

**Choix d'une base
de données pour
stocker un
historique
d'événements**

Frontend

- Plateforme de site web
- Django/PostgreSQL/SOLR



The screenshot shows the homepage of the 'OPEN GEOMECHANICS' journal. At the top, there are navigation links for 'HOME', 'ABOUT', and 'SUBMIT'. Below these is a search bar. The main content area features a large image of hands holding a piece of translucent material. Below the image, the text reads: 'Editorial', 'Andó, Edward ; Dijkstra, Jelke ; Frost, David ; Herle, Ivo ; Hurley, Ryan ; Marks, Benjy ; Muir Wood, David ; Tamagnini, Claudio ; Viggiani, Cino', and 'This Editorial is the first publication from the journal Open Geomechanics, a radically open-access scientific journal for Geomechanics Research, edited by Geomechanics researchers for Geomechanics researchers. We believe that the results of scientific...'. The date 'Published: 2019-01-28' and a 'PDF' link are also visible. At the bottom, there are logos for the European Union, ALERT Geomaterials, Mathdoc, and the Centre Mersenne logo.

The screenshot shows the website for 'ANNALES HENRI LEBESGUE'. It features a navigation menu with links for 'CONTENTS', 'EDITORIAL BOARD', 'SUBMISSION', and 'ABOUT'. Below the menu, there are links for 'PREVIOUS ARTICLE', 'TABLE OF CONTENTS', and 'NEXT ARTICLE'. The main content area displays the title 'VOISIN, CLAIRE', 'Chow ring and gonality of general abelian varieties', and 'Annales Henri Lebesgue, Volume 1 (2018), p. 313-332'. There are also links for 'Download PDF' and 'Download bibTeX entry'. Below this, there is a 'Metadata' section with the DOI '10.5802/ahl.18' and 'KEYWORDS: Abelian varieties, covering gonality, zero-cycles, Chow ring'. An 'Abstract' section follows, starting with 'We study the (covering) gonality of abelian varieties and their orbits of zero-cycles for rational equivalence. We show that any orbit for rational equivalence of zero-cycles of degree k has dimension at most $k-1$. Building on the work of Pirola, we show that very general abelian varieties of dimension g have (covering) gonality at least $f(g)$, where $f(g)$ grows

Backend



- Même plateforme
- Authentification/CKEditor

The screenshot displays the 'PTF Tools' web interface. On the left is a dark sidebar with a list of collections: AFST, AHL, AIF, ALCO, AMBP, CML, JEP, JTNB, MSIA, OGEO, PMB, SMAI-JCM, NUMDAM, and HISTORIQUE. The main content area shows a document page for 'No temporal distributional limit theorem for a.e. irrational translation' by Dolgopyat, Dmitry; Sarig, Omri. The page includes a progress bar at 36%, navigation buttons (Previous, Next), and action buttons like 'Matching', 'Tous', 'Annuler', 'Marquer les ids comme vérifiés', and 'Marquer les ids comme non vérifiés'. Below the title, there is an abstract section and a 'References' section with three entries, each having a green checkmark and a trash icon. The bottom of the page shows a footer with a page number '3'.



COLLECTIONS

AFST

AHL

AIF

ALCO

AMBP

CML

JEP

JTNB

MSIA

OGEO

PMB

SMAI-JCM

Import Edit Matching Deploy Archive CLOCKSS Tous
Revus Tous
Errors + Warnings Tous

2019-03

246 events

2 Errors

1 Warnings

+ ♥ Matching	2019-03-18 09:35	AIF_2011__61_3_799_0	3 ids	HTTPConnectionPool(host='doi.crossref.org', por... Warning	
♥ Matching	2019-03-18 09:33	ALCO_2018__1_1_3_0	[18] zbl:1384.05175		
♥ Matching	2019-03-18 09:33	ALCO_2018__1_1_23_0			
♥ Matching	2019-03-18 09:32	ALCO_2018__1_1_3_0			
+ ♥ Matching	2019-03-18 09:32	AHL_2018__1__1_0	2 ids		
+ Deploy	2019-03-18 09:28	AIF_2017__67_4			
Numdam	2019-03-18 09:28	AIF_2017__67_4_1783_0			
Numdam	2019-03-18 09:28	AIF_2017__67_4_1741_0			
+ Numdam	2019-03-18 09:28	AIF_2017__67_4_1725_0		Command 'pdf2djvu --quiet --dpi 600 --output /m... Error	
Numdam	2019-03-18 09:28	AIF_2017__67_4_1671_0			

