{SDD}

2015

Software Design
& Development

Choosing the right language for NoSQL: Static vs. Dynamic

Michael Kennedy
@mkennedy
http://blog.michaelckennedy.net

Session Outline

- The question
- Examples of static and dynamic experiences
- Quickly learn about NoSQL and DocumentDBs
- Is there such a thing as a schemaless database?
- What's NoSQL + C# like?
- What's NoSQL + Python like?
- Performance showdown
- Best practices
- Blurring the lines – dynamic data access with C#

What's the question and is there an answer?

Given that NoSQL databases are usually schemaless (think dynamically typed), is it more appropriate to use dynamic languages to interact with them?*

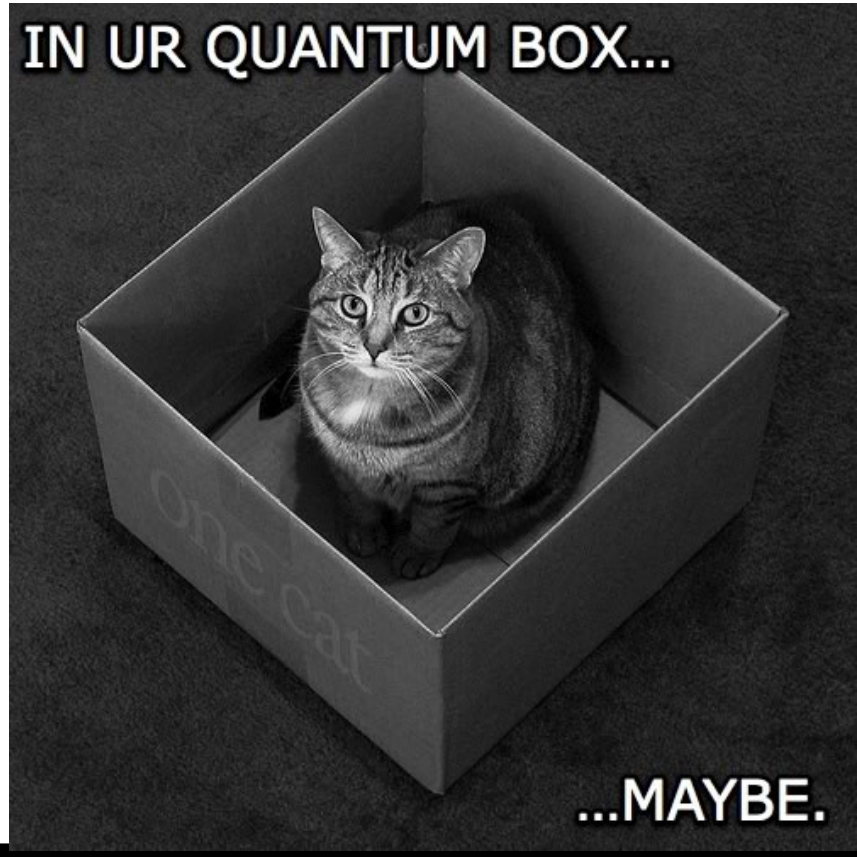
* Schemaless: We'll discuss what this means as we go.

Is there an answer?

There *might* not be a clear cut answer.

The goal is to

- Understand the trade-offs each style brings
- Look at arguments and counter-arguments
- See performance differences



Languages

Static vs. Dynamic languages

Static Languages	Dynamic Languages
C#	Python
Java	Ruby
C++	JavaScript

Typically **compiled (maybe JIT)**

Typically **interpreted**

C# vs. Python

We need to pick two concrete languages

Visual C#

C#

```
public class ShoppingCart: Entity, IEnumerable<CartItem>
{
   // ...
   public void AddItem(CartItem item)
    {
      if (item == null) {
         throw new ArgumentNullException("item");
      }
      this.Items.Add(item);
      Log("Adding new item: {0} for ${1}",
         item.Name, item.Price);
   }
}

ShoppingCart cart = new ShoppingCart();
cart.AddItem( new CartItem("Tesla", 120000) );

foreach (CartItem i in cart) {
   // use i...
}
```

Python

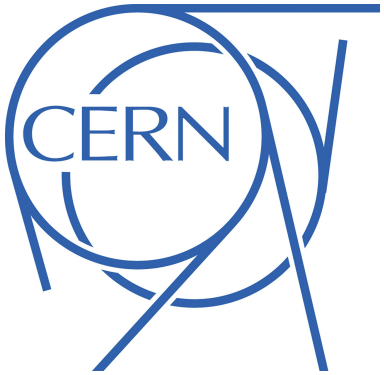
```
class ShoppingCart(Entity):
   # ...
   def AddItem(self, item):
      if item == null:
         raise TypeError("item")

      self.Items.Add(item)
      log("Adding new item: {0} for ${1}".
           format(item.Name, item.Price))


cart = ShoppingCart()
cart.AddItem( CartItem("Tesla", 120000) )

for i in cart:
   # use i...
```

Real things are built with dynamic languages



Episode #4: Enterprise Python and Large-Scale Projects
http://www.talkpythontome.com/episodes/show/4

A quick demo of that weirdo Python thing

NoSQL

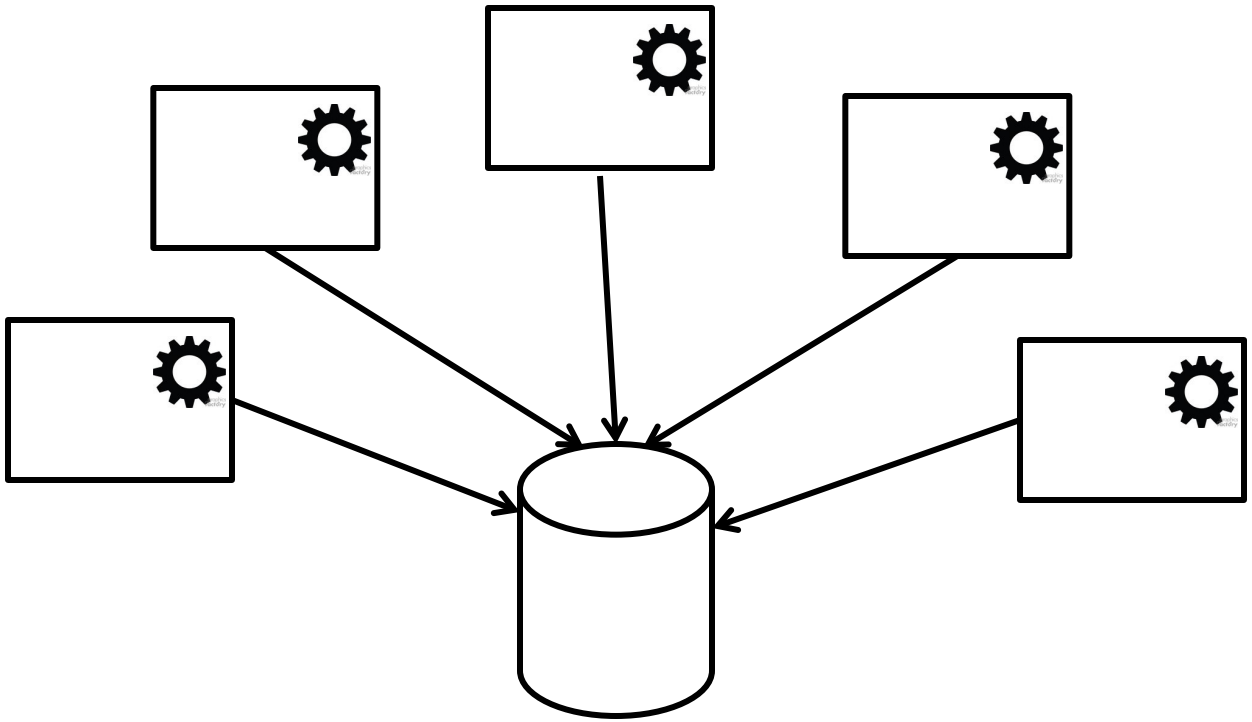
NoSQL: What is NoSQL?

