Documents and BSON

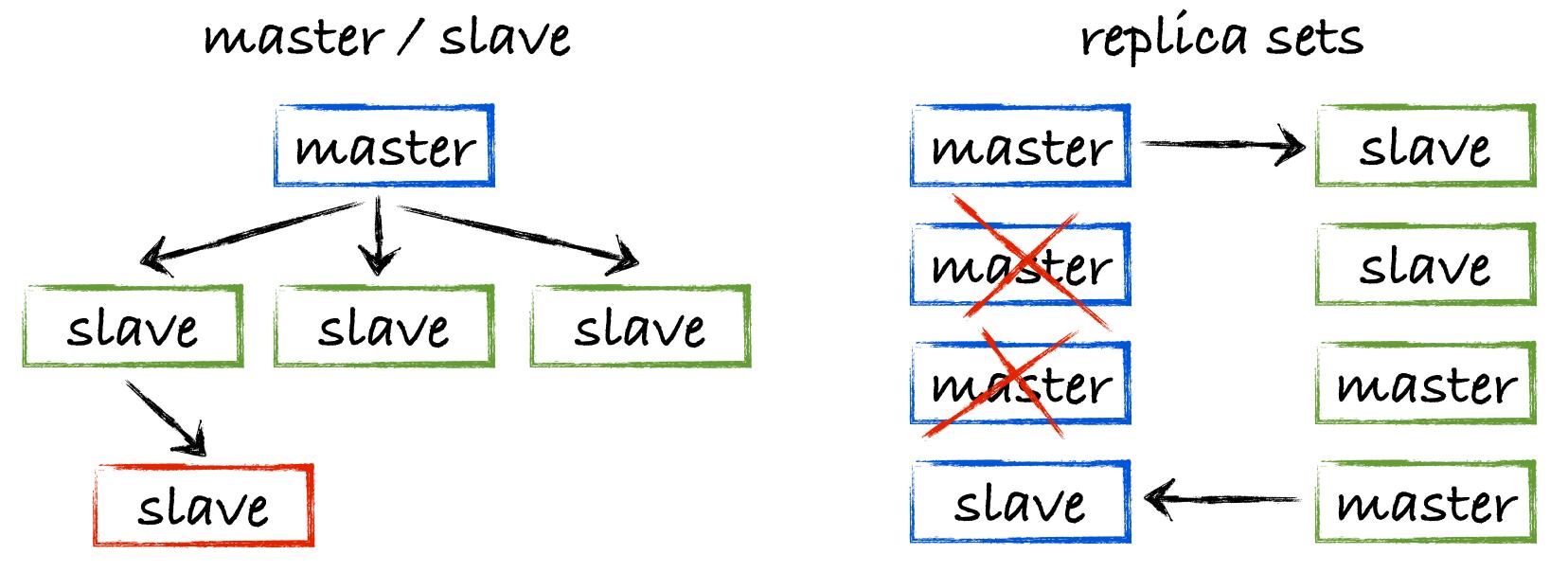
{"hello": "world"}

