hello $Mongo

Introduction à MongoDB

http://www.croes.org/gerald/conf/php/lille/2011/helloMongo.pdf

Gérald Croës
http://croes.org/gerald/blog/
@geraldcroes

Cédric Derue
@cderue

Il est encore temps de quitter la salle...

- Pas de schéma
- Non transactionnel
- Pas de jointures
- Pas de contrainte d'intégrité
- Pas de SQL
- Peu d'intégration avec des outils d'entreprise
- Limité à 2Go <- *Version 32bits*

… ou alors de rester

- Facilité
- Rapidité
- Scalabilité
- Pas de limitation (64bits)
- Disponible sur la plupart des plateformes

Plan

- Intro : Persistance des données
- Partie 1 : Développement d'une application de gestion de conférence
- Partie 2 : Mongo & ODM (Doctrine)
- Partie 3 : Index

Persistance des données

Persistance des données

Pourquoi une base de données ?

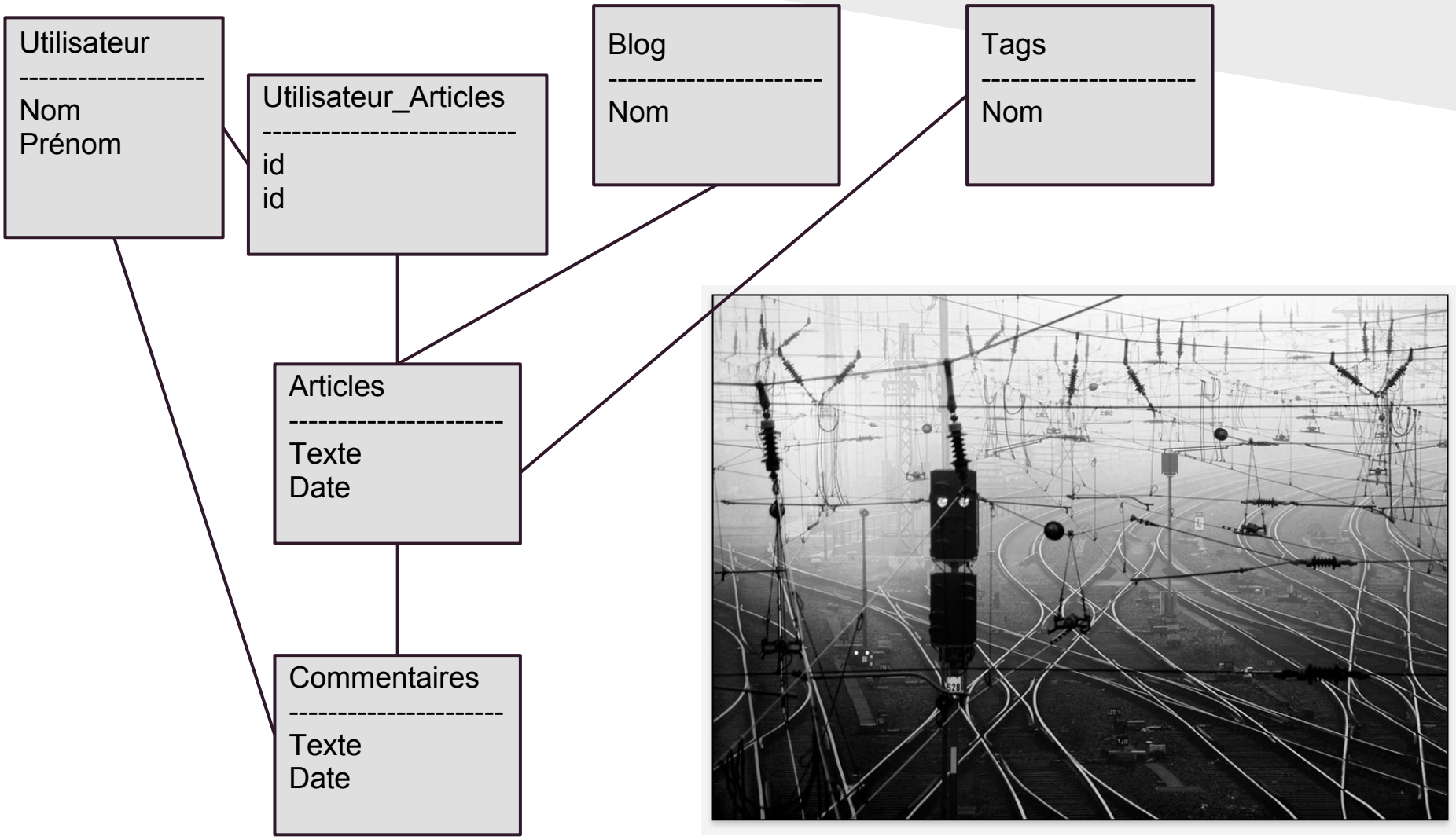
- **Lecture**
- **Ecriture**

- Concurrence d'accès
- Recherche efficace
- Interopérabilité

La réponse MongoDB

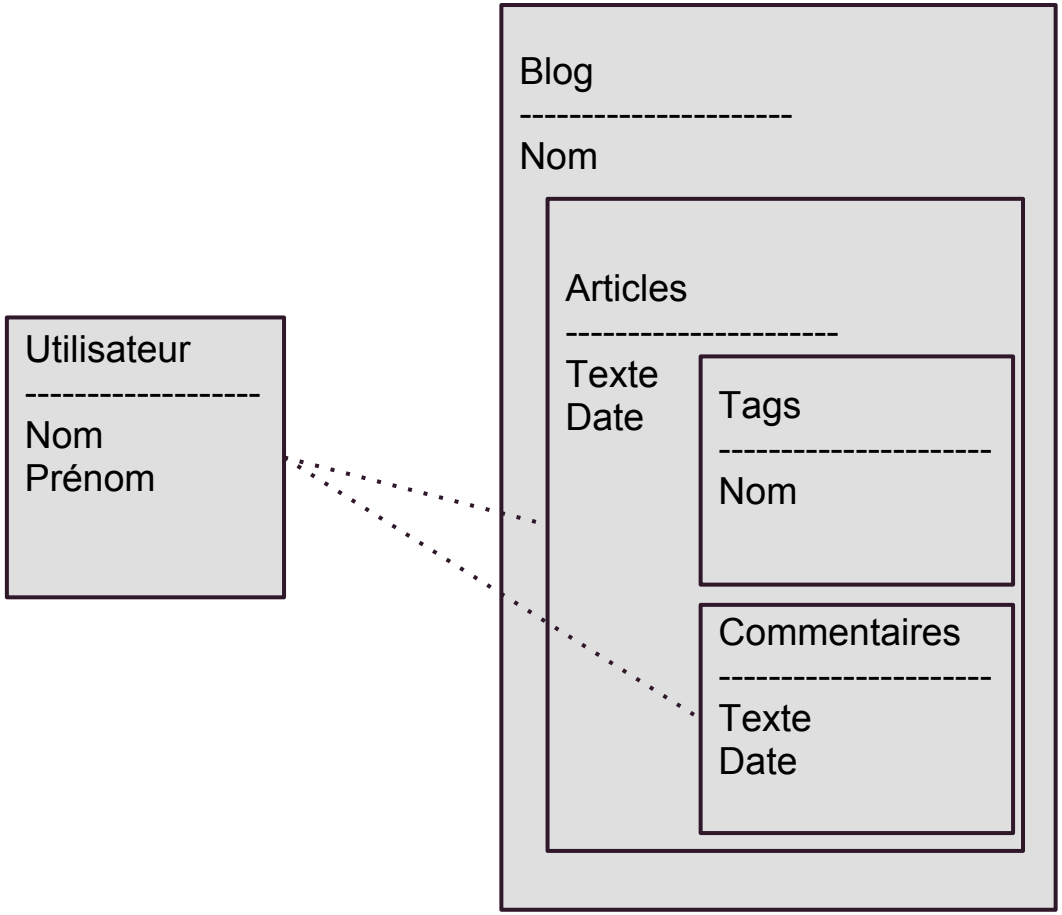
- Lecture / Ecriture des données très rapide
- Base de données orientée Document
- Format JSON / BSON
- Scalabilité horizontale

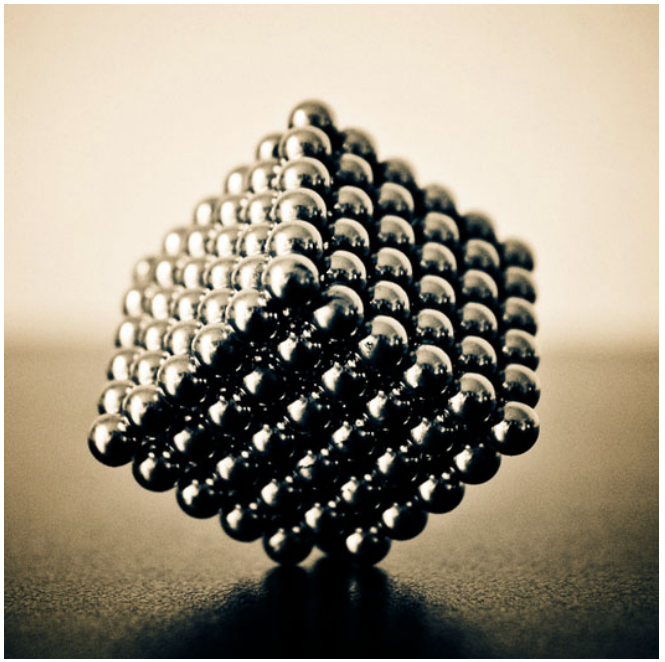
Le cas du blog relationnel…



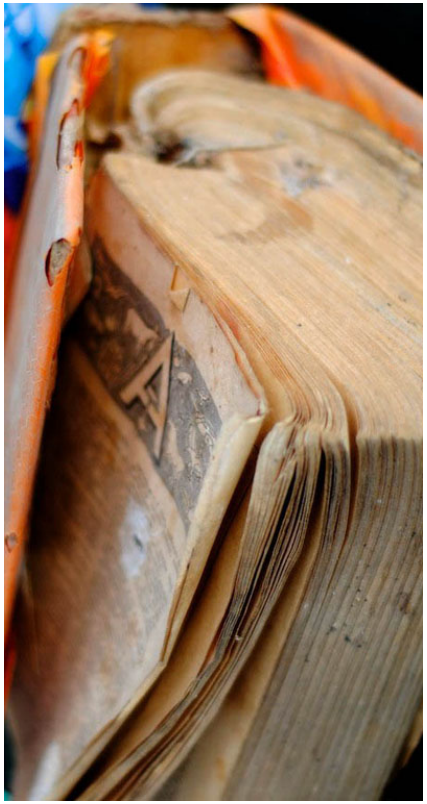
Le cas du blog {Documents}

Cohésion avec le modèle Objet





Petit lexique non illustré



Relationnel	MongoDB
Base de données	Base de données
Table	Collection
Index	Index
Enregistrement	Document
Colonne	Champ / Propriété
Clé étrangère	Référence
Jointures	Documents embarqués
Clé primaire	ObjectID
ORM	ODM