Michael's NoSQL definition:

> "Database systems which are cluster-friendly and which trade inter-entity relationships for simplicity and performance."

History: Databases have allowed little flexibility

The integration database:



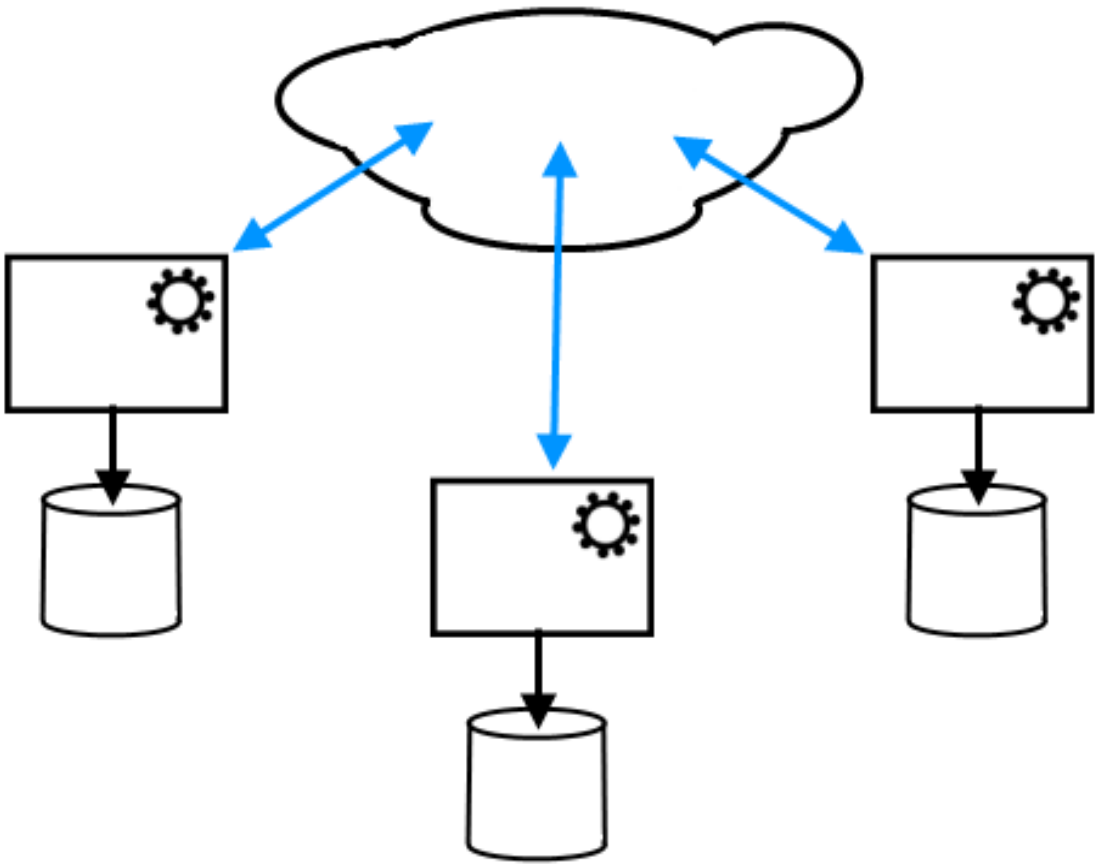
History: Application database

- Many large IT groups are moving away from integration databases towards application databases (1) + services and SOA.
  - services (SOA) provide another way to design systems:
  - you access data through the service layer
  - application databases provide data for a single service
- Very recently **microservices** seem to be catching on
- However:

Application DB !=> NoSQL

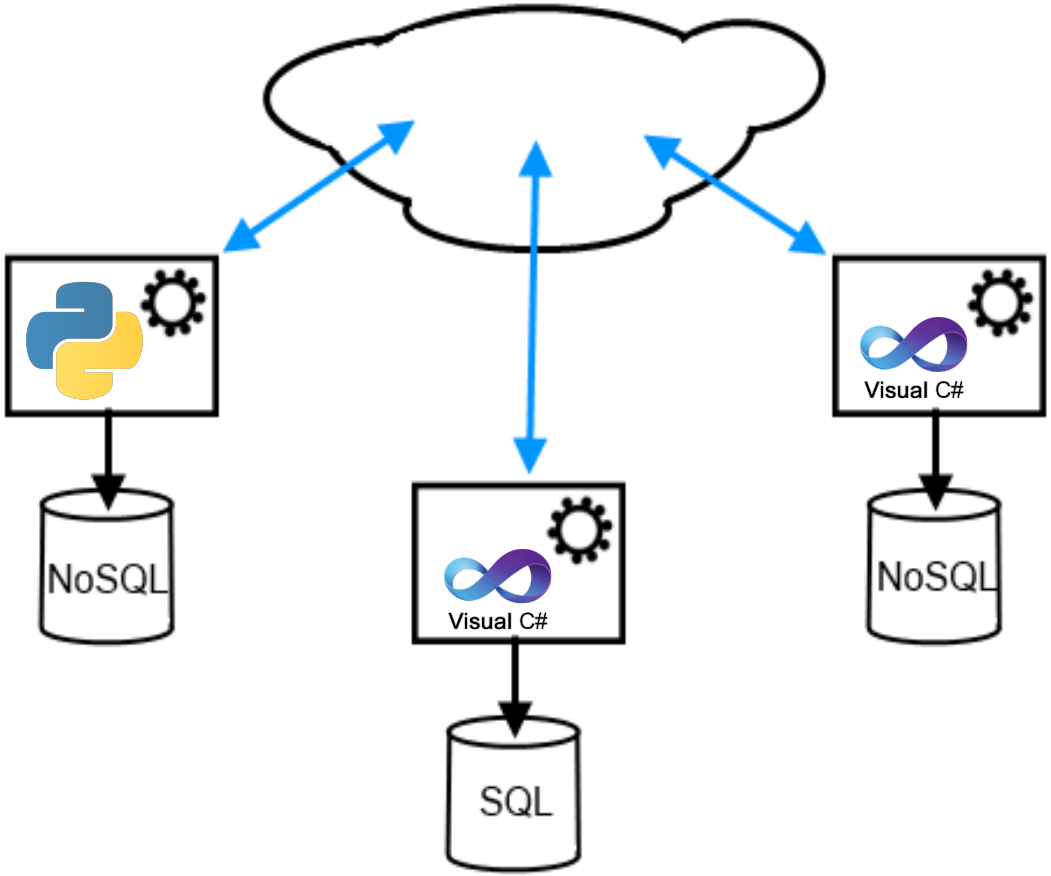
History: Industry is moving away from integration DBs

- The application database + services solution:



History: Polyglot persistence

Polyglot persistence and application databases:



