CHANNEL 2

JOBINTECH
CONFERENCE 2024

2-3 OTTOBRE
ONLINE

LIVE TECH WORKSHOP
GIOVEDI' 3 OTTOBRE ORE 18.00

FerretDB: un'alternativa a MongoDB davvero open-source!

con Marco Breveglieri
Software Developer e Trainer

jobintech.com

Rete Informatica Lavoro
Il portale del lavoro dedicato agli informatici

FerretDB

Una alternativa a MongoDB davvero open-source

Workshop

jobintech.com

CompilaQuindiVa

Let's code together

LIVE Coding twitch @compilaquindiva

Marco Breveglieri

Software Developer | Trainer and Consultant | Tech Content Creator

@ABLS Team - Reggio Emilia (Italy)

AGENDA

- SQL... o NoSQL?
- MongoDB in breve
- FerretDB: cos’è e a cosa serve
- FerretDB: come è fatto
- Demo time!
- Conclusioni finali
- Risorse e approfondimenti
- Q & A

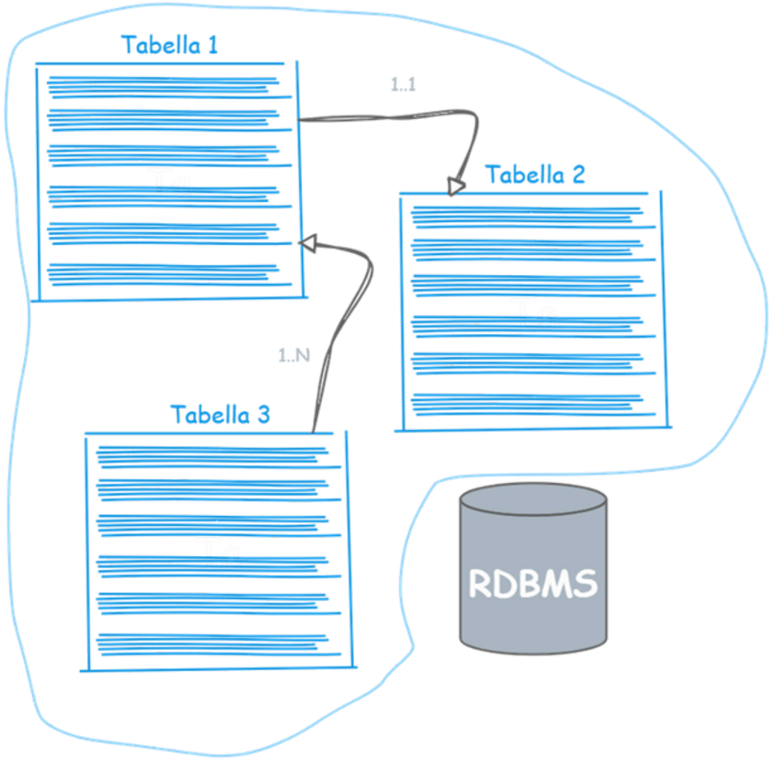
UNA PREMESSA...

SQL... O NO-SQL?

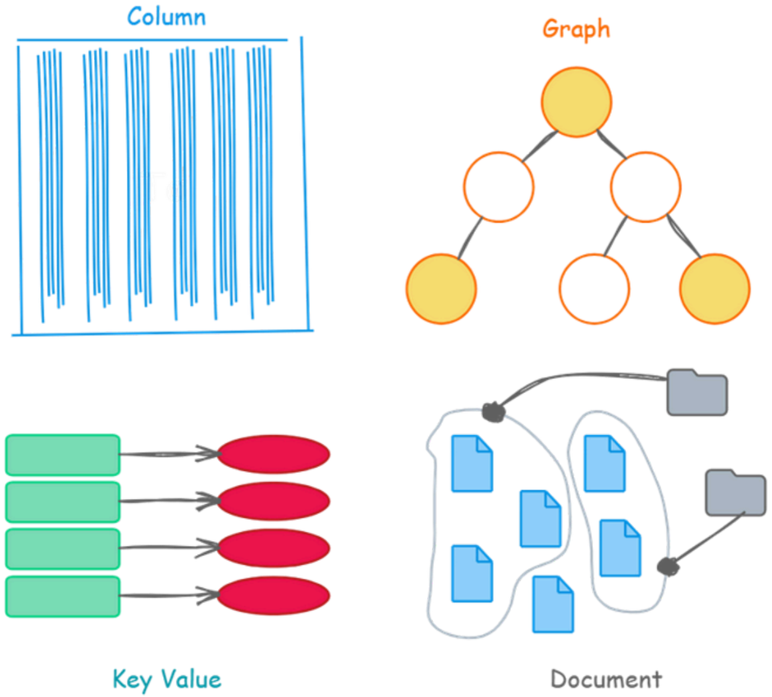
...questo è il dilemma!

