Paweł Rajba
pawel@cs.uni.wroc.pl
http://pawel.ii.uni.wroc.pl/

MongoDB

Agenda

- NoSQL introduction
- MongoDB
  - Getting started
  - Shell
  - Storage
  - Structure
  - Manipulating data
  - Quering data
  - Replication
  - Client application

NoSQL Introduction

- New trend in a database world
- NoSQL stands for „Not only SQL"
- Different mindset
  - No joins
  - No complex transactions
- Easily scalable
- ACID in a node, BASE outside the node

Why?

- Simple flexible structure (no schemas)
- Easier scalable (mostly horizontally)
- Performance (especially for write and delete)
- Indexing
- Easy replication and failover suport
- … and many others

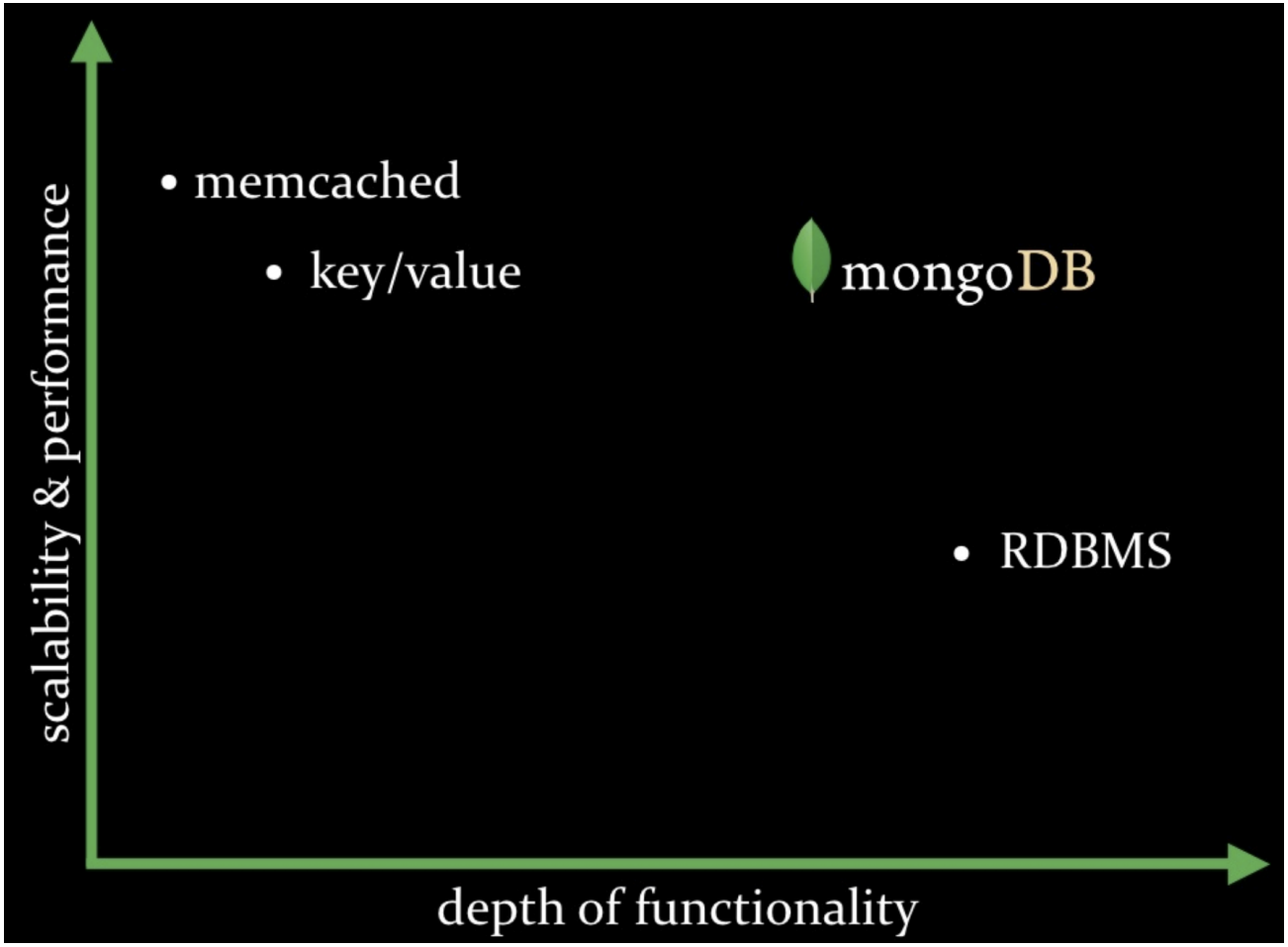
Types of NoSQL

- Key Value Store
  - Asdf
- Document Store
  - MongoDB
- Column Store
  - Cassandra
- Graph Database
  - Neo4J

MongoDB

- NoSQL DB
- Open source
- Document DB
- No schema mandatory
- Indexing
- Highly scalable
- MapReduce
- Easy Replication and sharding

MongoDB position



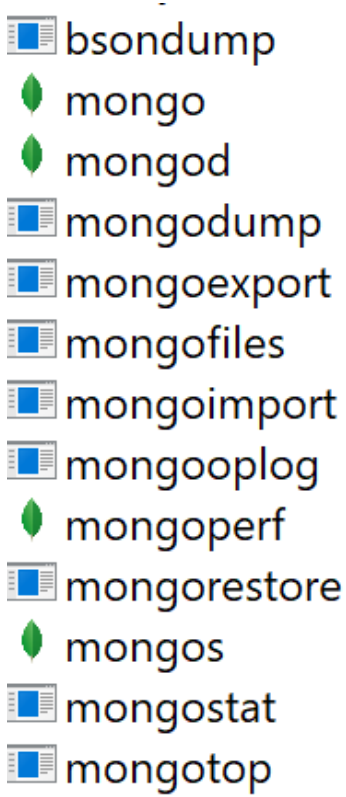
Getting started

- Web site: https://www.mongodb.com/
- Download: https://www.mongodb.com/download-center
- Different binaries: https://www.mongodb.org/dl/win32/x86_64-2008plus-ssl
- We can download latest stable version:

name	modified	size	md5	sig	sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-v3.4-latest-signed.msi	2018-06-08 19:41:25	159330816	md5		sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-debugsymbols-v3.4-latest.zip	2018-06-08 19:41:20	189116760	md5	sig	sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-v3.4-latest.zip	2018-06-08 19:41:15	247077361	md5	sig	sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-debugsymbols-v3.6-latest.zip	2018-06-08 18:58:40	248209864	md5	sig	sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-v3.6-latest-signed.msi	2018-06-08 18:58:34	192792064	md5		sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-v3.6-latest.zip	2018-06-08 18:58:26	306956817	md5	sig	sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-debugsymbols-4.0.0-rc4.zip	2018-06-07 15:25:38	177274152	md5	sig	sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-4.0.0-rc4-signed.msi	2018-06-07 15:25:28	194460160	md5		sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-4.0.0-rc4.zip	2018-06-07 15:25:19	228693792	md5	sig	sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-debugsymbols-4.0.0-rc3.zip	2018-06-06 15:12:42	177259827	md5	sig	sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-4.0.0-rc3-signed.msi	2018-06-06 15:12:37	194444288	md5		sha1	sha256
win32/mongodb-win32-x86_64-2008plus-ssl-4.0.0-rc3.zip	2018-06-06 15:12:30	228679154	md5	sig	sha1	sha256

Getting started

- Files we get in archive →
- Default
  - data folder: \Data\Db
  - port: 27017
- Options:
  - mongod --help
- Simplest usage:
  - mongod
  - mongo
- Better usage:
  - mongod --dbpath c:\Data\Databases\MongoDB
  - mongo