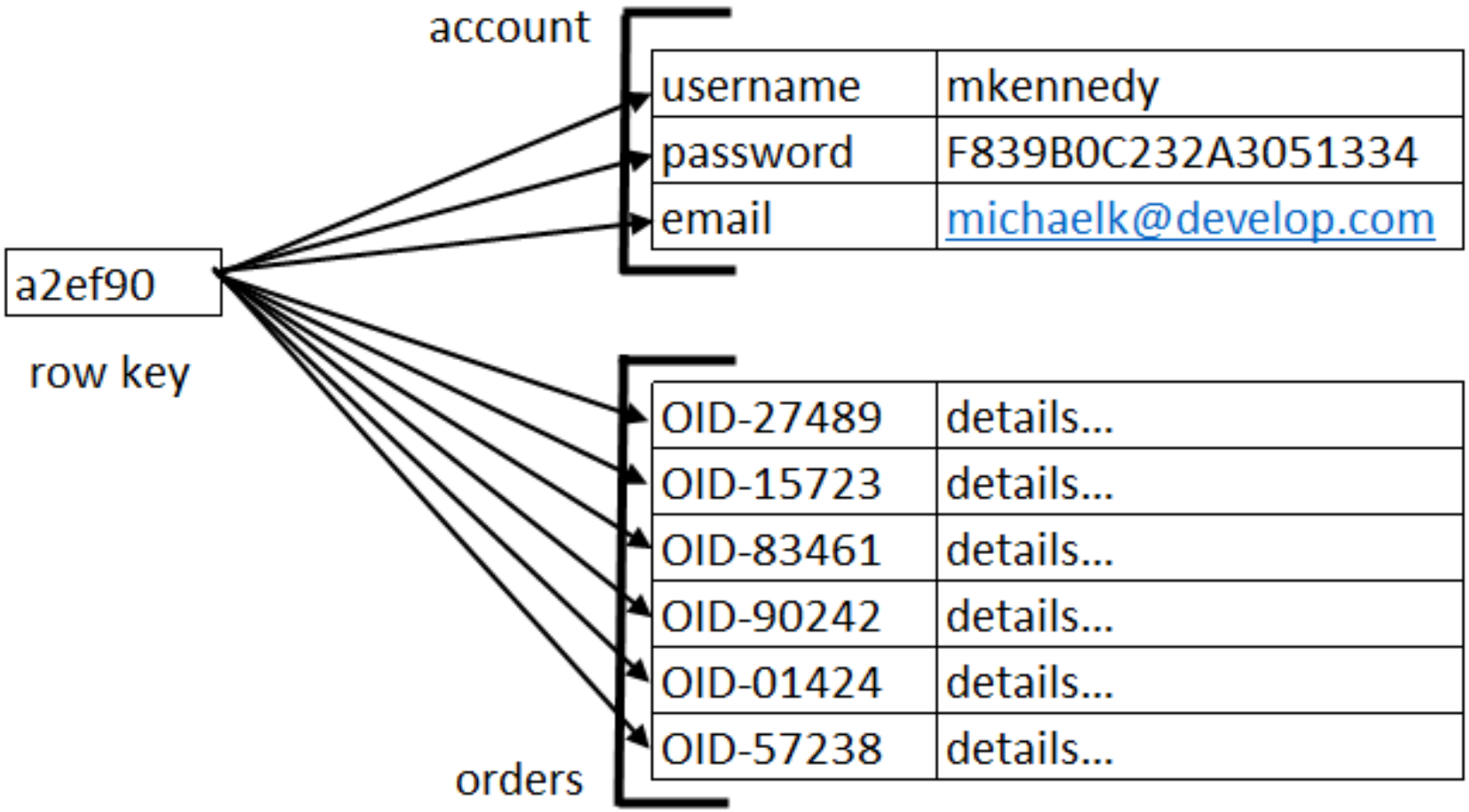
NoSQL: The 4 types of NoSQL databases

- **Key-value stores**:
  - Amazon DynamoDB
  - Riak
  - Memcached
- **Column-oriented databases**:
  - Hbase
  - Cassandra
  - Amazon SimpleDB
- **Graph databases**:
  - FlockDB
  - Neo4J
- **Document databases**:
  - MongoDB
  - CouchDB
  - RavenDB

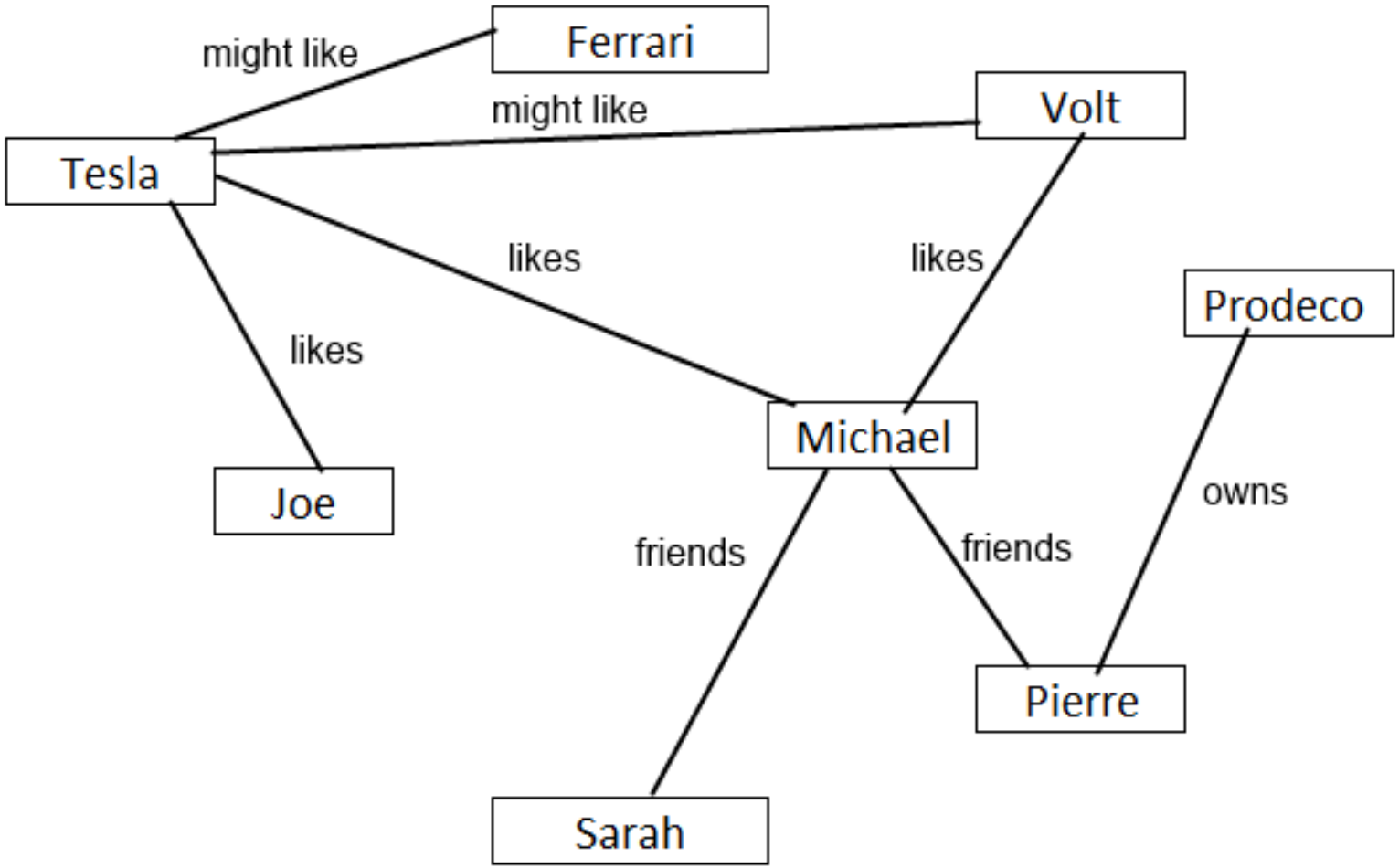
NoSQL: Key-value stores - how do they store data?

key	value
1	data-blob
2	data-blob
3	data-blob
4	data-blob
5	data-blob
...	...