Installation de MongoDB

- Téléchargement

wget http://fastdl.mongodb.org/linux/mongodb-linux-x86_64-2.0.1.tgz

tar xzf mongodb.tgz

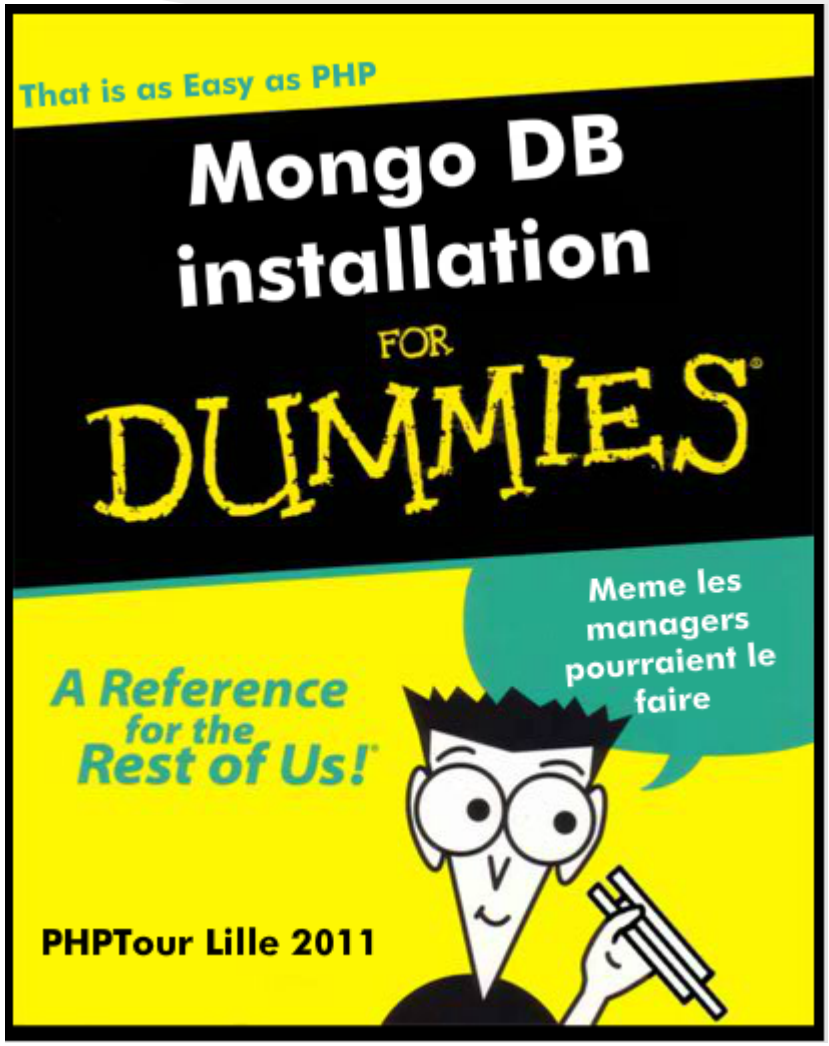
- Répertoire de données

sudo mkdir -p /data/db

sudo chown `id -u` /data/db

- Lancement

mongodb/bin/mongod



Installation du driver PHP

pecl install mongo

extension = mongo.so | php_mongo.dll

mongo

MongoDB Support	enabled
Version	1.2.6

Directive	Local Value	Master Value
mongo.allow_empty_keys	0	0
mongo.allow_persistent	1	1
mongo.auto_reconnect	1	1
mongo.chunk_size	262144	262144
mongo.cmd	$	$
mongo.default_host	localhost	localhost
mongo.default_port	27017	27017
mongo.long_as_object	0	0
mongo.native_long	0	0
mongo.no_id	0	0
mongo.utf8	1	1

Un premier document

```
use phptour

conference = {
    titre : "hello $Mongo;",
    lieu : "Salle de conférence 1"
}

db.evenements.save(conference)
```

Un premier document - recherche

```
db.evenements.find()

{
    "_id" : ObjectId("xxxxxxxxxxxx"),
    "titre" : "hello $Mongo;",
    "lieu" : "Salle de conférence 1"
}
```

Un identifiant _id généré automatiquement

Qu'avons nous donc ?

```
show dbs
local (empty)
phptour
test (empty)

show collections
evenements
system.indexes
```

Premiers constats

- Bases de données dynamiques
- Collections dynamiques
- Enregistrements ajoutés sans contrainte
- JSON

Pas de schéma préalable

JSon ?

JSON = JavaScript Object Notation

```
{
    'langage' : {
      'nom' : 'json',
      'points_forts' : ['lisible',
                        'souple',
                        'intégré',
                        'compact']
  }
}
```

```
<?php
echo json_encode(
        array(
        'language'=>array(
                        'nom' => 'json',
                        'points_forts' => array(
                                        'lisible',
                                        'souple',
                                        'intégré',
                                        'compact'
                                )
                        )
        )
);

//json_decode
```

BSon - Binary JSon

- Une surcouche de JSon
- Plus rapide à lire / écrire
  - Ecriture des entiers tels que
  - Informations supplémentaires de taille

```
{"hello": "world"}  →  "\x16\x00\x00\x00\x02hello\x00
                        \x06\x00\x00\x00world\x00\x00"
```

Retour au document - PHP

```
//connection à la base de données
$ct = new Mongo();

//Sélection de la base phptour
$db = $ct->phptour;

//Sélection de la collection conferences
$evenements = $db->evenements;

$conferenceMongo = array('titre' => 'hello $Mongo;',
                         'lieu' => 'Salle de conférence 1');
$evenements->insert($conferenceMongo);
```

Les objets Mongo (1/2)

- Mongo
  - Connexion à la base
  - close(), connect(), listDBs(), selectDB(), selectCollection()
- MongoDB
  - Base de données
  - createCollection(), selectCollection(), createDBRef(), getDBRef(), drop(), getGridFS()

Sélection des données - PHP

```
//Sélection du contenu de la base
foreach ($evenements->find() as $document) {
    echo "{$document['titre']} ({$document['_id']})\n\r";
}

//Sélection d'un seul élément
$document = $evenements
            ->findOne(array('_id'=>new MongoId
          ('xxxxx')));

//Sélection du seul titre de la conférence
$document = $evenements
            ->findOne(array('_id'=>new MongoId('xxxxx')),
                      array('titre'));
```

Les objets Mongo (2/2)

- MongoCollection
  - Collection
  - count(), find(), findOne(), insert(), remove(), save(), update()

- MongoCursor
  - Curseur sur résultat de requête
  - getNext(), count(), hint(), limit(), skip(), sort()

Modification du schéma

```
//Remplacement de tout le document
db.evenements.update(
   {titre : "hello $Mongo;"},
   {titre : "hello $Mongo;",
    lieu : "Salle de conférence 1",
    date : ISODate('2011-11-24 10:15:00')
   }
)

//Utilisation d'opérateurs
db.evenements.update(
  {titre : "hello $Mongo;"},
  {$set : {date : ISODate('2011-11-24 10:15:00')}}
)
```

Les opérateurs de modification

- $set - ajout d'une propriété
- $unset - suppression d'une propriété
- $inc - incrémenter la valeur d'une propriété numérique
- $rename - renommer une propriété

Modification du schéma - PHP

```
$ct = new Mongo();
$evenements = $ct->phptour->evenements;
$evenements->update(
   array('titre'=>'hello $Mongo;'),
   array('titre'=>'hello $Mongo;',
         'lieu'=>'Salle de conférence 1',
         'date'=>new MongoDate(strtotime('2011-11-24 10:
15:00'))));

//Utilisation d'opérateurs
$evenements->update(
    array('titre'=>'hello $Mongo;'),
    array('$set'=>array('date'=>new MongoDate(strtotime
('2011-11-24 10:15:30'))))
);
```