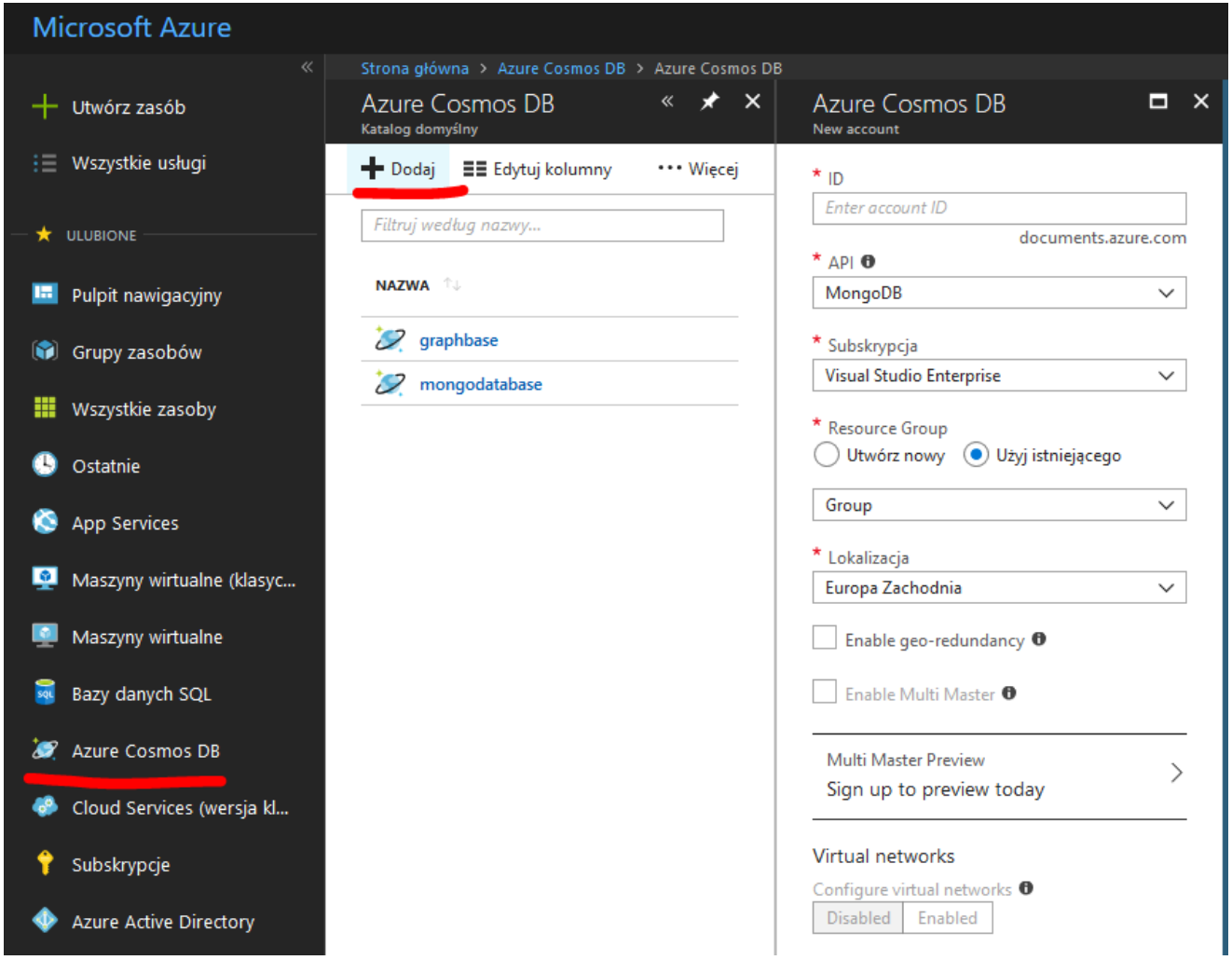


Getting started

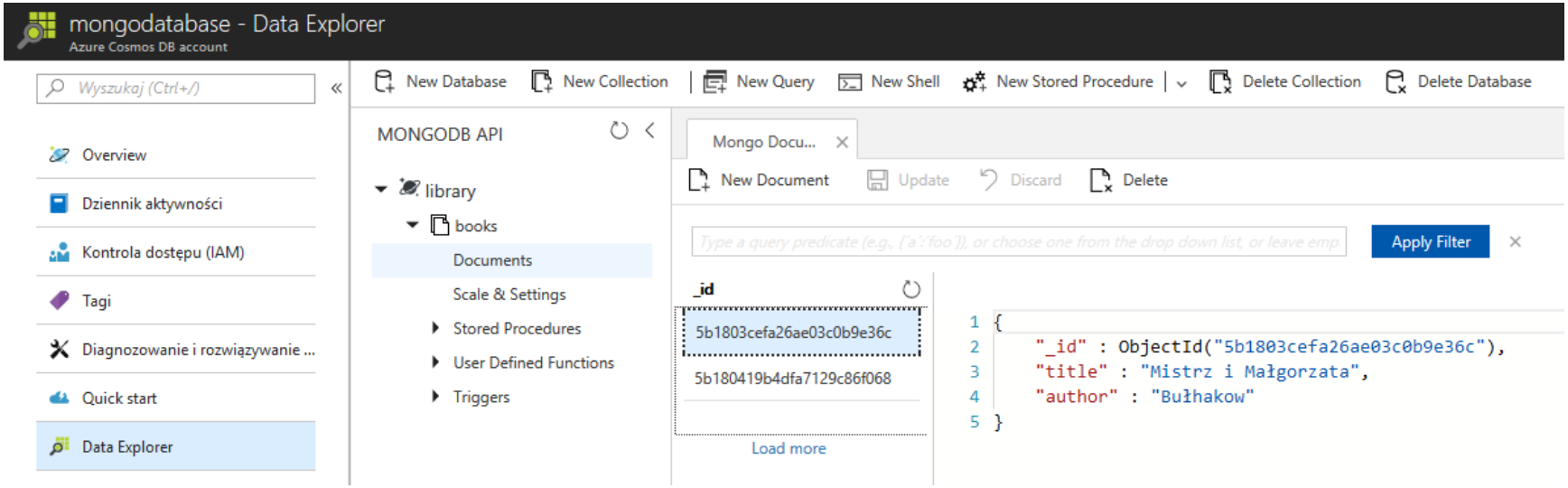
- Better usage:
  - mongod --dbpath c:\Data\Databases\MongoDB\db
  - mongo
- Best usage:
  - mongod -f c:\Data\Databases\MongoDB\mongod.conf
- Config file:

```
1 dbpath=C:\Data\Databases\MongoDB\db