NoSQL: Column-oriented DBs - how do they store data?



NoSQL: Graph DBs - how do they store data?



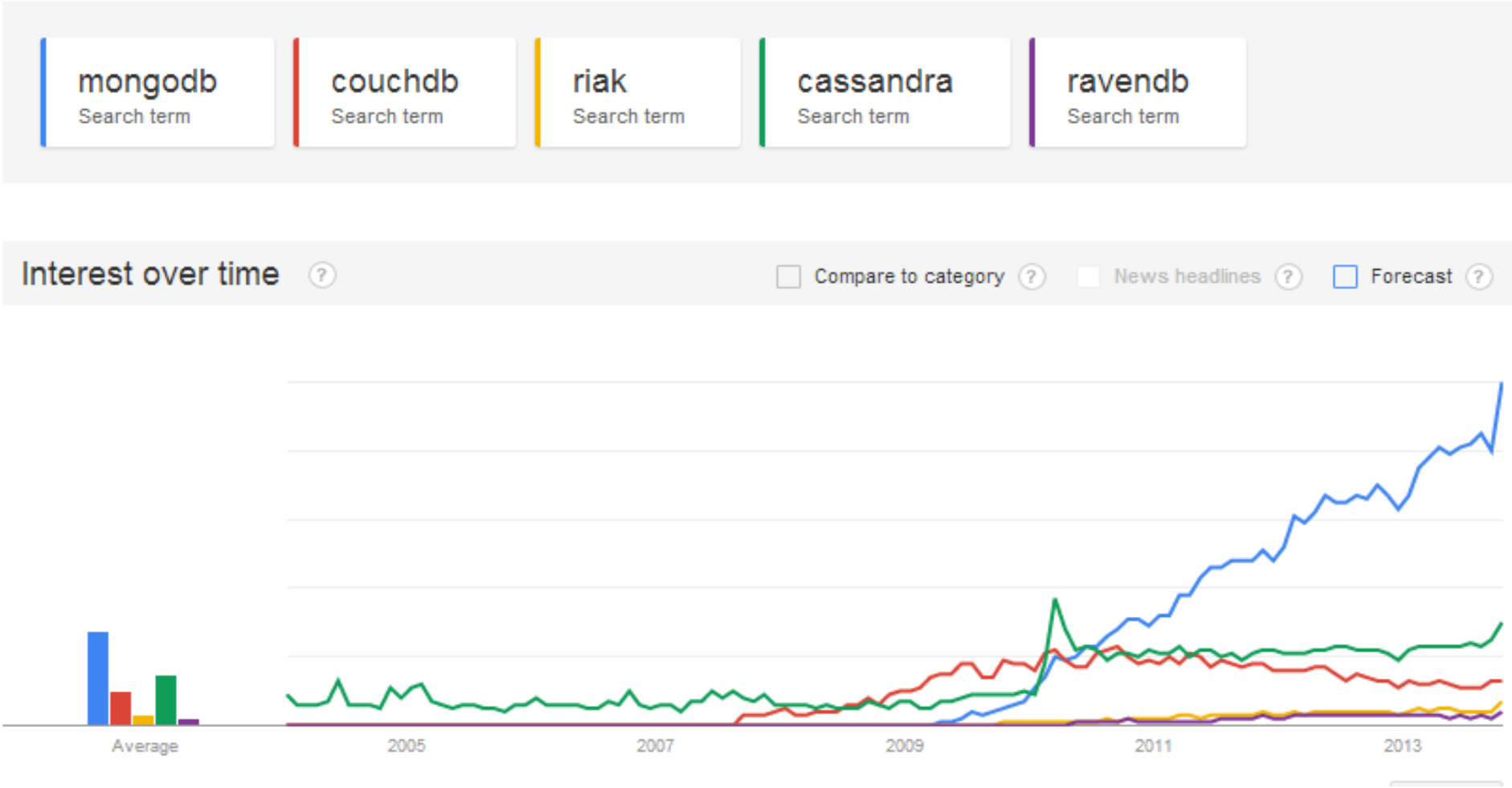
NoSQL: Document DBs - how do they store data?

```
{
    "_id" : ObjectId("524ca37bd588bf0e4c1ff713"),
    "Name" : "Intensive C++ Training",
    "ActiveCourse" : true,
    "NewCourse" : false,
    "CourseHighlights" : "...",
    "Prerequisites" : "...",
    "Engagements" : [
        {
            "_id" : ObjectId("524ca37bd588bf0e4c1ff714"),
            "CourseId" : ObjectId("524ca37bd588bf0e4c1ff713"),
            "StartDate" : ISODate("2010-03-15T07:00:00Z"),
            "..." : "..."
        },
        {
            "_id" : ObjectId("524ca37bd588bf0e4c1ff715"),
            "CourseId" : ObjectId("524ca37bd588bf0e4c1ff713"),
            "StartDate" : ISODate("2011-04-11T07:00:00Z"),
            "..." : "..."
        }
    ],
    "CourseAliases" : [],
    "UrlPath" : "intensive-c++-training"
}
```

Choosing a document database: MongoDB

http://www.mongodb.org

MongoDB: Why MongoDB



http://www.google.com/trends/explore#q=mongodb%2C%20couchdb%2C%20%2Fm%2F04f32m3%2C%20ravendb%2C%20raik&cmpt=q

Using MongoDB from C#

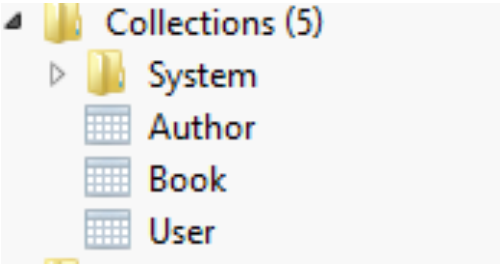
Implicit schema

- MongoDB does **not** maintain an **explicit** schema
- Your application can enforce an **implicit** schema
- C# classes are the natural way to define an implicit schema

C# classes and collections

- In most cases, MongoDB collections are mapped to one C# class
  - can aggregate other classes
  - supports inheritance (see later)
- Default serialization is what you want in most cases
  - all public read / write properties are serialized
  - out of the box support for Array, List, Dictionary
- Each top level class **must** define an **Id** property

```
class Author { }
class Book   { }
class User   { }
```



Example

```
public class Book // Document root
{
 public ObjectId Id { get; set; }
 public string Title{ get; set; }
 public string ISBN { get; set; }
 public string AuthorName { get; set; }
 public List<BookReview> Reviews { get; set; }
 private bool PopularToday;
}
```

```
public class BookReview // Never root of document, no need for ID
{
 public string ReviewerName { get; set; }
 public string Comment { get; set; }
 public int Stars { get; set; }
 public string StarsLabel {
     get { return string.Join("",Enumerable.Repeat("*", Stars)); } }
}
```

