# Moving MongoDB Workloads to Postgres with FerretDB

Alexey Palazhchenko
FerretDB Inc.

Belgrade, Serbia 2023
PERCONA UNIVERSITY

FerretDB PERCONA Singidunum University

# MongoDB: The Good Parts

# MongoDB: The Good Parts

- Great developer experience

# MongoDB: The Good Parts

- Great developer experience
- Good drivers

# MongoDB: The Good Parts

- Great developer experience
- Good drivers
- Great community (back in the day)

# MongoDB: The Bad Parts

# MongoDB: The Bad Parts

- Reliability, performance, scalability

# MongoDB: The Bad Parts

- Reliability, performance, scalability

# MongoDB: The Bad Parts

- Reliability, performance, scalability
- Does too much?

# MongoDB: The Bad Parts

- Reliability, performance, scalability
- Does too much?
- Company loses money

# MongoDB: The Bad Parts

- Reliability, performance, scalability
- Does too much?
- Company loses money
- All-in on Atlas

# MongoDB: The Bad Parts

- Reliability, performance, scalability
- Does too much?


- Company loses money
- All-in on Atlas
- License change

# Not a legal advice

# AGPL v3

# AGPL v3

- «If you modify the Program»

# AGPL v3

- «If you modify the Program»
- «All users interacting with it remotely through a computer network»

# AGPL v3

- «If you modify the Program»
- «All users interacting with it remotely through a computer network»
- «Corresponding Source»

# AGPL v3

- «If you modify the Program»
- «All users interacting with it remotely through a computer network»
- «Corresponding Source»
- «Including scripts to control those activities»

# SSPL

# SSPL

- «If you make the functionality of the Program»

# SSPL

- «If you make the functionality of the Program»
- «Available to third parties as a service»

# SSPL

- «If you make the functionality of the Program»
- «Available to third parties as a service»
- «Includes, without limitation, […]»

# SSPL

- «If you make the functionality of the Program»
- «Available to third parties as a service»
- «Includes, without limitation, […]»
- «Service Source Code»

# SSPL

- «If you make the functionality of the Program»
- «Available to third parties as a service»
- «Includes, without limitation, […]»
- «Service Source Code»
- «Including, without limitation, management software, user interfaces, application program interfaces, […], all such that a user could run an instance of the service»

# SSPL FAQ

# SSPL FAQ


https://opensource.org/sspl-not-open-source

# Protocol specification license

```
15  + .. note::
16  +
17  +    This MongoDB Wire Protocol Specification is licensed under a
18  +    `Creative Commons Attribution-NonCommercial-ShareAlike 3.0 United States License
19  +    <https://creativecommons.org/licenses/by-nc-sa/3.0/us/>`__. You may
20  +    not use or adapt this material for any commercial purpose, such as
21  +    to create a commercial database or database-as-a-service offering.
22  +
```