2 logpath=C:\Data\Databases\MongoDB\mongo-server.log
3 verbose=vvvv
```

Getting started (Azure)



Getting started (Azure)



Shell

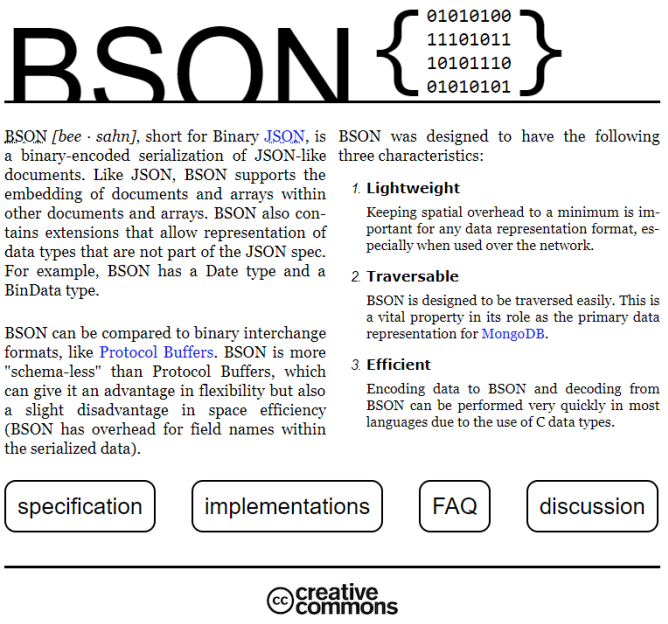
- mongo.exe
- JavaScript interpreter
- Multiline input is ok

```
> var hello = function() {
... print("Hello World!");
... }
> hello();
Hello World!
```