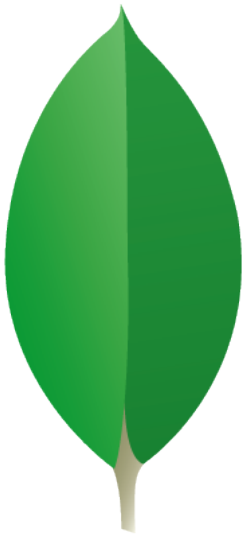
Example- 2

```
public class Book
{
…}
```

```
public class BookReview
{
…}
```

```
{
"_id" : ObjectId("5256dc0b0f6ccc2594c2adf8"),
    "ISBN" : "123-345-456",
    "AuthorName" : "J K Rowling",
    "Reviews" : [
        {
            "ReviewerName" : "Alice",
            "Comment" : "Great book",
            "Stars" : 5
        },
        {
            "ReviewerName" : "Bob",
            "Comment" : "Awful",
            "Stars" : 2
        }
    ],
    "Title" : "Harry Potter"
}
```

MongoDB + C# Demo



+

Basic Blog Web App

Using MongoDB from Python

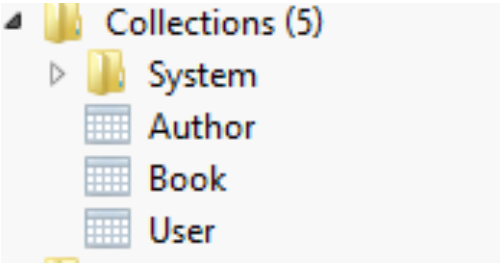
PyMongo

- MongoDB has an official Python driver
  - **pymongo**: https://pypi.python.org/pypi/pymongo
- Tutorials and documentation from MongoDB
  - Python Language Center http://docs.mongodb.org/ecosystem/drivers/python/
- Open-source on Github
  - https://github.com/mongodb/mongo-python-driver
- Supports
  - Python 3 and Python 2
  - Windows, OS X, Linux
- Installing pymongo from the installers on PyPI is preferred as they include the C-extensions

Python dictionaries and collections

- In most cases, MongoDB collections are mapped to Python dictionaries
  - can be object graphs (dictionaries, lists, fundamental types, and combinations thereof)
- Each top level dictionary **must** define an **_id** property

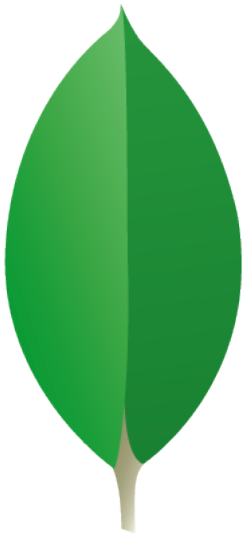
```
book = {
    '_id': ObjectId(),
    'title': 'Harry Potter',
    'isbn': '98382784792',
    'reviews' = [
       {'...': ...},
       {'...': ...}, ...
    ]
}
```



Example- 2

```
{
    "_id" : ObjectId("5256dc0b0f6ccc2594c2adf8"),
    "isbn" : "123-345-456",
    "author_name" : "J K Rowling",
    "reviews" : [
        {
            "reviewer_name" : "Alice",
            "comment" : "Great book",
            "stars" : 5
        },
        {
            "reviewer _name" : "Bob",
            "comment" : "Awful",
            "stars" : 2
        }
    ],
    "title" : "Harry Potter"
}
```

MongoDB + Python + PyMongo Demo



+

Basic Blog Web App

What about objects and classes?

We can use a Python ODM

ODM Choices in Python

There are many ODMs (Object Data Mappers) for MongoDB

- MongoEngine 

  - https://github.com/MongoEngine/mongoengine
- MotorEngine 

  - https://github.com/heynemann/motorengine
- MongoKit 

  - Python 2 only
  - https://github.com/namlook/mongokit

ODM Choices in Python (2)

There are many ODMs (Object Data Mappers) for MongoDB

- Ming [source forge]
  - http://merciless.sourceforge.net/tour.html
- Humongolus Watch 3 Star 55 Fork 13
  - https://github.com/entone/Humongolus
- MongoAlchemy Watch 12 Star 108 Fork 28
  - http://github.com/jeffjenkins/MongoAlchemy
- Minimongo Watch 12 Star 201 Fork 37
  - https://github.com/slacy/minimongo

MongoEngine is open-source

This repository Search | Explore | Gist | Blog | Help | mikeckennedy

MongoEngine / mongoengine

Watch 83 | Unstar 729 | Fork 502

A Python Object-Document-Mapper for working with MongoDB http://mongoengine.org

2,138 commits | 7 branches | 47 releases | 149 contributors

branch: master | mongoengine / +

Merge pull request #750 from foxx/patch-1 ...

yograterol authored 11 days ago — latest commit aa28abd517

docs	Fix multiple connections aliases being rewritten	13 days ago
mongoengine	Fix multiple connections aliases being rewritten	13 days ago
tests	Fix multiple connections aliases being rewritten	13 days ago
.gitignore	Fixed db_field load error	2 years ago
.travis.yml	Update .travis.yml for allow failures in pypy3	27 days ago
AUTHORS	Added myself to AUTHORS	12 days ago
CONTRIBUTING.rst	Updates to CONTRIBUTING.rst	a year ago
LICENSE	Update LICENSE	2 years ago
MANIFEST.in	Bump to v0.3	5 years ago
README.rst	Added landscape.io badge.	a month ago

Code | Issues 169 | Pull Requests 8 | Pulse | Graphs

HTTPS clone URL

https://github.com/MongoE

You can clone with HTTPS, SSH, or Subversion.

Clone in Desktop

Download ZIP

Example

```
class Book(Document): # Document root
    title = StringField(required=True)
    isbn = StringField(required=True)
    author_name = StringField()
    popular_today = BooleanField()
    reviews = ListField(EmbeddedDocumentField(BookReview))

    meta = dict(collection='Book')
```

```
# Never root doc, use EmbeddedDocument
class BookReview(EmbeddedDocument):
    reviewer_name = StringField()
    comment = StringField()
    stars = IntField()

    @property # not save to the DB
    def stars_label(self):
        star_text = ''
        for n in range(0, self.stars):
            star_text += '*'

        return star_text
```

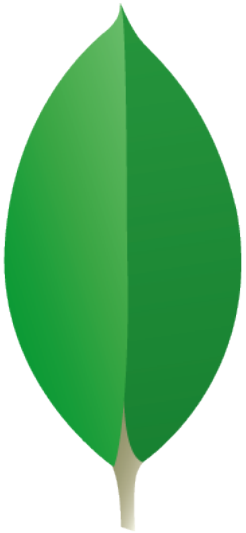
Example- 2

```
class Book:
    ...
```

```
class BookReview:
    ...
```

```
{
    "_id" : ObjectId("5256dc0b0f6ccc2594c2adf8"),
    "isbn" : "123-345-456",
    "author_name" : "J K Rowling",
    "reviews" : [
        {
            "reviewer_name" : "Alice",
            "comment" : "Great book",
            "stars" : 5
        },
        {
            "reviewer_name" : "Bob",
            "comment" : "Awful",
            "stars" : 2
        }
    ],
    "title" : "Harry Potter"
}
```