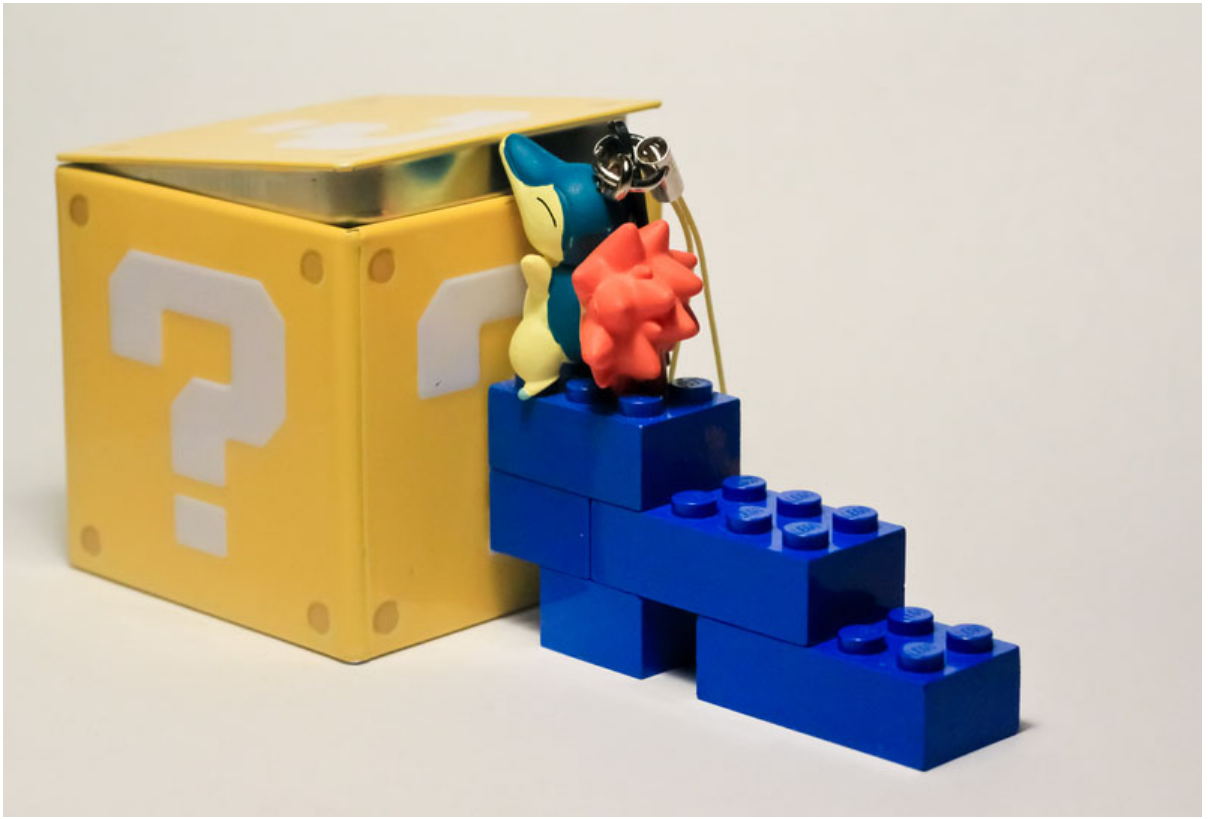
Quelques données supplémentaires

```
$keynote = array('titre' => "Keynote d'ouverture",
  'lieu' => 'Auditorium', 'date'=>new MongoDate(strtotime('2011-11-24 9:
00:00')),
  'type'=>'Keynote',      'minutes'=>30);
$repas = array('titre' => "Repas",
  'date'=>new MongoDate(strtotime('2011-11-24 12:45:00')),
  'type'=>'Pause', 'minutes'=>60);
$pauseMatin = array('titre' => "Pause",
  'date'=>new MongoDate(strtotime('2011-11-24 11:00:00')),
  'type'=>'Pause', 'minutes'=>15);
$pauseAM = array('titre' => "Pause",
  'date'=>new MongoDate(strtotime('2011-11-24 15:30:00')),
  'type'=>'Pause', 'minutes'=>15);

$evenements->insert($keynote);
$evenements->insert($repas);
$evenements->insert($pauseMatin);
$evenements->insert($pauseAM);
```

Deuxième constat

Données qui n'ont pas la même structure

- type
- lieu
- minutes



Requêtes type

```
> db.evenements.distinct("type");
["Keynote", "Pause"]

> db.evenements.distinct("minutes", {type : "Pause"})
[60, 15]

> db.evenements.find({type : "Pause"}).count()
3

> db.evenements.group({
... key : { minutes : true},
... cond : { type : "Pause" },
... reduce : function (obj, prev) { prev.nb += 1; },
... initial : {nb : 0}
... });
[ { "minutes" : 60, "nb" : 1 }, { "minutes" : 15, "nb" : 2 } ]
```

Requêtes type - PHP

```
$ct->phptour->command(array('distinct'=>'evenements', 'key'=>'type');

$ct->phptour->command(array('distinct'=>'evenements', 'key'=>'minutes',
'query'=>array('type' => 'Pause'));

$evenements->find(array('type'=>'Pause'))->count();

$evenements->group(
                    array('minutes'=>1),
                    array('nb'=>0),
                    'function (obj, prev) { prev.nb += 1; }',
                    array('type'=>'Pause')
                   );
```

Opérateurs de requête (1/2)

$lt	<
$lte	<=
$gt	>
$gte	>=

```
> db.evenements.find({minutes : {$gte : 30}})

{ "_id" : ObjectId("4ebf87db945630cc0a000000"), "titre" : "Keynote
d'ouverture", "lieu" : "Auditorium", "date" : ISODate("2011-11-24T08:00:
00Z"), "type" : "Keynote", "minutes" : 30 }
{ "_id" : ObjectId("4ebf87db945630cc0a000001"), "titre" : "Repas", "date" :
ISODate("2011-11-24T11:45:00Z"), "type" : "Pause", "minutes" : 60 }
```

Opérateurs de requête (2/2)

$exists
$or
$nor
$and
$mod
$ne
$in
$nin
$all

Exemples opérateurs

```
//PHP
$evenements->update(
   array('type'=>array('$exists'=>false)),
   array('$set'=>array('type'=>'Conference', 'minutes'=>45))
);

//Mongo
> db.evenements.find({minutes : {$in : [30, 15]}})
{ "_id" : ObjectId("4ebf87db945630cc0a000000"), "titre" : "Keynote
d'ouverture", "lieu" : "Auditorium", "date" : ISODate("2011-11-24T08:00:
00Z"), "type" : "Keynote", "minutes" : 30 }
{ "_id" : ObjectId("4ebf87db945630cc0a000002"), "titre" : "Pause", "date"
: ISODate("2011-11-24T10:00:00Z"), "type" : "Pause", "minutes" : 15 }
{ "_id" : ObjectId("4ebf87db945630cc0a000003"), "titre" : "Pause", "date"
: ISODate("2011-11-24T14:30:00Z"), "type" : "Pause", "minutes" : 15 }
```

