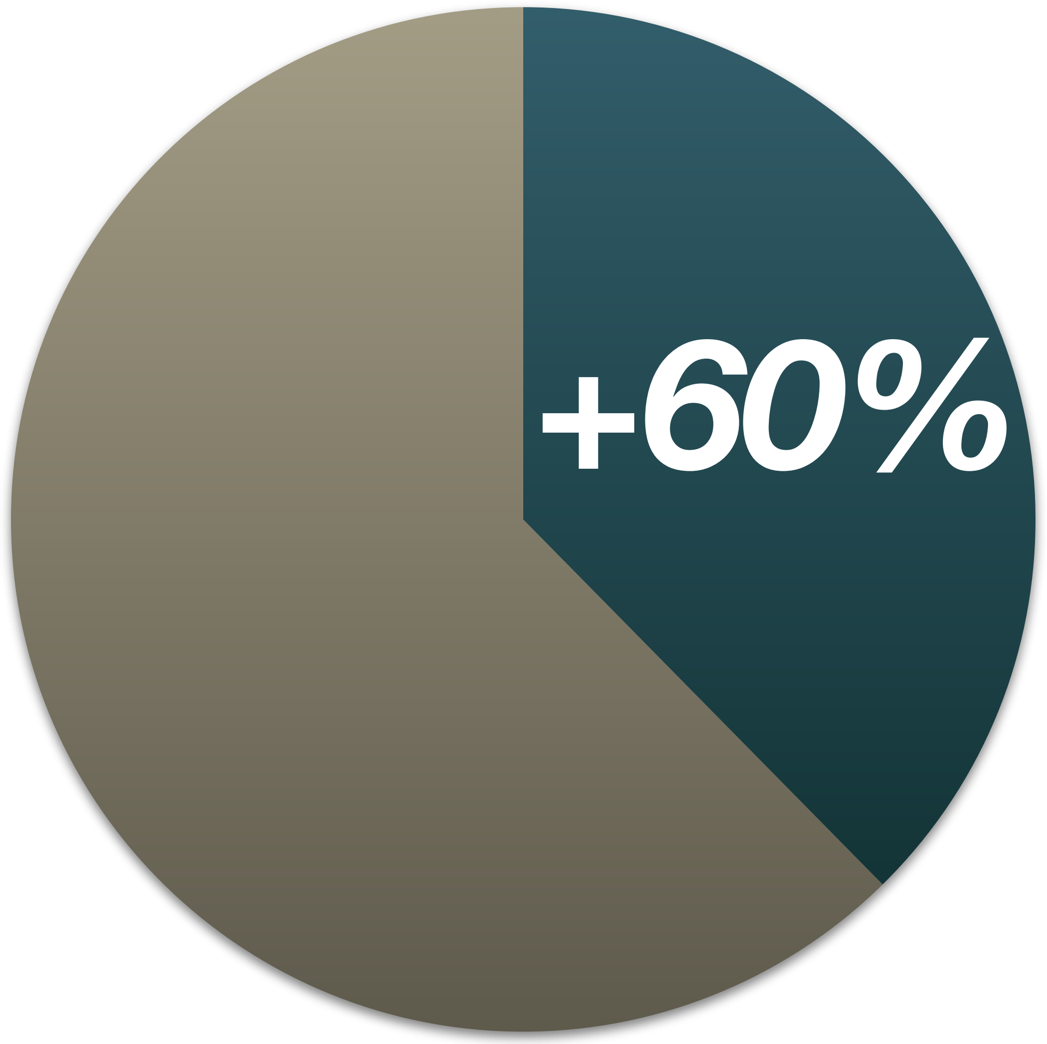


```
mysql> SELECT* from wp_posts\G;
*************************** 1. row ***************************
                                 ID: 1
                        post_author: 1
DATAENCR_ROW78554046491587009965446: 29495607717214
DATAENCR_ROW77757376495488715315488: 2266283194715967373 14
DATAENCR_ROW56375077432015664015539: AAAAAAAAAAAAAAAA
DATAENCR_ROW88286254868180333685591: AAAAAAAAAAAAAAAKATCMGQGOJCBJAGDKETJFJRGTFCA
PIOCCDSDWCEJIEXGABFIA
                       post_excerpt:
                        post_status: auto-draft
                     comment_status: open
                        ping_status: open
                      post_password:
DATAENCR_ROW83605008339051554955643: AAAAAAAAAAAAAAAA
                            to_ping:
                             pinged:
DATAENCR_ROW41110770157912084235698: 25291229220662
DATAENCR_ROW56556179447648929145765: 219470456213993202461
              post_content_filtered:
                        post_parent: 0
                               guid: http://10.4.16.20:8080/?p=1
                         menu_order: 0
                          post_type: post
                     post_mime_type:
                      comment_count: 0
```

RESULTS. EVALUATION



Average Performance Overhead



Database size increase

FUTURE CHALLENGES

- Cross-platform build
- PostgreSQL
- Multithreading environment
- Memory optimization

SUMMARY

WHAT?	• Real-time application's data protection
WHY?	• Outsourced data's privacy
HOW?	• OPE & FHE
AND..?	• Real life applications and development goals

THANK YOU