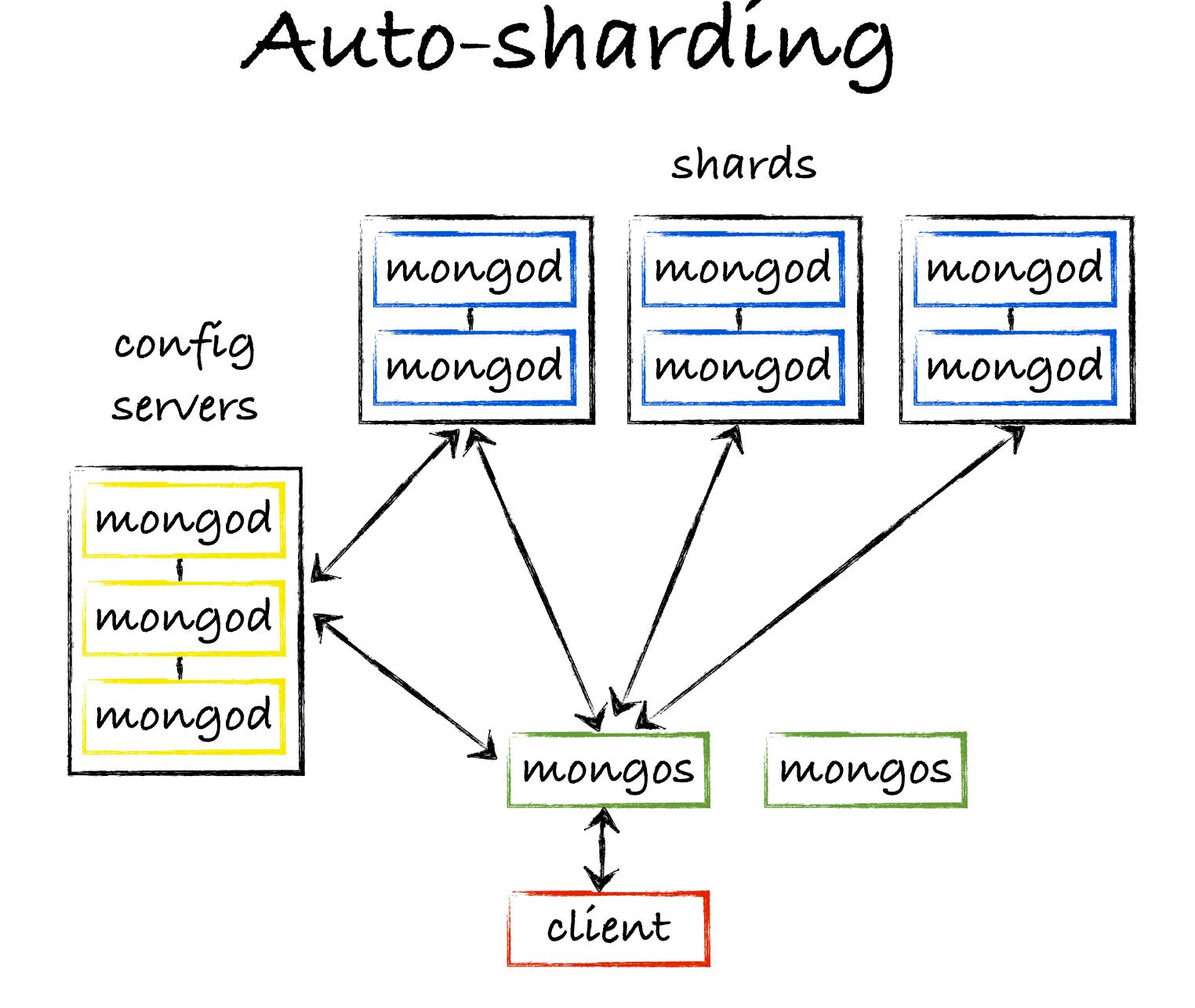
'\x16\x00\x00\x00\x00\x02hello
\x00\x06\x00\x00\x00\x00world\x00\x00'