Propriétés multivaluées

- Nouvelle table
- Jointures
- Clefs étrangères
- Contraintes d'intégrité
- Modification des services

Ajout de tags

```
conference_atoum = {
    type : 'Conférence',
    titre : 'Atoum',
    date : ISODate('2011-11-24 15:45:00'),
    minutes : 45,
    tags : ['Industrialisation', 'Tests Unitaires', 'Framework']
}
db.evenements.insert(conference_atoum)
--- --- --- --- --- --- --- --- --- --- --- --- --- --- --- --- ---
---
$evenements->insert(
array('type'=>'Conférence',
      'titre'=>'Atoum, le framework de tests unitaires simple, ...',
      'date'=>strtotime('2011-11-24 15:45:00'),
      'minutes'=>45,
      'tags'=>array('Industrialisation', 'Tests Unitaires', 'Framework'))
);
```

Requêtes types sur tableaux

```
> db.evenements.distinct('tags')
[   "Framework", "Industrialisation", "Tests Unitaires", "Experience",
    "XML", "XQuery", "Optimisation", "XHRProf", "Marketing", "Asynchrone",
    "Mongrel2", "ZeroMQ", "Cloud", "CMS", "REST", "Drupal", "Varnish",
    "Optimisation,XDebug", "Securité", "", "Documentation", "VoiP",
    "Outils" ]

> db.evenements.find({tags : {$in : ['Optimisation']}}).count()
8

> db.evenements.find({tags : {$all : ['Optimisation', 'Varnish']}}).
count()
1

> db.evenements.find({tags : {$nin : ['Optimisation']}, type :
'Conférences'}).count()
23

> db.evenements.find({tags : {$size : 0}})
...
```

Les opérateurs de modification sur tableaux

- $push
- $pushAll
- $addToSet
- $pop
- $pull
- $pullAll

Requêtes type sur tableaux

```
> db.evenements.update({tags : {$in : [""]}}, {$pull : {tags : ""}})


> db.evenements.update({tags : {$in : ["Optimisation,XDebug"]}},
                       {$pushAll : {tags : ["Optimisation", "XDebug"]},
                        $pull : {tags : "Optimisation,XDebug"}})

---> Field name duplication not allowed with modifiers

> db.evenements.update({tags : {$in : ["Optimisation,XDebug"]}},
                       {$pushAll : {tags : ["Optimisation", "XDebug"]}})

> db.evenements.update({tags : {$in : ["Optimisation,XDebug"]}},
                        {$pull : {tags : "Optimisation,XDebug"}})
```

Requêtes type sur tableaux - PHP

```
$evenements->update(
   array('tags'=>array('$in'=>array('Optimisation,XDebug'))),
   array('$pushAll'=>array('tags'=>array('Optimisation', 'XDebug')))
);

$evenements->update(
   array('tags'=>array('$in'=>array('Optimisation,XDebug'))),
   array('$pull'=>array('tags'=>'Optimisation,XDebug'))
);

$evenements->update(
    array('tags'=>array('$in'=>array(""))),
    array('$pull'=>array('tags'=>""))
);
```

Map / Reduce

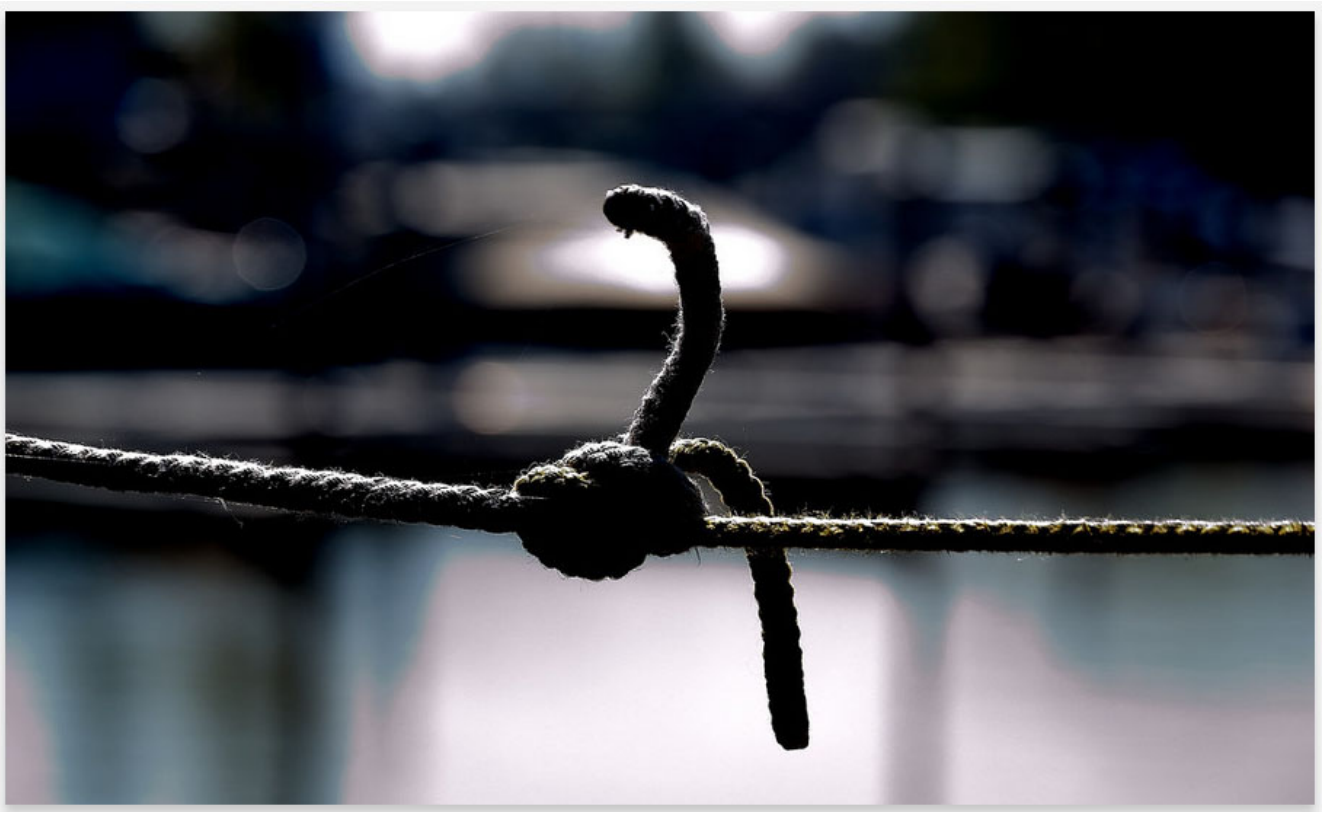
```
map = function() {
    if (!this.tags) {
        return;
    }

    for (index in this.tags) {
        emit(this.tags[index], 1);
    }
}


reduce = function(previous, current) {
    var count = 0;

    for (index in current) {
        count += current[index];
    }

    return count;
}
```

```
result = db.runCommand({
... "mapreduce" : "evenements",
... "map" : map,
... "reduce" : reduce,
... "out" : {inline : 1}})
```

```
{
"results" : [ {"_id" : "Asynchrone", "value" : 1},
              {"_id" : "Experience","value" : 5},
              {"_id" : "Framework","value" : 5},
              {"_id" : "Industrialisation","value" : 7},
              {"_id" : "Optimisation","value" : 10},
              {"_id" : "Outils","value" : 1},
              {"_id" : "Tests Unitaires","value" : 3},
              {"_id" : "Varnish","value" : 1},
              {"_id" : "VoiP","value" : 1},
              {"_id" : "XDebug","value" : 2},
              {"_id" : "XHRProf","value" : 1},
              {"_id" : "XML","value" : 1},
              {"_id" : "XQuery","value" : 1},
              {"_id" : "ZeroMQ","value" : 1}
       ],
       "timeMillis" : 64,
       "counts" : {"input" : 36,"emit" : 48,"reduce" : 6,"
output" : 22},
       "ok" : 1
}
```