SQL VS NOSQL

DATABASE SQL (RDBMS)



DATABASE NOSQL



SQL VS NOSQL: LE DIFFERENZE

SQL

- Sfruttano il linguaggio SQL
- Basati su tabelle, colonne e righe
- Schema rigido e predeterminato
- Sfruttano le transazioni (ACID)
- Supportano query basate su join
- Devono essere normalizzati
- Scalano verticalmente

NOSQL

- Usano linguaggi “non solo” SQL
- Basati su strutture non tabellari
- Schema flessibile e destrutturato
- Uso di transazioni non richiesto
- Non richiedono join complessi
- Usano forme “denormalizzate”
- Scalano orizzontalmente

SQL VS NOSQL: SCENARI D'USO

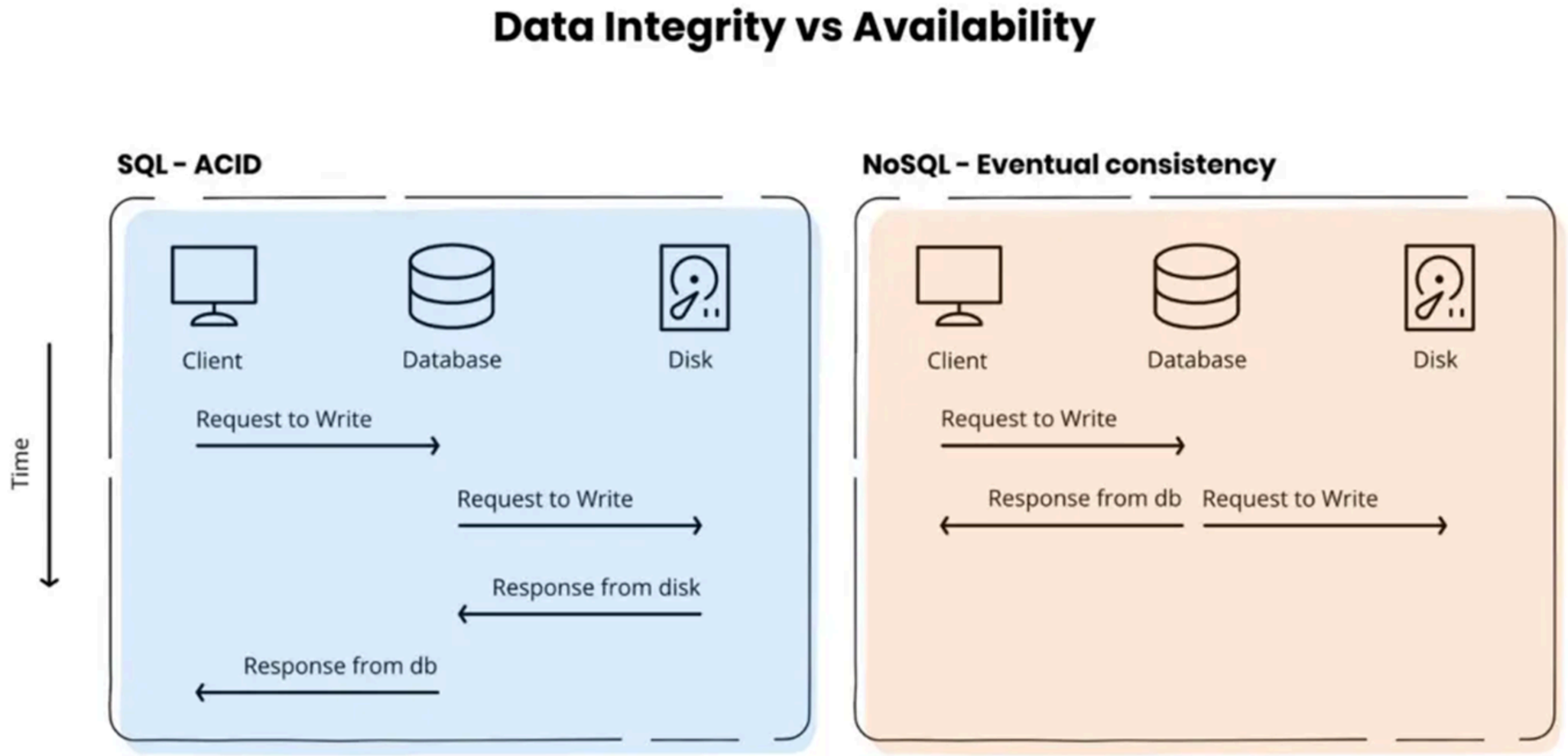
SQL

- Dati estremamente strutturati
- Integrità dei dati essenziale
- Riduzione di dati duplicati
- Accesso basato su ORM
- Schema rigido e creato up-front
- Quantità di dati limitata

NOSQL