# Alternatives?

# Alternatives?

**Amazon**
DocumentDB

# Alternatives?

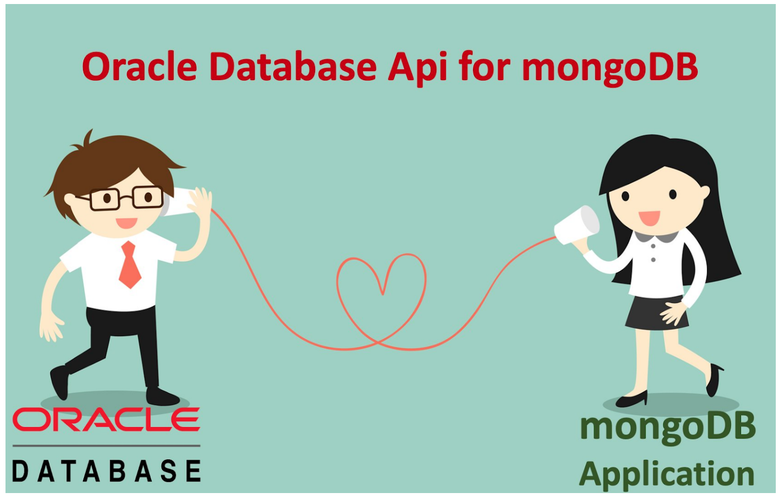
**Amazon**
DocumentDB

Azure
Cosmos DB

# Alternatives?

**Amazon**
DocumentDB

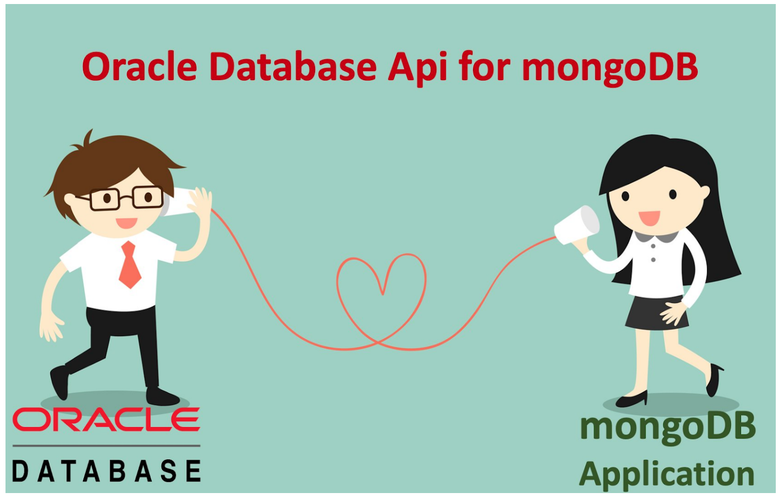
Azure
Cosmos DB



# Alternatives?

**Amazon**
DocumentDB

Azure
Cosmos DB

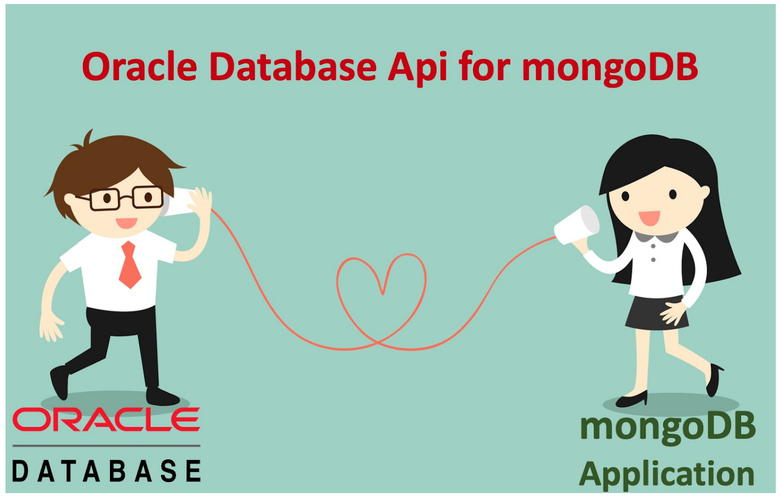



IBM
Informix database
software

# Alternatives?

**Amazon**
DocumentDB

MariaDB®
MaxScale




Azure
Cosmos DB

IBM
Informix database
software

# Our idea

# Our idea

- Should work with unmodified MongoDB drivers and apps

# Our idea

- Should work with unmodified MongoDB drivers and apps
- Should not store data itself

# Our idea

- Should work with unmodified MongoDB drivers and apps
- Should not store data itself
- Should be a honest open-source solution

# Who are you?

# Who are you?

Peter Zaitsev, Advisor
Percona co-founder

# Who are you?

Peter Zaitsev, Advisor
Percona co-founder

Peter Farkas, CEO
ex-Percona, Cloudera
10+ years of experience
leading database software
and services teams

# Who are you?

Peter Zaitsev, Advisor
Percona co-founder

Peter Farkas, CEO
ex-Percona, Cloudera
10+ years of experience leading database software and services teams

Alexey Palazhchenko, CTO
ex-Percona
10+ years of experience leading software engineering teams

# FerretDB

# FerretDB

- Stateless proxy for MongoDB wire protocol

# FerretDB

- Stateless proxy for MongoDB wire protocol
- Stores data in PostgreSQL (or SQLite)

# FerretDB

- Stateless proxy for MongoDB wire protocol
- Stores data in PostgreSQL (or SQLite)
- Apache License 2.0

# FerretDB

# FerretDB

- Launched 1¾ years ago

# FerretDB

- Launched 1¾ years ago
- v1.5.0 released 3 days ago

# FerretDB

- Launched 1¾ years ago
- v1.5.0 released 3 days ago
- Production-ready*

# Why PostgreSQL?

# Why PostgreSQL?

- Open-source, huge community

# Why PostgreSQL?

- Open-source, huge community
- Very reliable

# Why PostgreSQL?

- Open-source, huge community
- Very reliable
- Good JSON support

# Why PostgreSQL?

- Open-source, huge community
- Very reliable
- Good JSON support
- Many companies run both PostgreSQL and MongoDB