Schema-free collections

{"hello": "world",
"foo": [{"bar": 1}]}

Replication



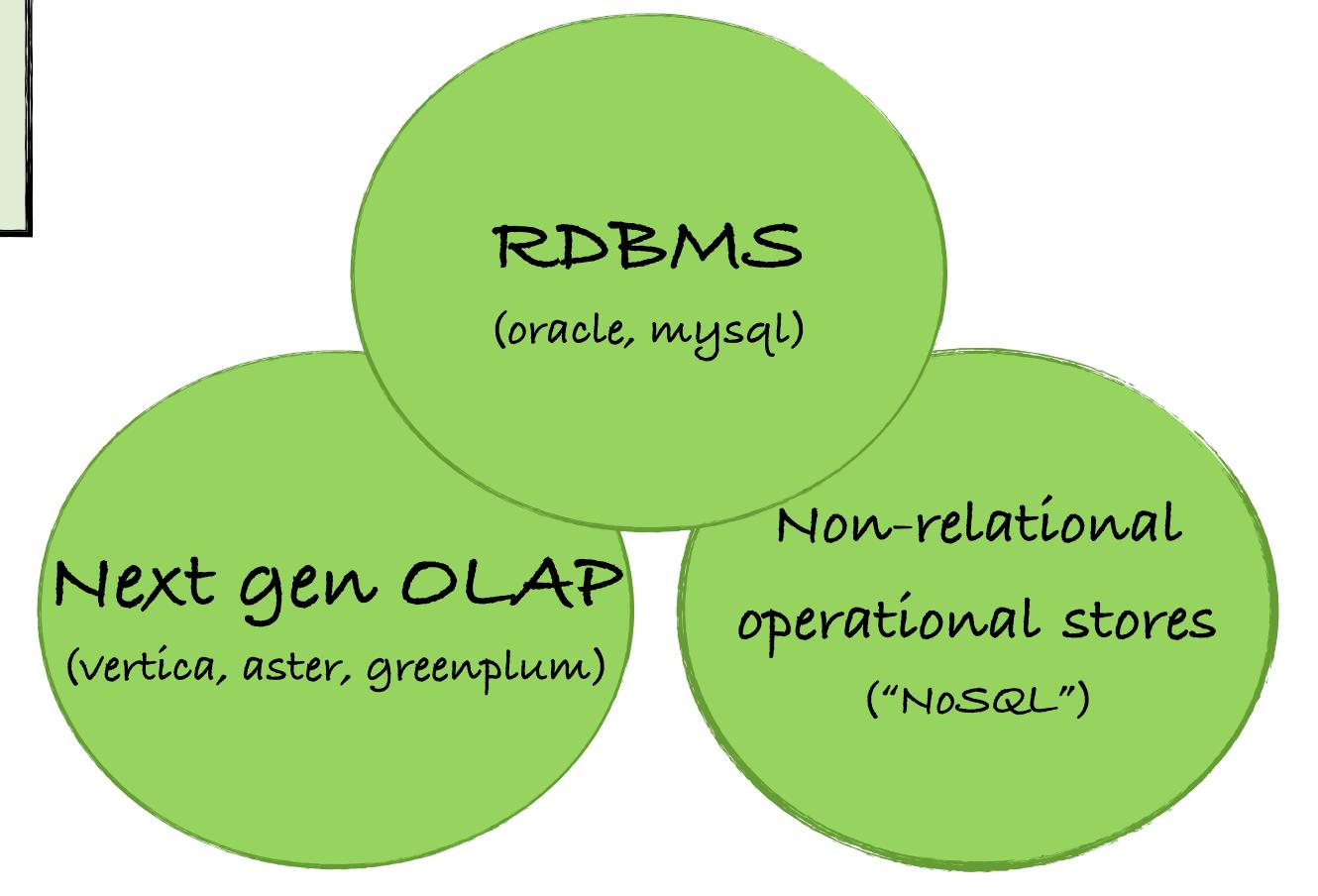




open-source, high-performance, schema-free, document-oriented database

http://www.mongodb.org
http://api.mongodb.org/python
@mongodb
@mongodb
@mongodb

"One size fits all"
no longer



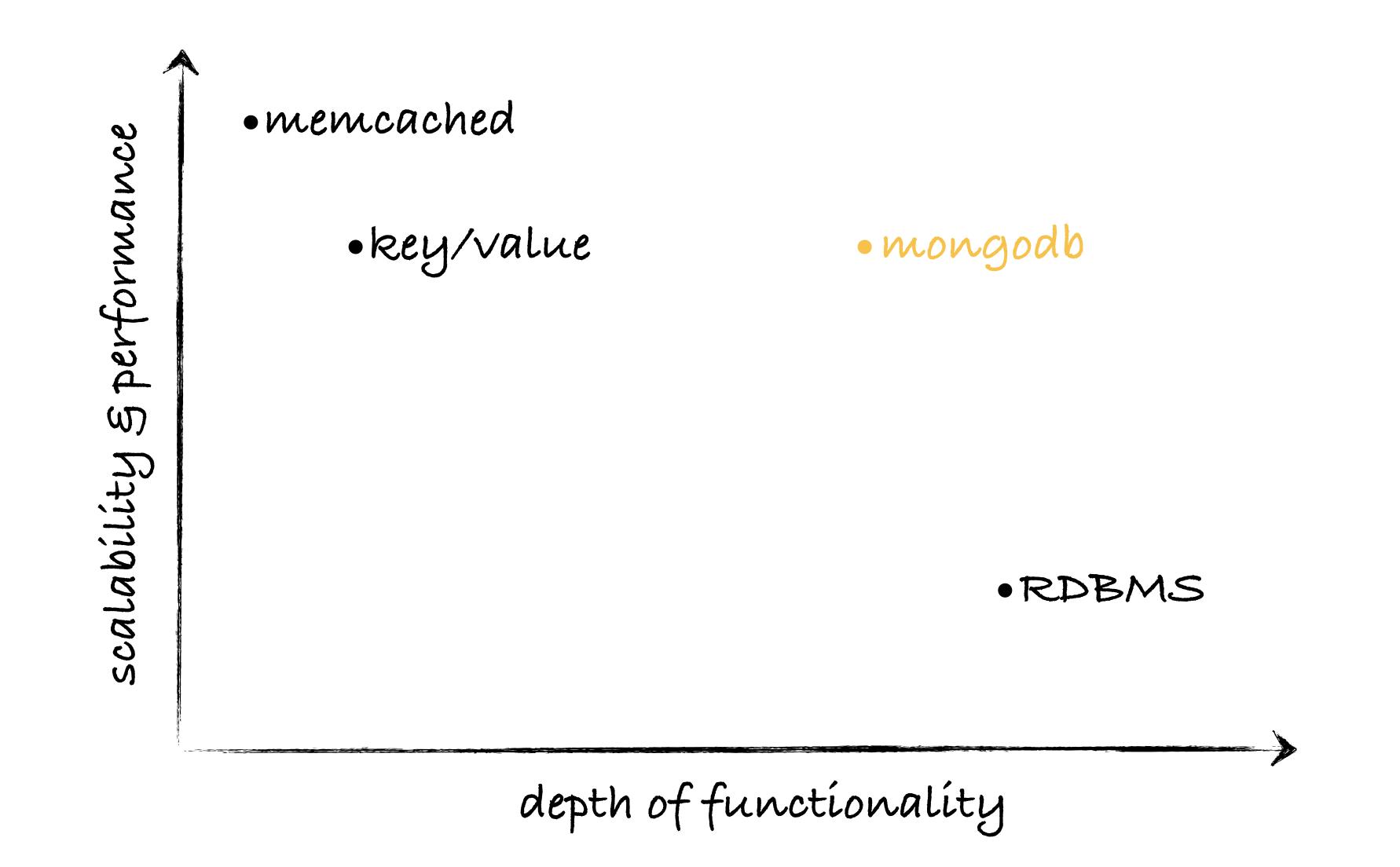
Scaling out

No joins Light transactional semantics

Side effect

New data models Improved developer experience

Focus on performance



Working with PyMongo

```
>>> import datetime
>>> import pymongo
>>> connection = pymongo.Connection()
>>> db = connection.test
>>> # Inserting documents
>>> db.posts.insert({"author": "mike",
                     "date": datetime.datetime.utcnow(),
 • • •
                     "text": "my blog post...",
                     "tags": ["mongodb", "python"]})
ObjectId('4b743bdde6fb1b969b000000')
>>> # Atomic update modifiers
>>> comment = {"author": "eliot",
               "date": datetime.datetime.utcnow(),
• • •
               "text": "great post!"}
>>> db.posts.update({"_id": pymongo.objectid.ObjectId('4b743bdde6fb1b969b000000')},
                    {"$push": {"comments": comment}})