- Dati in formato variabile
- Scalabilità dei dati essenziale
- Tolleranza e duplicazione dei dati
- Mappatura diretta su oggetti
- Schema flessibile e dinamico
- Scenario che coinvolge *Big Data*

SQL VS NOSQL



Un esempio

SQL VS NOSQL

Qualche nome...

CONOSCIAMO MONGODB

MONGODB IN BREVE

MONGODB È UN DATABASE NOSQL DI TIPO “DOCUMENT ORIENTED”

- **Document**: è il singolo documento che contiene i dati
  - Struttura formata da coppie chiave/valore
  - E’ rappresentato da un oggetto simil-JSON
  - Viene salvato su disco in formato BSON

- **Collection**: è un contenitore di uno o più documenti
  - Simile alla tabella di un classico RDMS (quasi)
  - Ha una struttura schema-less (contiene oggetti di qualsiasi tipo)

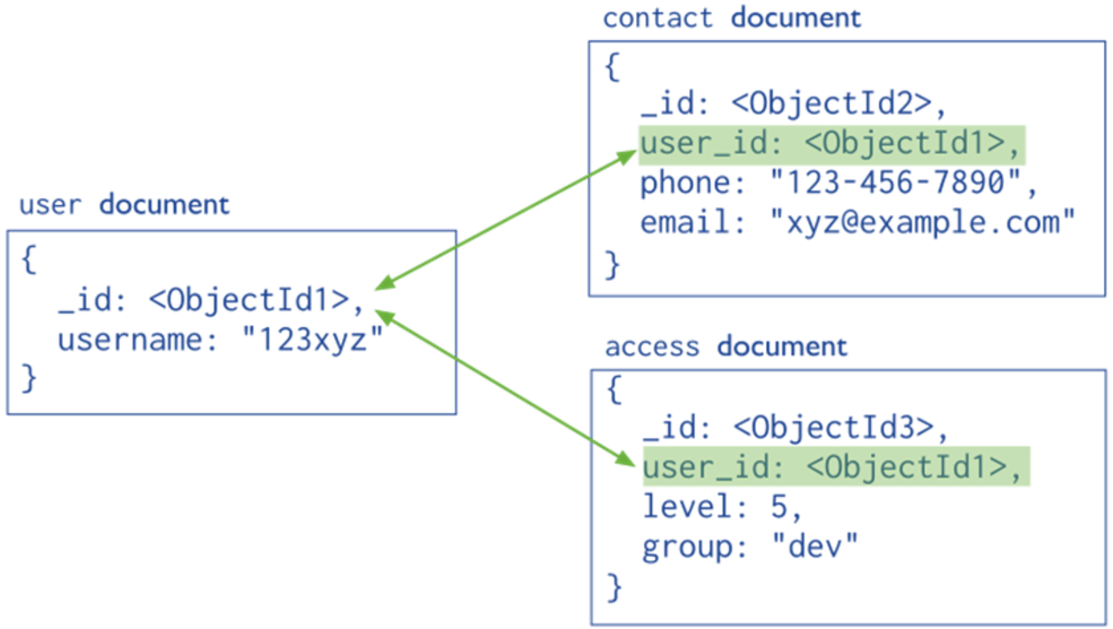
- **Database**: ciascun database può contenere una o più collection