MongoDB + Python + MongoEngine Demo



+

+

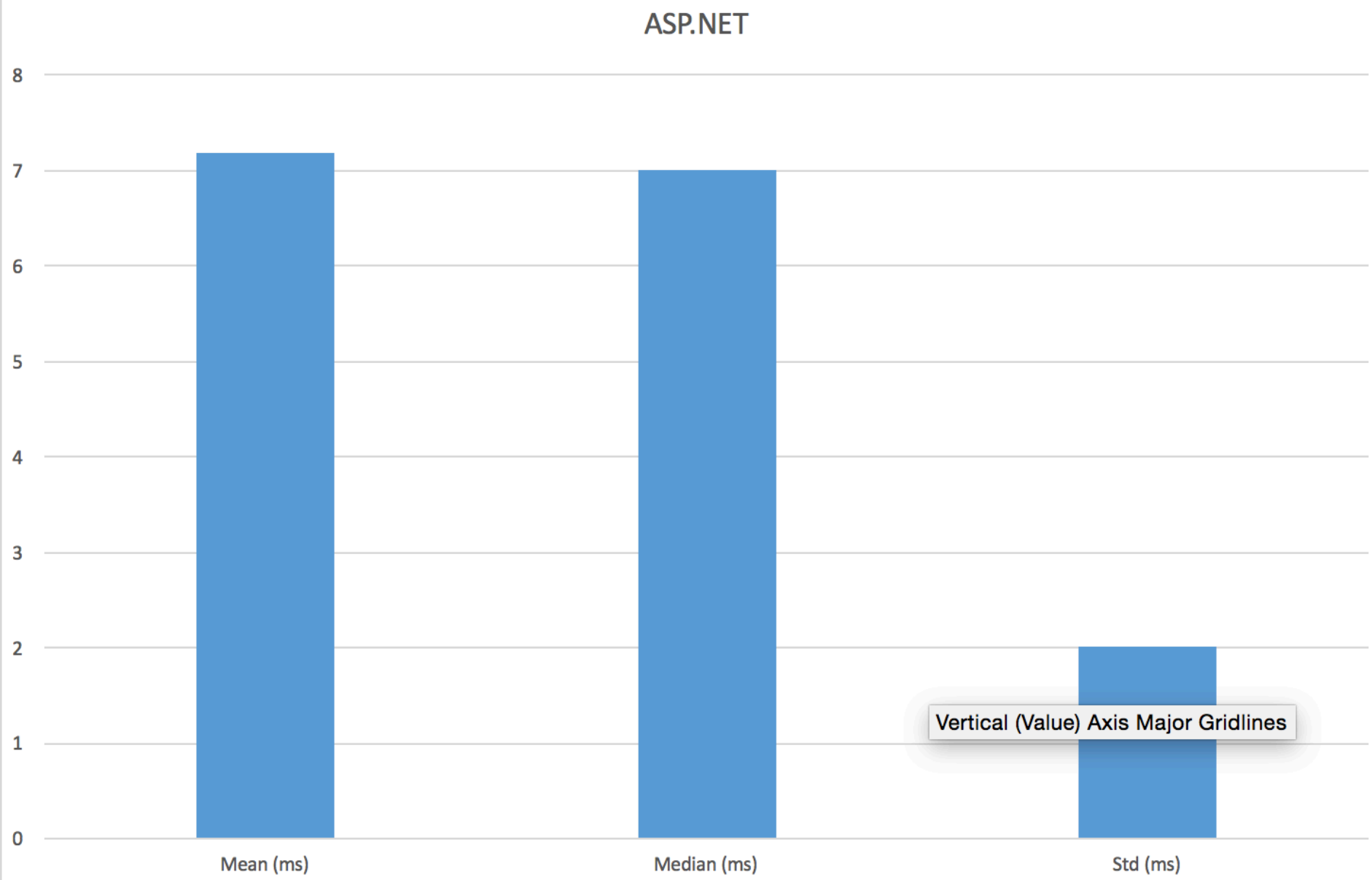
mongo engine

Basic Blog Web App

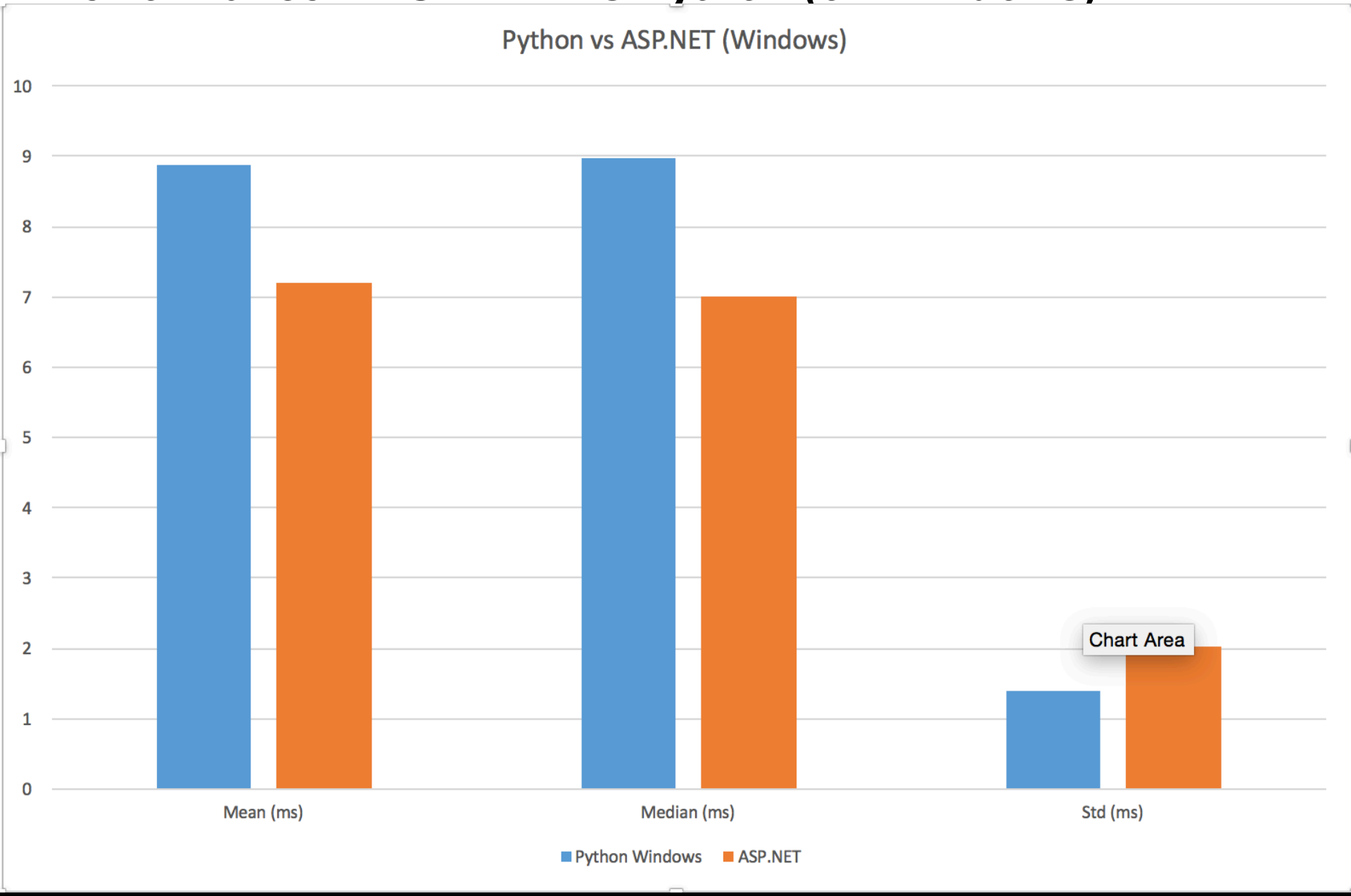
Performance

Is there a significant performance difference?