# Storing BSON in jsonb

# Storing BSON in jsonb

- Field order is significant in BSON documents

# Storing BSON in jsonb

- Field order is significant in BSON documents
- Field order is not preserved by jsonb objects

# Storing BSON in jsonb

- Field order is significant in BSON documents
- Field order is not preserved by jsonb objects

```
{
    "$s": {
        "$k": [
            "platform",
            "application"
        ]
    },
    "platform": "Node.js v14.17.3",
    "application": "mongosh 1.0.1"
}
```

# Storing BSON in jsonb

# Storing BSON in jsonb

- BSON has more data types than JSON

# Storing BSON in jsonb

- BSON has more data types than JSON
  - int32, int64, double

# Storing BSON in jsonb

- BSON has more data types than JSON
  - int32, int64, double
  - timestamp

# Storing BSON in jsonb

- BSON has more data types than JSON
  - int32, int64, double
  - timestamp
  - JavaScript code *with scope*

# Storing BSON in jsonb

- BSON has more data types than JSON
  - int32, int64, double
  - timestamp
  - JavaScript code *with scope*

```
{
    "$s": {
        "p": {
            "v": {
                "t": "int"
            },
        },
    },
    "v": 42.0
}
```

# Comparing BSON values

# Comparing BSON values

- We need to compare BSON values for filtering and sorting

# Comparing BSON values

- We need to compare BSON values for filtering and sorting
- Including values of different types

# Comparing BSON values

- We need to compare BSON values for filtering and sorting
- Including values of different types
- And problematic values like nulls

# Comparing BSON values

- We need to compare BSON values for filtering and sorting
- Including values of different types
- And problematic values like nulls

```
WHERE v < 42
ORDER BY v
```

# Comparing BSON values

- We need to compare BSON values for filtering and sorting
- Including values of different types
- And problematic values like nulls

```
WHERE v < 42
ORDER BY v
```

```
int32, int64, double
```

# Comparing BSON values

- We need to compare BSON values for filtering and sorting
- Including values of different types
- And problematic values like nulls

```
WHERE v < 42
ORDER BY v
```

```
int32, int64, double
string
```

# Comparing BSON values

- We need to compare BSON values for filtering and sorting
- Including values of different types
- And problematic values like nulls

```
WHERE v < 42
ORDER BY v
```

```
int32, int64, double
string
object, array
```

# Comparing BSON values

# Comparing BSON values

- NaN < 1

# Comparing BSON values

- NaN < 1
- null < NaN

# Comparing BSON values

- NaN < 1
- null < NaN
- [ ] < null

# Comparing BSON values

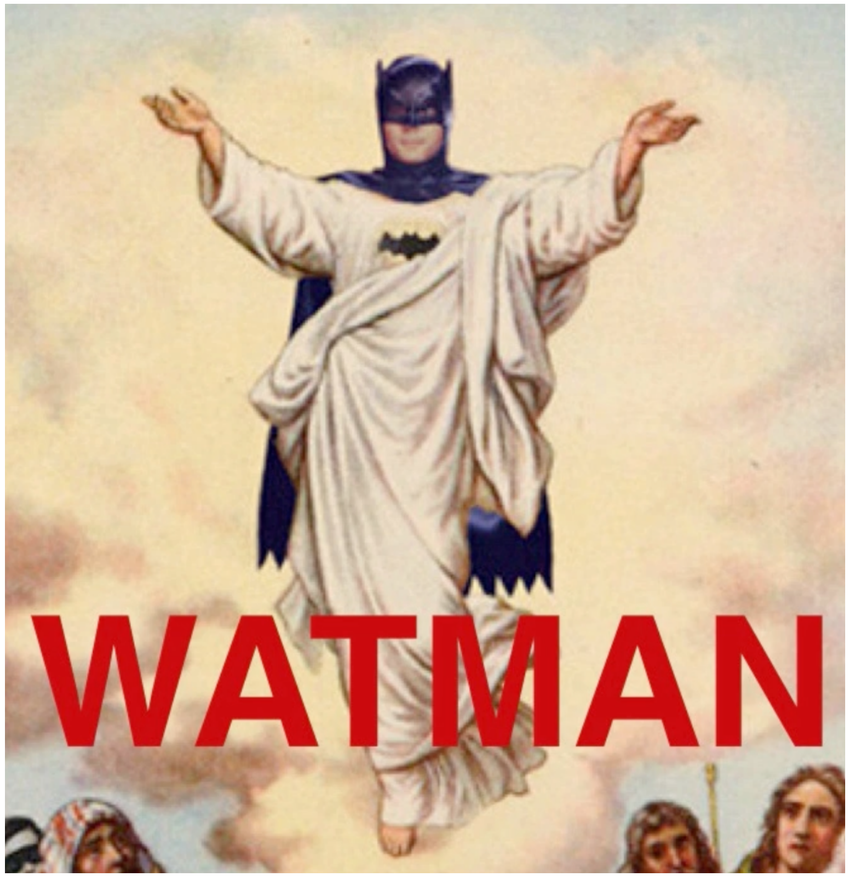
- NaN < 1
- null < NaN
- [ ] < null
- [ ] < [null]