ESEMPI DI DOCUMENTO

```
{
  _id: <ObjectId1>,
  username: "123xyz",
  contact: {
             phone: "123-456-7890",
             email: "xyz@example.com"
           },
  access: {
             level: 5,
             group: "dev"
           }
}
```

Embedded sub-document

Embedded sub-document



PUNTI DI FORZA

- E’ un database di tipo “document oriented” molto accessoriato
- Ha uno Schema aperto è flessibile (e pure validabile)
- Gestisce indice per velocizzare ricerche e aggiornamenti
- Consente interrogazioni tramite un Query Language
- Adotta un Aggregation Framework per le estrazioni di dati più complesse
- Sono disponibili tantissimi Driver per svariati linguaggi di programmazione
- Supportato da tantissimi anni e molto apprezzato dalla community

DEMO TIME

CAMBIO LICENSING

- Prima del **2018**
  - ***AGPL - GNU Affero General Public License***: se modifichi il codice di MongoDB, devi rilasciare pubblicamente il codice modificato.
- Dopo il **2018**
  - ***SSPL - Server Side Public License***: se esegui MongoDB come servizio, devi rilasciare la parte di codice che usa MongoDB come servizio.

BENVENUTO!
FerretDB

FERRETDB: UNA VERA ALTERNATIVA A MONGODB

- **E’ una soluzione completamente “open source”!**
  - *Sfrutta come backend solo prodotti open-source*
  - *E’ stato scritto in linguaggio Go*

- **Traduce le richieste effettuate con il protocollo di MongoDB in query SQL, usando PostgreSQL come motore di database**

- **E’ compatibile con tutti i driver, tool e app per MongoDB**
  - *Consente di usare lo stesso linguaggio di MongoDB, rimanendo tuttavia open-source*
  - *Può essere usato come sostituto diretto di MongoDB 6.0+*

- **Il team aggiunge costantemente nuove feature per incrementarne la già elevata compatibilità dietro feedback degli sviluppatori**
  - *Sono in corso di sviluppo nuove estensioni per compatibilità con altri backend*

- **La prima release GA (v1.0) è stata lanciata ad aprile 2023.**

PERCHE' FERRETDB?

- Consente di lavorare con i driver MongoDB già disponibili e collaudati, senza modifiche al codice esistente
- Supporta framework e tool progettati per MongoDB
- Offre una soluzione che è completamente open-source!
- Apre la strada potenziale all'uso di altri backend (è un layer software).
- Pur usando PostgreSQL (o SQLite) non richiede esperienza con questi database (seppur consigliata).

LE SFIDE DA AFFRONTARE

- **MongoDB usa il formato BSON**
  - E’ un JSON in formato binario
  - Supporta tipi aggiuntivi oltre a quelli standard
  - L’ordine di chiavi e valori va mantenuto

- **Dialogo basato su formato JSON**
  - E’ necessario leggere e scrivere documenti JSON in arrivo
  - Occorre trasformare BSON in JSON e viceversa

- **Necessità di persistenza dei dati**
  - I documenti vanno letti e scritti su uno storage, o meglio un backend
  - Il backend è intercambiabile

- **Interrogazione della base dati**
  - Occorre implementare una business logic per query e altre operazioni (es. CRUD)

- **Modello a oggetti**
  - Servono DTO per memorizzare e convertire i dati in ingresso e in uscita

GLI “STRATI” DI FERRETDB

COMPONENTI DEL SISTEMA

APPLICATION (DRIVER) LAYER

- **Legge e scrive documenti in formato BSON compatibili con MongoDB**
  - BSON è un formato JSON binario
  - Supporta tipi aggiuntivi oltre a quelli standard JSON
- **Trasforma i documento in un formato gestibile dal business**
  - E’ necessario salvare i dati in memoria
  - I dati devono essere trattabili internamente dal business layer