Références

- L'équivalent de la clef étrangère
- MongoID / MongoDBRef



Requêtes type sur références

```
> geraldcroes = {nom : 'Croes', 'prenom' : 'Gerald', entreprise : 'Alptis'}
> db.conferenciers.save(geraldcroes)
> geraldcroes
{
    "nom" : "Croes",
    "prenom" : "Gerald",
    "entreprise" : "Alptis",
    "_id" : ObjectId("4ec933769f028924de1fb0df")
}

> conference_gerald = db.evenements.findOne({titre : 'hello $Mongo;'})
> conference_gerald.conferenciers = [new DBRef('conferenciers',
geraldcroes._id)]
> db.evenements.save(conference_gerald)
> db.evenements.find({conferenciers : {$in : [new DBRef('conferenciers',
geraldcroes._id)]}})
```

Requêtes type sur références - PHP

```
$cd = array('prenom'=>'Cedric', 'nom'=>'Derue', 'entreprise'=>'Alptis');
$conferenciers->save($cd);

$conferenceMongo = $evenements->findOne(array('titre'=>'hello $Mongo;'));
$conferenceMongo['conferenciers'][] = $conferenciers->createDBRef($cd);
$evenements->save($conferenceMongo);


$evenements->findOne(
    array('conferenciers'=>array(
        '$in'=>array($conferenciers->createDBRef($cd))))
    )
);


$evenements->find(array('conferenciers'=>array('$size'=>2)))
```

Documents embarqués

- Un niveau d'arborescence en plus

Requêtes type sur documents embarqués

```
> conferencemongo = db.evenements.findOne({titre : 'hello $Mongo;'})
> conferencemongo.plan =
... [
...   {titre : 'Introduction', slides : [1,2,3,4]},
...   {titre : 'Persistance des donnees', slides : [5,6,7]}
... ]
> db.evenements.save(conferencemongo)

//Quelle conférence dispose d'une partie intitulée introduction ?
> db.evenements.find({'plan.titre' : 'Introduction'})
...

> db.evenements.distinct('plan.titre')
['Introduction', 'Persistance des donnees']
```

Requêtes type sur documents embarqués - PHP

```
$plan = array(
    array('titre'=>'Introduction', 'slides'=>array(1,2,3,4)),
    array('titre'=>'Persistance des données', 'slides'=>array(5,6,7)),
    array('titre'=>'Premier document', 'slides'=>array(8,9))
);

$conferenceMongo = $evenements->findOne(array('titre'=>'hello $Mongo;'));
$conferenceMongo['plan'] = $plan;

$evenements->save($conferenceMongo);
```

GridFS - Stockage de fichiers

- 2 collections spéciales : files et chuncks

Ajout / Lecture de fichiers

```
//Ecriture d'un fichier
$grid = $ct->phptour->getGridFS();
$id = $grid->storeFile("conferences.csv", array('meta1'=>'Superbe',
'meta2'=>'Vraiment superbe'));


//Lecture du fichier
$conferences = $grid->findOne(array('_id'=>$id));
echo $conferences->getBytes();
```