• • •
>>> # Dynamic queries
>>> db.posts.find_one()
{u'tags': [u'mongodb', u'python'],
 u'text': u'my blog post...',
 u'author': u'mike',
 u'comments': [{u'date': datetime.datetime(2010, 2, 11, 17, 19, 10, 64000),
                       u'text': u'great post!',
                       u'author': u'eliot'}],
 u'date': datetime.datetime(2010, 2, 11, 17, 18, 21, 716000),
 u'_id': ObjectId('4b743bdde6fb1b969b000000')}
>>> db.posts.find({"author": "mike"})
<pymongo.cursor.Cursor object at 0x100529ad0>
>>> for post in db.posts.find({"author": "mike"}):
... post['text']
 • • •
u'my blog post...'
>>> # Advanced query operators
• • •
>>> db.posts.find().sort("date", pymongo.DESCENDING).limit(10)
<pymongo.cursor.Cursor object at 0x100529bd0>
>>> last_week = datetime.datetime.utcnow() + datetime.timedelta(days=-7)
>>> db.posts.find({"date": {"$gt": last_week}})
 <pymongo.cursor.Cursor object at 0x100529a10>
>>> # Reaching inside embedded lists and documents
>>> db.posts.find({"tags": "mongodb"})
<pymongo.cursor.Cursor object at 0x100529a50>
>>> db.posts.create_index("tags")
u'tags_1'
>>> db.posts.create_index("comments.author")
u'comments.author_1'
```