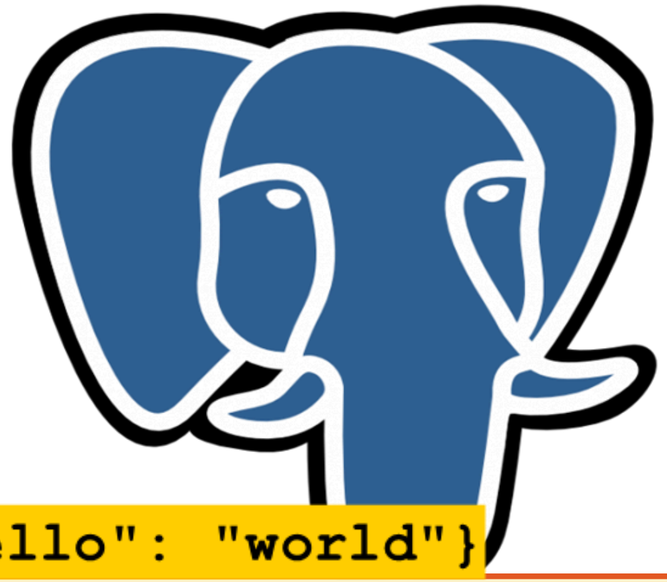
```
{"hello": "world"}
```

```
\x16\x00\x00\x00                  // total document size
\x02                              // 0x02 = type String
hello\x00                         // field name
\x06\x00\x00\x00world\x00         // field value
\x00                              // 0x00 = type EOO ('end of object')
```

BUSINESS LAYER

- **Go è un linguaggio open source by Google**
- **Semplice da programmare, efficiente, dalla compilazione veloce**
- **I dati vengono gestiti da un business layer sviluppato in linguaggio Go**
  - Consente di tenere i dati in memoria
  - I tipi dei campi sono quelli nativi di Go
  - Go dispone di un *Garbage Collector*
- **Include gli handler di comandi e la logica generale di processing**
  - Le funzionalità sono implementate usando le API distribuite con Go
  - Per le funzionalità legate a JSON, BSON, ecc. si usano package open-source
- **Offre diversi punti di estensibilità (es. il supporto a diversi backend)**

BACKEND LAYER

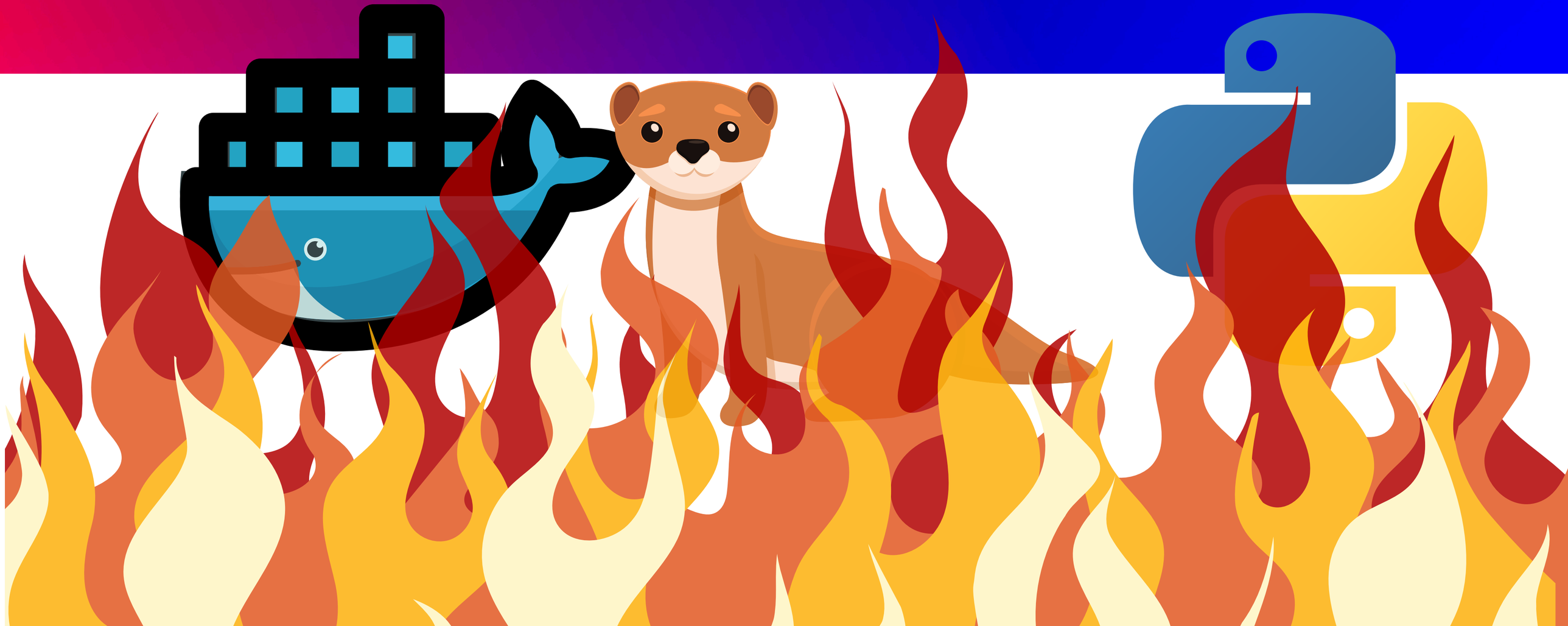


```
{"hello": "world"}
```

```
{
  "$s": {
    "p": {
      "hello": {"t": "string"}
    },
    "$k": ["hello"]
  },
  "hello": "world"
}
```