- Run external script
  - mongo C:\Data\Databases\MongoDB\booksCount.js
  - > load('C:/Data/Databases/MongoDB/booksCount.js')
- Non-interactive mode
  - mongo localhost/admin --eval "printjson(db.runCommand({logRotate:1}))"

Storage

- Mongo talk with JSONs
  - JSONs can be easily imported and queried
- Documents stored in BSON
  - http://bsonspec.org/



Structure

- Structure hierarchy
  - Instance → Databases → Collections → Documents
- Collection optionally may have a schema
  - https://docs.mongodb.com/manual/core/schema-validation/
- Rule: every document must have a key
  - _id
    - Present in all documents
    - Unique across collection
    - Any type (except array)

Manipulating data

- Basic
  - show dbs
  - use library
  - db
  - db.books.save({_id:1, author:"Bułhakow", title:"Mistrz i Małgorzata"})
  - db.books.save({_id:2, author:"Golden", title:"Wyznania gejszy"})
  - db.books.save({_id:3, author:"Golding", title:"Władca much"})
  - db.books.find()
- ObjectId
  - db.books.save({author:"Rowling", title:"Harry Potter"})
  - ObjectId()
  - ObjectId().getTimestamp()
- Insert command
  - db.books.save({_id:3, author:"Orwell", title:"Folwark zwierzęcy"})
  - db.books.save({_id:3, author:"Golding", title:"Władca much"})
  - db.books.insert({_id:3, author:"Orwell", title:"Folwark zwierzęcy"})
  - db.books.insert({_id:4, author:"Orwell", title:"Folwark zwierzęcy", rating:8})
- More
  - https://docs.mongodb.com/manual/tutorial/insert-documents/

Manipulating data

- Update problem with save
  - > var b = db.books.findOne({_id:4})
  - > b.rating = b.rating+1;
    - here someone else gets the book and modify rating
  - > db.books.save(b);

  - > db.books.save({_id:4, author:"Orwell", title:"Folwark zwierzęcy", rokWydania:1945})
  - -- and then
  - > db.books.save(b);

Manipulating data

- Update command
  - db.col.update(query, update, options)
- Examples
  - db.books.update({_id:4}, {$inc:{rating:1}});
- Operators
  - $inc{rating:1}
  - $set:{y:3}
  - $unset:{y:0}
  - $rename:{'rko': 'rok’}
- More
  - https://docs.mongodb.com/manual/tutorial/update-documents/

Manipulating data

- Delete
  - db.books.deleteMany({})
  - db.inventory.deleteOne({ _id: 3})
- More
  - https://docs.mongodb.com/manual/tutorial/remove-documents/

Quering data

- Query
  - db.col.find(query, projection)
    - Projection: {field:0|1, field:0|1, ...} (all 0 or all 1)
- More
  - https://docs.mongodb.com/manual/tutorial/query-documents/
- Extend our data

```
db.books.save({_id:4, author:"Orwell", title:"Folwark zwierzęcy", year:1945, rating:8,
location:{room:4,segment:2}, catalogue:
[
{number:"A01", available:true},
{number:"A02", available:false, rentDate:'2018-01-01'},
{number:"A03", available:true},
]
})
db.books.save({_id:5, author:"Steinbeck", title:"Grona gniewu", rating:7, location:{room:4,segment:3},
catalogue:
[
{number:"A04", available:false, rentDate:'2018-01-02'},
{number:"A05", available:false, rentDate:'2018-01-03'},
]
})
```