Intégration dans une application PHP

Frameworks MVC

- Zend Framework 1 et 2
- Symfony
- CakePHP
- Lithium

CMS

- Joomla
- Drupal

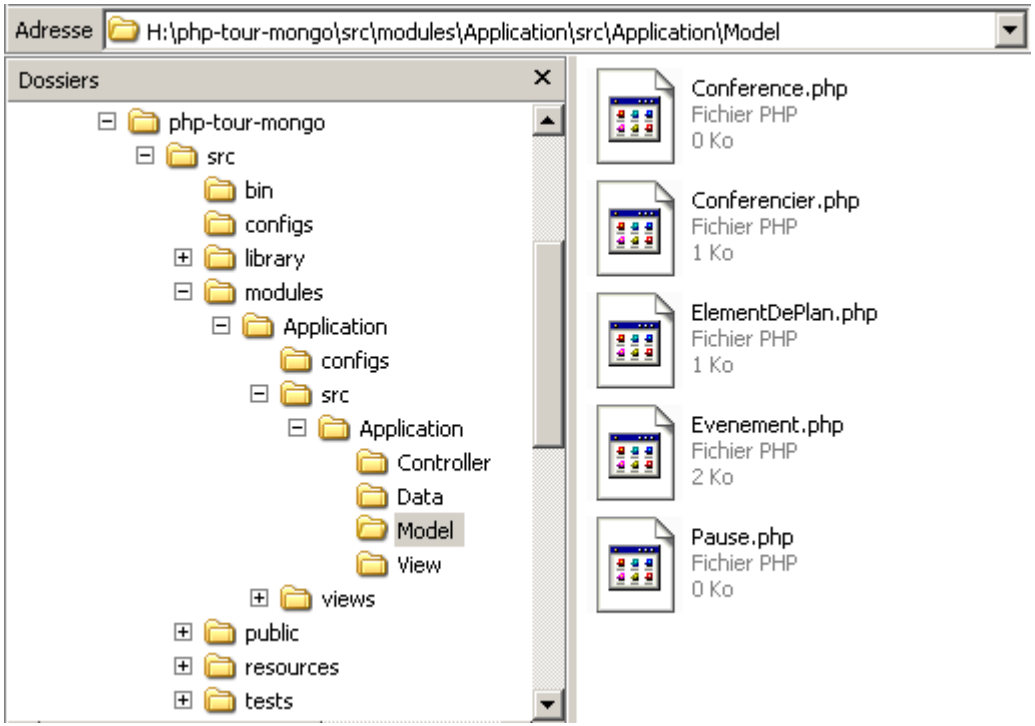
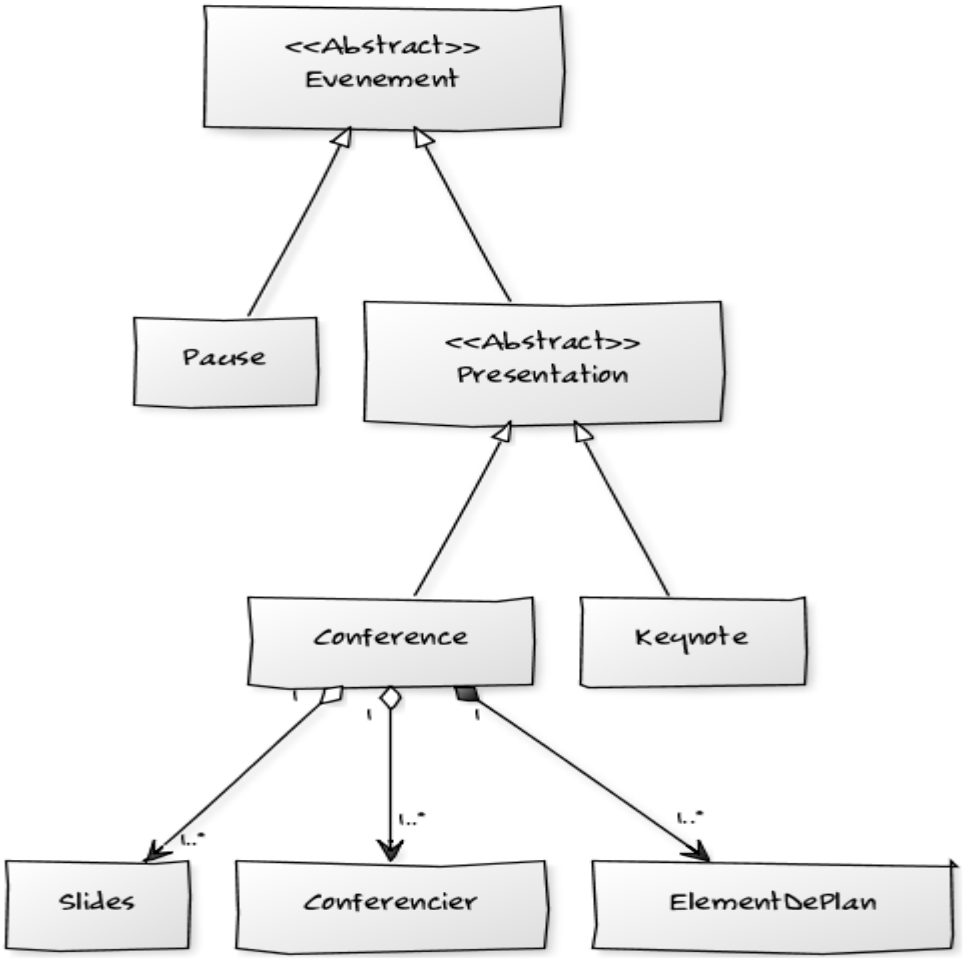


Diagramme de classes



Classes avec Doctrine ODM (1/7)

```
namespace PHPTour;

/** @Document */
class Conferencier
{
    /** @Id */
    private $id;

    /** @String */
    private $nom;

    /** @String */
    private $prenom;

    /** @String */
    private $entreprise;
}
```

Classes avec Doctrine ODM (2/7)

```
namespace PHPTour;

/**
 * @Document
 * @InheritanceType("SINGLE_COLLECTION")
 * @DiscriminatorField(fieldName="type")
 * @DiscriminatorMap({"PHPTour\Conference"="Conference","PHPTour\Pause"="
Pause"})
 */
abstract class Evenement
{
    /** @Id */
    protected $id;

    /** @String */
    protected $titre;

    /** @Date */
    protected $date;

    /** @Int */
    protected $minutes;
}
```

Classes avec Doctrine ODM (3/7)

```
namespace PHPTour;

/**
 * @Document
 */
abstract class Presentation extends Evenement
{
   /** @String */
   protected $lieu;

   /** @ReferenceMany(targetDocument="Conferencier")*/
   protected $conferenciers;
}
```

Classes avec Doctrine ODM (4/7)

```
namespace PHPTour;

/** @Document */
class Pause extends Evenement
{

}

/**
 * @Document
 */
class Keynote extends Presentation
{
}
```

Classes avec Doctrine ODM (5/7)

```
namespace PHPTour;

/** @Document */
class Slides
{
    /** @Id */
    private $_id;
    /** @File */
    private $_fichier;
    /** @Field */
    private $uploadDate;
    /** @Field */
    private $length;
    /** @Field */
    private $filename;
    /** @Field */
    private $md5;
}
```

Classes avec Doctrine ODM (6/7)

```
namespace PHPTour;

/**
 * @EmbeddedDocument
 */
class ElementDePlan
{
    /**
     * @String
     */
    protected $titre;

    /**
     * @Collection
     */
    protected $slides;
}
```

Classes avec Doctrine ODM (7/7)

```
namespace PHPTour;

/**
 * @Document
 */
class Conference extends Presentation
{
   /** @Collection */
   private $tags;

   /** @EmbedMany */
   protected $plan;

   /**@ReferenceOne(targetDocument="Slides") */
   protected $slides;
}
```