- **Estensione del business model implementato in Go**
  - Strutturato a interfacce intuitive (*Backend*, *Collection*, *Database*, ...)
  - Gli oggetti del modello sono «as small as possible»
- **Serializzazione in JSON e storage**
  - Gli oggetti Go vengono salvati in JSON e poi immagazzinati su backend (es. *PostgreSQL*)
  - Il backend PostgreSQL può essere sostituito da altri (es. *SQLite*... e qualcuno sta già sviluppando quello per *MySQL*!)
  - L'ordine delle chiavi e dei valori deve essere mantenuto (incorpora uno Schema)
- **Perché PostgreSQL come prima scelta?**
  - Open source, maturo e affidabile
  - Supporto tecnico eccellente (ufficiale o community)
- **Supporta nativamente JSON (inclusi indici e query)**

DEMO TIME



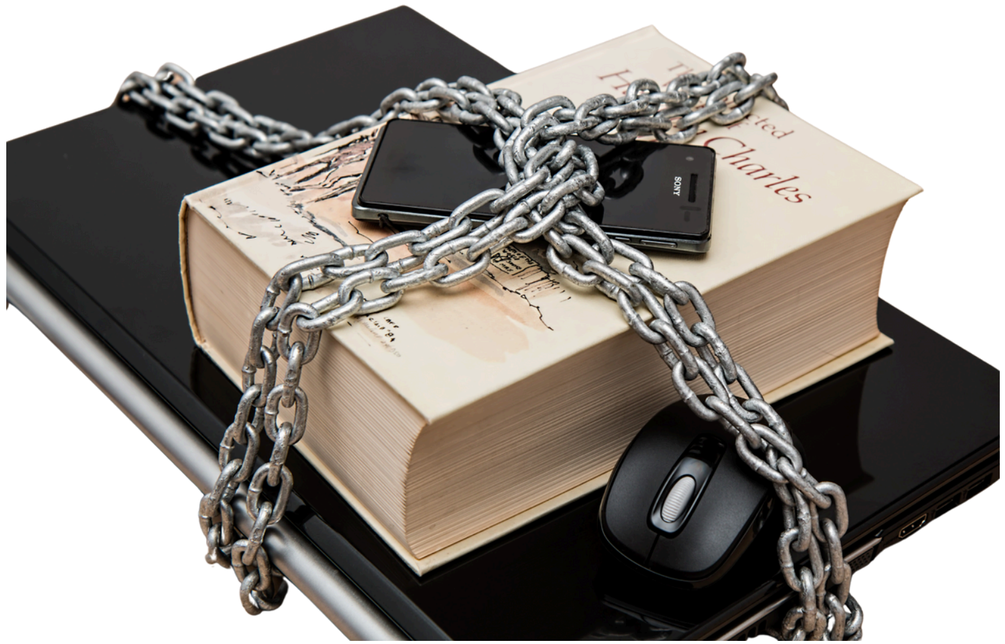
CONCLUSIONI

L’obiettivo di FerretDB è massimizzare l’accessibilità delle funzionalità di storage e la compatibilità con MongoDB per costituire una valida alternativa open source a quest’ultimo

E' LIMITATO?

- **FerretDB è diverso da MongoDB su alcuni aspetti**
  - Comparazione non uniforme tra "jsonb" (PostgreSQL) e BSON (MongoDB)
  - Alcuni valori e tipi sono assenti o diversamente rappresentati (es. NaN, null, nested arrays, negative zero, Infinity, ...)*
  - Gestione delle sessioni
  - Richieste tradizionali e aggregation

- **Mancano alcune macro-feature**
  - Sharding e Clustering
  - Backup, Restore e Administration
  - Altre feature di "classe Enterprise" (alcune di queste sono implementate da PostgreSQL)



È STABILE?

- **FerretDB è sottoposto a rigorosi test legati allo sviluppo e alla solidità del tool**
  - Presenti test specifici per «use case» particolari
  - Alcuni test scritti dai mantainer stessi, altri da sviluppatori backend
  - Comprende suite di test lunghi eseguiti su shard dedicati

- **Non tutto l'ecosistema FerretDB deve essere testato**
  - Test dei driver non necessari (già eseguiti dai developer che li creano per MongoDB)