Quering data

- Basic queries
  - > db.books.find({_id:4});
  - > db.books.find({_id:4}, {_id:1});
  - > db.books.find({_id:4}, {_id:0});
  - > db.books.find({_id: {$gt:2}})
  - > db.books.find({_id: {$not:{$gt:2}}})
  - > db.books.find({_id: {$in:[1,2]}})
  - > db.books.find({_id: {$nin:[1,2]}})
  - > db.books.find({author:/^Gold/});
  - > db.books.find().count()

Quering data

- Nested documents
  - > db.books.find({"location.room":4});
  - > db.books.find({"catalogue.available":true}, {_id:1});
  - > db.books.find({"catalogue.available":false}, {_id:1});
- Where
  - > db.books.find({$where: "this.author=='Golden' || this.title=='Władca much'"});
- Sorting
  - > db.books.find({}, {title:1}).sort({'catalogue.available':-1,title:1});
- Paging
  - > db.books.find({},{_id:1}).sort({_id:1}).skip(2).limit(2);

Quering data

- Iterating cursor

  > var c = db.books.find({},{title:1});

  > c.size()

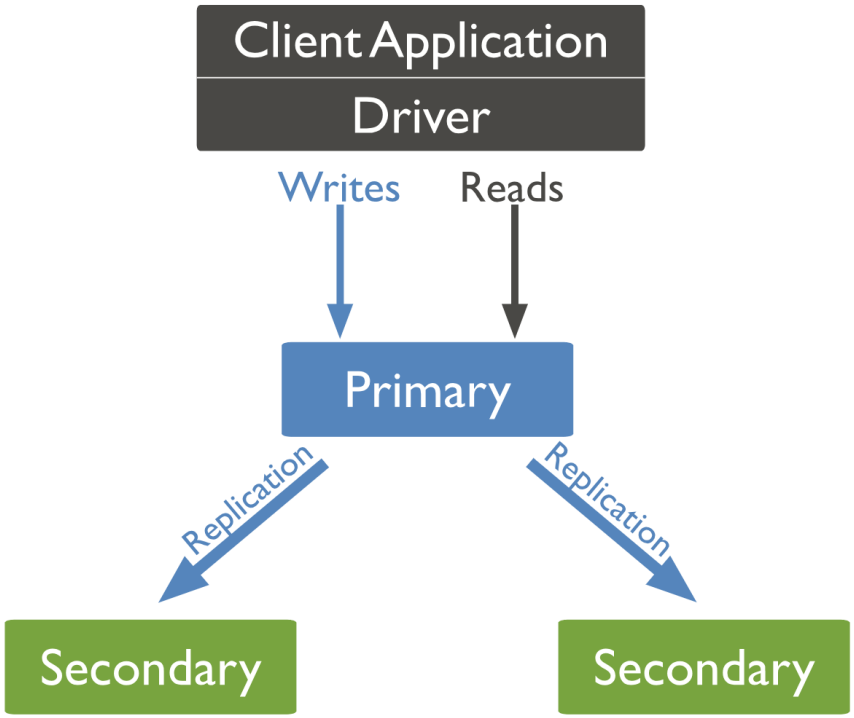
  > c.hasNext()

  > c.forEach(function(d){ print(d.title); })

  > c.hasNext()