# Comparing BSON values

- NaN < 1
- null < NaN
- [ ] < null
- [ ] < [null]
- null </> [null]
  (depends on sorting order)

# Comparing BSON values

- NaN < 1
- null < NaN
- [ ] < null
- [ ] < [null]
- null </> [null]
  (depends on sorting order)




https://www.destroyallsoftware.com/talks/wat

# Comparing BSON values

# Comparing BSON values

- Do a comparison in Go

# Comparing BSON values

- Do a comparison in Go
- Fetch all data

# Comparing BSON values

- Do a comparison in Go
- Fetch all data
- Add great integration tests

# Comparing BSON values

- Do a comparison in Go
- Fetch all data
- Add great integration tests
- Remove support for edge-cases

# Comparing BSON values

- Do a comparison in Go
- Fetch all data
- Add great integration tests
- Remove support for edge-cases
- Pushdown simple queries

# Comparing BSON values

- Do a comparison in Go
- Fetch all data
- Add great integration tests
- Remove support for edge-cases
- Pushdown simple queries
  - WHERE _jsonb->’v’ = 42

# Comparing BSON values

- Do a comparison in Go
- Fetch all data
- Add great integration tests
- Remove support for edge-cases
- Pushdown simple queries
  - WHERE _jsonb->’v’ = 42
- Make indexes work

# Comparing BSON values

- Do a comparison in Go
- Fetch all data
- Add great integration tests
- Remove support for edge-cases
- Pushdown simple queries
  - WHERE _jsonb->’v’ = 42
- Make indexes work

- Generate complex queries?

# Comparing BSON values

- Do a comparison in Go
- Fetch all data
- Add great integration tests
- Remove support for edge-cases
- Pushdown simple queries
  - WHERE _jsonb->’v’ = 42
- Make indexes work

- Generate complex queries?
- Use stored procedures?

# Comparing BSON values

- Do a comparison in Go
- Fetch all data
- Add great integration tests
- Remove support for edge-cases
- Pushdown simple queries
  - WHERE _jsonb->’v’ = 42
- Make indexes work

- Generate complex queries?
- Use stored procedures?
- Use custom operators?

# Comparing BSON values

- Do a comparison in Go
- Fetch all data
- Add great integration tests
- Remove support for edge-cases
- Pushdown simple queries
  - WHERE _jsonb->’v’ = 42
- Make indexes work

- Generate complex queries?
- Use stored procedures?
- Use custom operators?
- Use custom data type?

# Comparing BSON values

- Do a comparison in Go
- Fetch all data
- Add great integration tests
- Remove support for edge-cases
- Pushdown simple queries
  - WHERE _jsonb->’v’ = 42
- Make indexes work

- Generate complex queries?
- Use stored procedures?
- Use custom operators?
- Use custom data type?
- Use custom extension?

# Integration tests

# Integration tests

- Our own compatibility tests

# Integration tests

- Our own compatibility tests
- Tests of existing applications

# Integration tests

- Our own compatibility tests
- Tests of existing applications
- «MongoDB API Tester» tests

# Integration tests

- Our own compatibility tests
- Tests of existing applications
- «MongoDB API Tester» tests
- Community testers

# Community

# Community

- Roadmap: https://github.com/orgs/FerretDB/projects/2

# Community

- Roadmap: https://github.com/orgs/FerretDB/projects/2
- Please star us now: https://github.com/FerretDB/FerretDB

# Community

- Roadmap: https://github.com/orgs/FerretDB/projects/2
- Please star us now: https://github.com/FerretDB/FerretDB
- And give it a try tomorrow

# Questions?

- Thank you!
- Questions?

- https://ferretdb.io/
- https://github.com/FerretDB
- https://twitter.com/ferret_db
- https://techhub.social/@ferretdb