XML / YAML

```xml
<?xml version="1.0" encoding="UTF-8"?>
<doctrine-mongo-mapping ...>
    <document name="PHPTour\Conferencier">
        <field fieldName="id" id="true" />
        <field fieldName="nom" type="string" />
        <field fieldName="prenom" type="string" />
        <field fieldName="entreprise" type="string" />
    </document>
</doctrine-mongo-mapping>
```

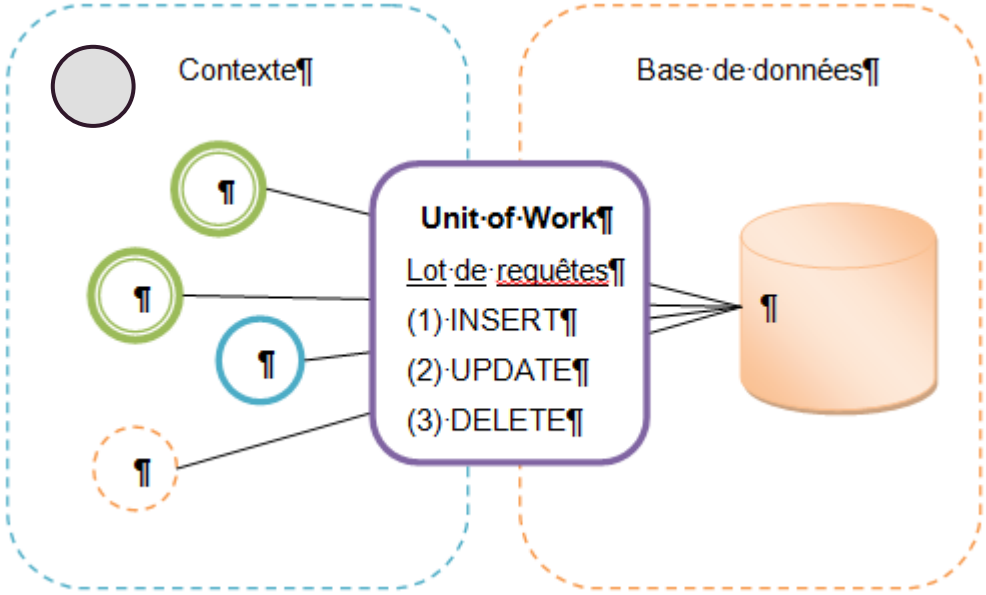
```yaml
PHPTour\Conferencier:
  fields:
    id:
      type: id
      id: true
    nom:
      type: string
    prenom:
      type: string
    entreprise:
      type: string
```

DocumentManager : initialisation

```
$config = new \Doctrine\ODM\MongoDB\Configuration();
$config->setProxyDir('path/to/geearate/proxies');
$config->setProxyNamespace('Proxies');
$config->setHydratorDir('path/to/generate/hydrators');
$config->setHydratorNamespace('Hydrators');


$reader = new \Doctrine\Common\Annotations\AnnotationReader();
$reader->setDefaultAnnotationNamespace('Doctrine\ODM\MongoDB\Mapping\\');
$config->setMetadataDriverImpl(new AnnotationDriver($reader));


$dm = \Doctrine\ODM\MongoDB\
DocumentManager::create(new Mongo(), $config);
```

DocumentManager

- Accès central de la persistance des données
- Implemente le pattern Unif Of Work
- Gestion de l'état des objets en mémoire (NEW, MANAGED, DETACHED ou REMOVED)



Requêtes avec Doctrine

```
// SELECT ALL
$conferences = $dm->getRepository('\PHPTour\Conference')->findBy(array());

// SELECT BY ID
$conference = $dm->find('\ PHPTour\Conference', $id);

// SELECT BY CRITERIA
$conference = $dm->getRepository('\PHPTour\Conference')
                ->findOneBy(array('titre' => 'Hello $Mongo'));

// INSERT OR UPDATE
$dm->persist($conference);

// DELETE
$dm->remove($conference);
```

GridFS

```
$conference = $dm->createQueryBuilder
('\PHPTour\Conference')
                ->field('titre')
                ->equals('hello $Mongo;')
                ->getQuery()
                ->execute()
                ->getSingleResult();

header('Content-type: application/pdf');
echo $conference->getSlides()->getFichier()->getBytes();
```

Performances - Indexes

```
bd.Collection.ensureIndex({monChamp : 1})
bd.Collection.dropIndex()
bd.Collection.dropIndex({monChamp : 1})
```

- Opération coûteuse et bloquante
- Peut être réalisée en tâche de fond

```
db.Collection.ensureIndex({monChamp : 1},
                          {background : 1})
```

Performances - Indexes (2)

- Index sur plusieurs champs

```
db.ensureIndex({nom : 1, prenom : 1});
```

- Assurer l'unicité

```
db.ensureIndex({nom : 1, prenom : 1},
                {unique : 1});
```

Performances - Indexes (3)

Pas toujours efficaces

- négations ($ne, $not, ...)
- Opérations arythmétiques ($mod, ...)
- Map / Reduce (boite noire pour l'optimiseur)
- La plupart des expressions rationelles (/a/)

Outils dev. et admin.

- RockMongo
- Fang of Mongo
- MongoExplorer (silverlight)
- MongoHub
- PHPMoAdmin
- Meclipse
- MongoVue

Questions ?

https://github.com/geraldcroes/MongoConference/

Photographies

Hangin by a thread - http://www.500px.com/photo/3159488 - Scott Smora
Pacific coast rail line - http://www.500px.com/photo/1739071 - Lachlan
Crossroads - http://500px.com/photo/206202 - DMitry Kot
Old book - http://500px.com/photo/841759 - Amar Chauhan
Inappropriately Dressed - White dress fast bike - http://500px.com/photo/330960 - **Apollinaria Toloraya**
junction - http://500px.com/photo/3113517 - **Kai Ziehl**
the cube - http://500px.com/photo/1365861 - Stephen L.
Sans titre - http://500px.com/photo/1125283 - **James Tatum**
What's in the box?- http://500px.com/photo/1292897 - **Andrew Do**
gears of war - http://500px.com/photo/672778 - Dave Kane
Link - http://500px.com/photo/514318 - **Djura Stankovic Photography**
Russian dolls lined up - http://www.flickr.com/photos/sunnyuk/3240916291/ - sunnyUK
records - http://500px.com/photo/1084138 - **Jackson Carson**
microphone - http://www.500px.com/photo/1910034 - Tchon T