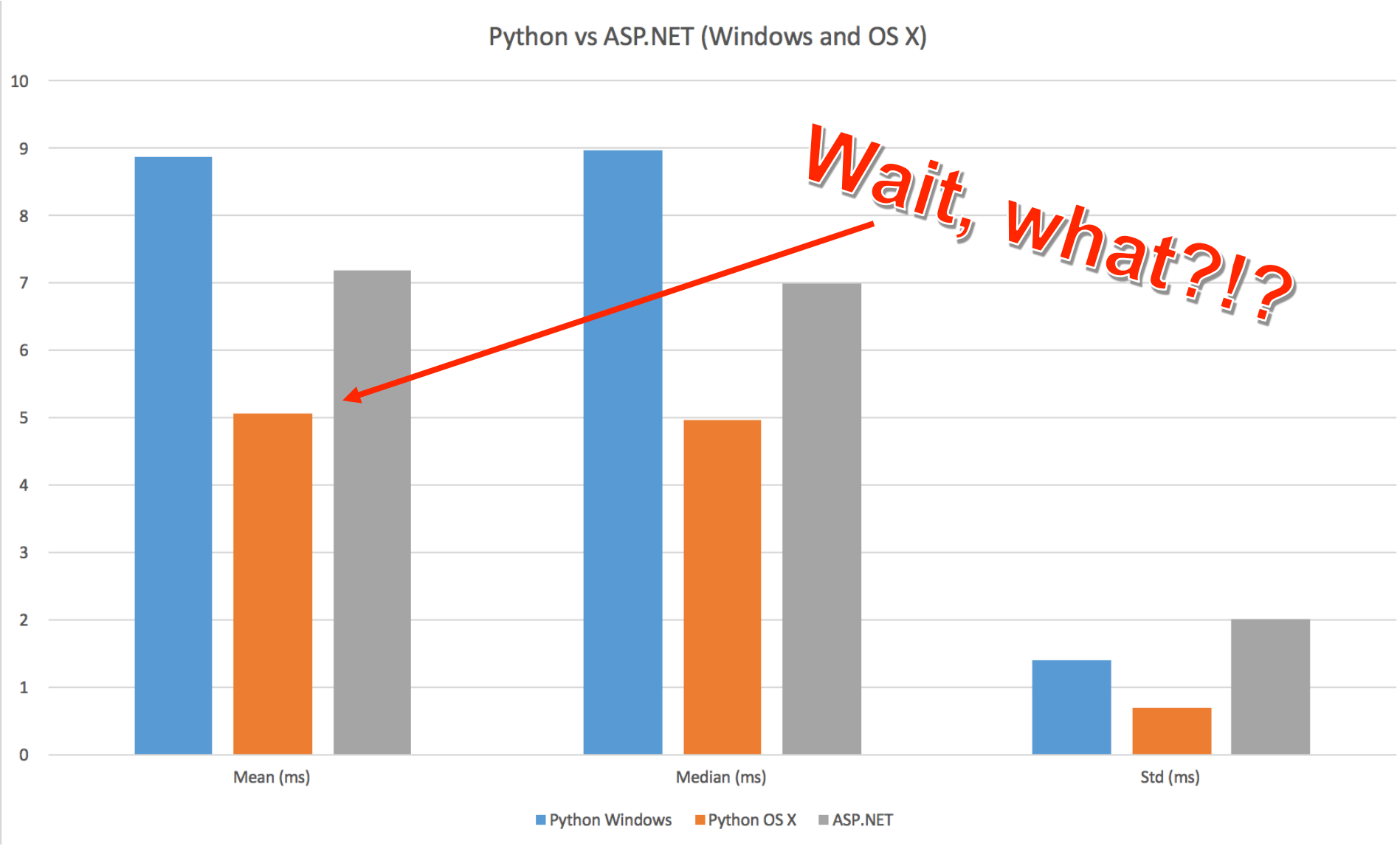
Performance – ASP.NET, 1000 serial requests



Performance – ASP.NET vs Python (on Windows)

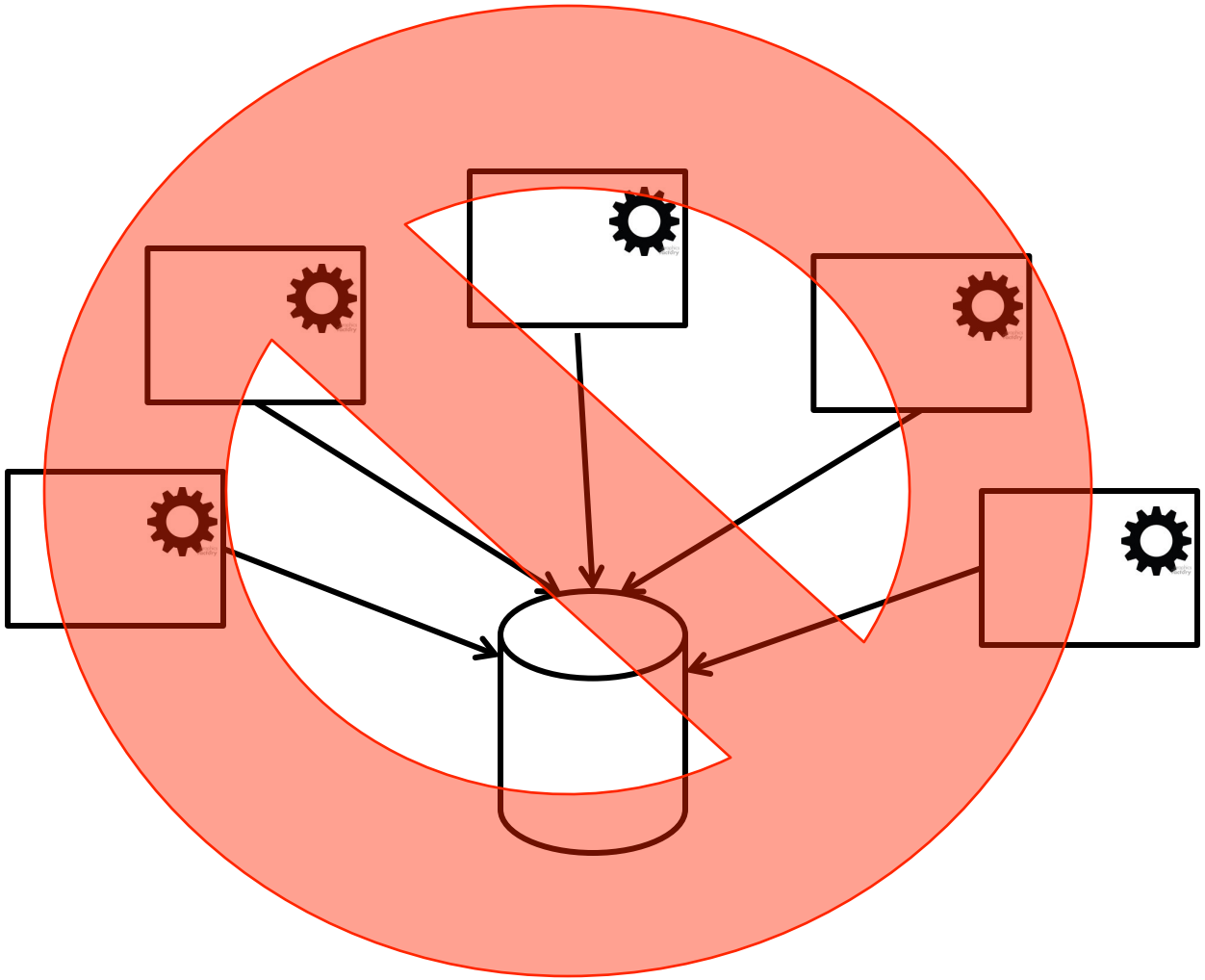


Performance – ASP.NET vs Python (Windows) vs Python (OS X)