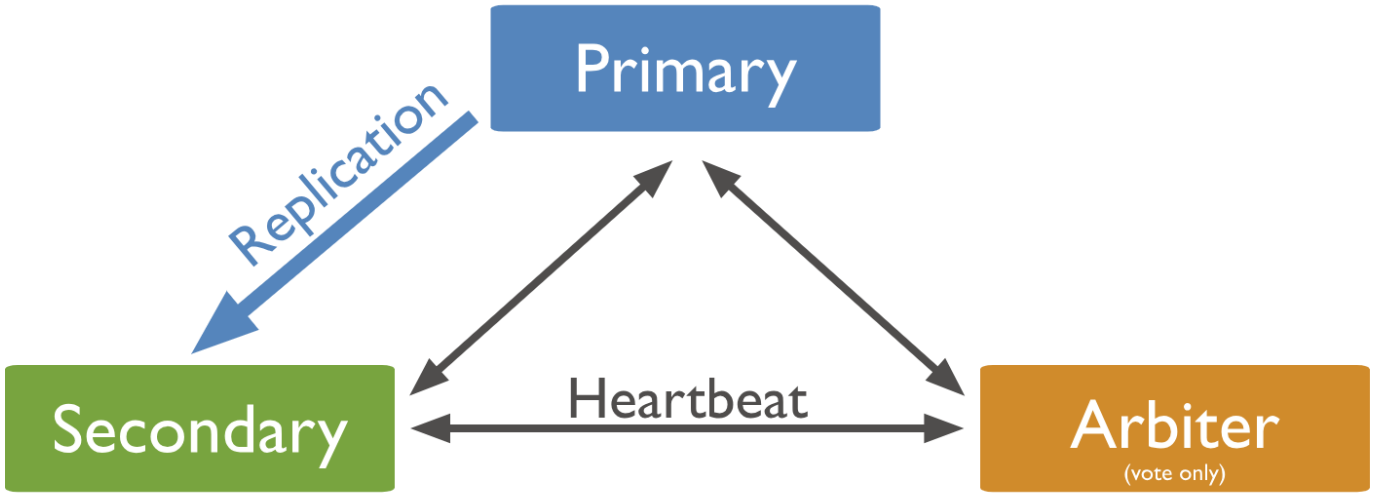
Replication

- Replica Set concept



Replication

- Role of arbiter
  - Doesn't have data, can be weak node
  - Support voting for primary in case there are even numer of nodes



Replication

- DEMO
  - Create folders
    - c:\Data\Databases\MongoDB\db1
    - c:\Data\Databases\MongoDB\db2
    - c:\Data\Databases\MongoDB\db3
  - Run 3 instances
    - start "A" mongod --dbpath c:\Data\Databases\MongoDB\db1 --port 10000 --replSet "demo"
    - start "B" mongod --dbpath c:\Data\Databases\MongoDB\db2 --port 20000 --replSet "demo"
    - start "C" mongod --dbpath c:\Data\Databases\MongoDB\db3 --port 30000 --replSet "demo"
  - Run shell
    - mongo --port 10000

Replication

- DEMO
  - Create a configuration object
    (more https://docs.mongodb.com/manual/reference/replica-configuration/)
    - var rsConfig={ _id: "demo", members: [{_id: 0, host: 'localhost:10000', priority: 10}, {_id: 1, host: 'localhost:20000'}, {_id: 2, host: 'localhost:30000', arbiterOnly: true}]};
  - Let's take a look
    - rsConfig
  - Initiatiate a cluster
    - rs.initiate(rsConfig)