  - PostgreSQL in sé e package open-source già sottoposti a loro test dedicati

- **Stabilità del runtime di Go**
  - Runtime testato da Google
  - Pipeline e tool di testing sono quelli di Go (non vanno creati ad hoc)

È SUPPORTATO?

- **E’ open source!**
  - Si può contribuire al progetto se si conosce il linguaggio Go
  - Si possono segnalare issue direttamente su GitHub

- **Il team adotta metodologie e tool globalmente noti e diffusi**
  - Container (Docker), Task Runner (go-task), Linters (go-lint), versioning, regole nella gestione del ciclo di sviluppo
  - GitHub Actions per linting, test, security scan, action dedicate (anch’esse pubbliche e disponibili su GitHub!)

- **FerretDB è il «core product» della startup che lo produce (*FerretDB Inc.*)**

IMHO...

- Su Windows il supporto è limitato: ad esempio, si compila a mano
- Mancano feature interessanti di MongoDB (es. Capped Collections)
- L'amministrazione è diversa rispetto a MongoDB
- Adatto a scenari dove non abbiamo a che fare con grossi volumi di dati (inteso come «non Big Data»)
- Progetto recente e (al momento) soggetto a molte modifiche (molto pioneristico a mio avviso)
  -

- Installato su sistema operativo Linux oppure in Docker è perfetto!
- Le feature più usate ci sono tutte e funzionano su diversi backend
- Per chi conosce PostgreSQL tutto risulta più semplice e familiare
- Può essere usato anche in «modalità Embedded» con SQLite e pure integrato nell'eseguibile!
- Progetto gestito bene, interessante, non banale, curato e molto promettente! **Ed è open-source!**

RISORSE E APPROFONDIMENTI

FerretDB

Sito ufficiale 👉 https://www.ferretdb.com
Repository su GitHub 👉 https://github.com/FerretDB
Blog 👉 https://blog.ferretdb.io
Documentazione 👉 https://docs.ferretdb.io
Roadmap ufficiale 👉 https://github.com/orgs/FerretDB/projects/2
Store 👉 https://store-ferretdb.myshopify.com

MongoDB

Sito ufficiale di MongoDB 👉 https://www.mongodb.com
Webinar introduttivo 👉 https://www.youtube.com/watch?v=vk0WDErLkKU

Domande e risposte

GRAZIE! 🤗