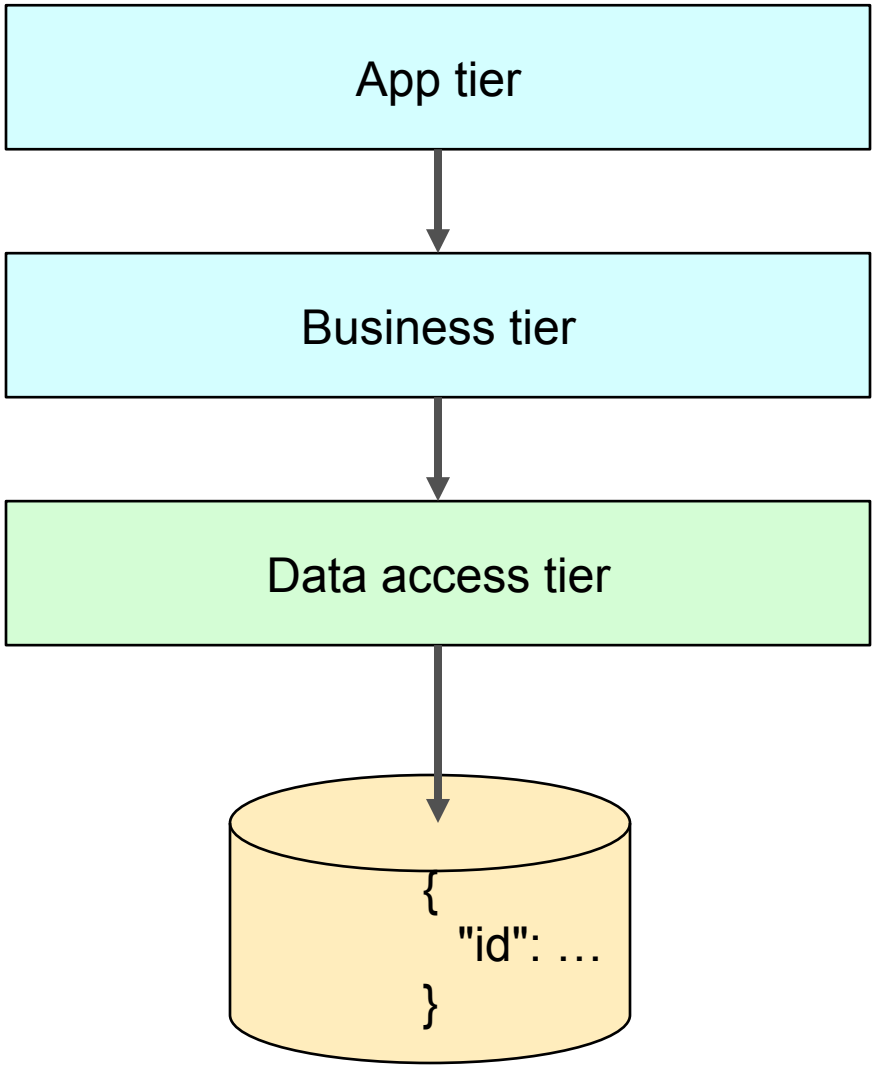


Guidance

Integration databases

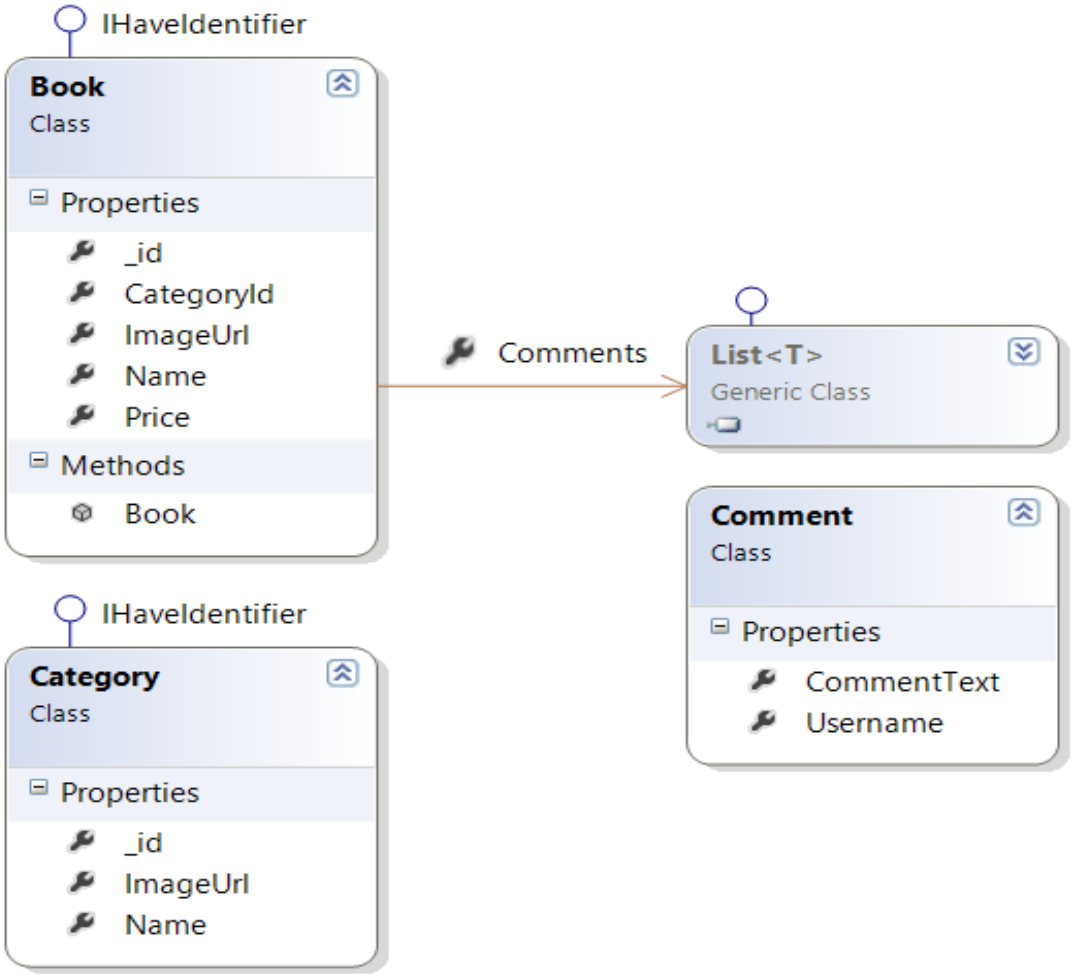


Use a proper data access layer



Dynamic languages

Use classes to provider structure



Don't forget the indexes!

The single biggest performance consideration is whether your database has indexes. Make sure it does!

Summary

- Document DBs provide the best general case NoSQL DB
- MongoDB is far and away the most popular document DB
- Your Mongo context is your implicit schema
- Use strongly typed collection to respect the schema
- Mongo + LINQ == heaven
- Use Update instead of retrieve + save
- Prepare yourself for simpler development and faster applications

Thanks for coming!

STAY IN TOUCH

Blog: blog.michaelckennedy.net

Twitter: @mkennedy

Google+: http://bit.ly/kennedy-plus

GitHub: github.com/mikeckennedy