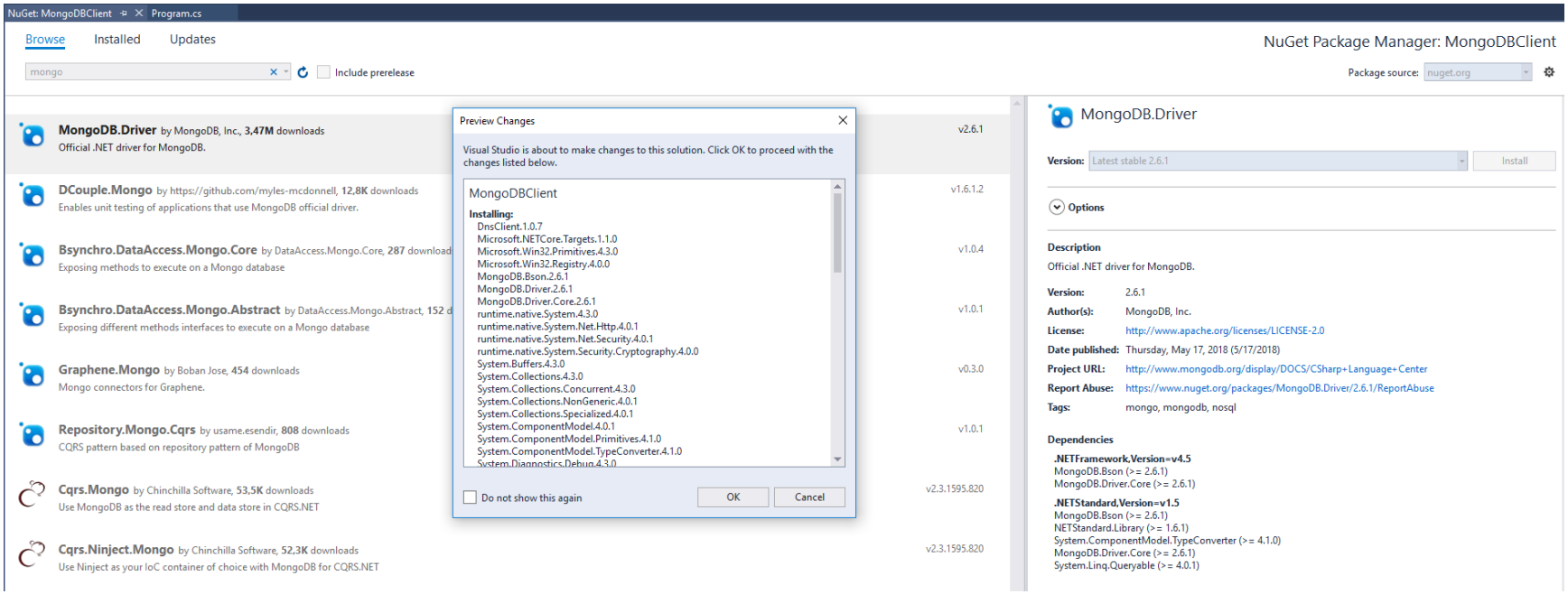
Replication

- DEMO
  - Save something
    - db.books.save({_id:1, title:"Mistrz i Małgorzata"})
    - db.books.find()
  - Let's check second server
    - mongo --port=20000
    - db.books.save({_id:2, title:"Wyznania gejszy"})
    - db.books.find();
    - db.setSlaveOk();
    - db.books.find();

Replication

- DEMO
  - Let's check replication
    - Kill PRIMARY
    - Check SECONDARY
  - Resurrect PRIMARY
    - start "A" mongod --dbpath c:\Data\Databases\MongoDB\db1 --port 10000 --replSet "demo"
  - Check again

Client application



Other interesting stuff

- Indexing
- Aggregation
  - https://docs.mongodb.com/manual/aggregation/
- Views
- MapReduce
- Capped Collections
- Geo

References

- Introductions
  - https://www.mongodb.com/nosql-explained
  - https://www.slideshare.net/Leesy/an-introduction-to-nosql-mongodb/
  - https://www.slideshare.net/mdirolf/introduction-to-mongodb
  - https://www.slideshare.net/mongodb
  - https://www.slideshare.net/drumwurzel/intro-to-mongodb/
  - https://www.toptal.com/database/the-definitive-guide-to-nosql-databases
- Documentation
  - https://docs.mongodb.com/manual/crud/
  - https://www.tutorialspoint.com/mongodb/index.htm
- Client C# application
  - https://docs.mongodb.com/ecosystem/drivers/csharp/
  - http://mongodb.github.io/mongo-csharp-driver/2.2/reference/driver/
  - https://blog.oz-code.com/how-to-mongodb-in-c-part-1/
  - https://code.visualstudio.com/docs/azure/mongodb
- Cloud Hosting
  - https://mlab.com/