COLLECTIONS

AFST

AHL

AIF

ALCO

AMBP

CML

JEP

JTNB

MSIA

OGEO

PMB

SMAI-JCM

Import

Edit

Matching

Deploy

Archive

CLOCKSS

Tous

Reuves ▾

Tous

Errors

+ Warnings

Tous

2019-03

247 events

2 Errors

1 Warnings

♥ Matching	2019-03-18 14:13	AHL_2018__1__127_0	zbl:1368.37016	
- ♥ Matching	2019-03-18 09:35	AIF_2011__61_3_799_0	<div style="background-color: #333; color: white; padding: 2px; border-radius: 3px; display: inline-block;">3 ids</div> HTTPConnectionPool(host='doi.crossref.org', por... Warning	
		numdam:AIF_2011__61_3_799_0 [8] doi:10.1016/S1631-073X(02)02537-2 [11] doi:10.1007/3-540-45328-8_2	HTTPConnectionPool(host='doi.crossref.org', port=80): Read timed out. (read timeout=2) Traceback (most recent call last): File "/var/www/ptf_tools/current/venv/lib/python3.5/site-packages/requests/packages/urllib3/connectionpool.py", line 386, in _make_request six.raise_from(e, None) File "", line 2, in raise_from File "/var/www/ptf_tools/current/venv/lib/python3.5/site-packages/requests/packages/urllib3/connectionpool.py", line 382, in _make_request httplib_response = conn.getresponse() File "/usr/lib/python3.5/http/client.py", line 1197, in getresponse response.begin() File "/usr/lib/python3.5/http/client.py", line 297, in begin version, status, reason = self._read_status() File "/usr/lib/python3.5/http/client.py", line 258, in _read_status line = str(self.fp.readline(_MAXLINE + 1), "iso-8859-1") File "/usr/lib/python3.5/socket.py", line 575, in readinto return self._sock.recv_into(b) socket.timeout: timed out During handling of the above exception, another exception occurred: Traceback (most recent call last): File "/var/www/ptf_tools/current/venv/lib/python3.5/site-packages/requests/adapters.py", line 423, in send timeout=timeout	



Besoins

Historique des événements de ptf-tools

Regrouper par date, type

Séparé de la base de données des articles

Pas de schéma

Ajouts d'événements à partir d'un json

Besoins

Exemples

```
{
  'type': "deploy",
  'pid': "ALCO_2018_1_5_",
  'status': "OK",
}

{
  'type': "matching",
  'pid': "AIF_2015_65_6_2331_0",
  'status': "WARNING",
  'message': "MR ne répond pas",
  'ids': [ {'type': 'zbl', 'id': 'zbl1', 'seq': 15},
            {'type': 'zbl', 'id': 'zbl2', 'seq': 22} ]
}
```

MongoDB

“Il vaut mieux se tromper en allant de l'avant
que d'avoir raison en reculant.”

Citation de Frédéric Dard ; Les pensées de San-Antonio (1996)



- Document database
- JSON documents
- Gratuit et open-source
- Très bien documenté
- Extensions pour Django



Djongo

- Recommandé
- S'utilise comme l'ORM standard de Django
- Compatible avec django-admin



A utiliser si MongoDB est la base de donnée principale de l'application

The logo for MongoEngine, featuring the text "MongoEngine" in a bold, white, sans-serif font. The text is centered within a white rectangular border. The background behind the border is a vibrant, abstract brushstroke that transitions from a deep red at the top to a bright orange at the bottom, with a white-to-orange gradient background for the entire slide.

Création de classes multiples, même si tous les documents sont stockés dans la même collection

```
class MatchedID(EmbeddedDocument):
    type = StringField(max_length=10)
    id = StringField(max_length=30)
    seq = IntField()

class Event(Document):
    type = StringField(max_length=30)
    pid = StringField(max_length=100)
    status = StringField(max_length=30)
    message = StringField()
    created_on = DateTimeField()

    meta = {'allow_inheritance' }

class MatchingEvent(Event):
    ids = ListField(EmbeddedDocumentField(MatchedID))
```

The logo for MongoEngine, featuring the text "MongoEngine" in a bold, white, sans-serif font. The text is centered within a white rectangular border. The background behind the border is a vibrant, horizontal brushstroke that transitions from a deep red at the top to a bright orange at the bottom.

MongoEngine



On manipule les objets
comme d'habitude

```
myclass.objects.filter(...)
```



Lent

<https://stackoverflow.com/questions/35257305/mongoengine-is-very-slow-on-large-documents-compared-to-native-pymongo-usage>

MongoEngine is spending
ages converting all the
returned arrays to dicts

A graphic on the left side of the slide featuring a white rectangular border. Inside the border, there are two horizontal brush strokes: a red one on top and an orange one on the bottom. The text 'MongoEngine' is written in white, bold, sans-serif font across the red stroke.

MongoEngine

`myclass.objects.all()` ne préserve pas l'ordre d'insertion

Pour utiliser l'agrégation, il faut de toute façon utiliser la syntaxe de `pymongo`

The logo for pymongo, featuring the word "pymongo" in a white, lowercase, sans-serif font. The text is centered within a white rectangular border. The background behind the border is a vibrant, abstract brushstroke that transitions from a deep red at the top to a bright orange at the bottom, with a white-to-orange gradient.

Aggregation pipeline

- Permet d'enchaîner les étapes pour agréger les données
- Stages: \$match, \$group, \$facet, \$addFields, \$sort...
- Syntaxe basée sur du JSON

The logo for pymongo, featuring the word "pymongo" in a white, lowercase, sans-serif font. It is positioned on the left side of the slide, overlaid on a large, abstract brushstroke graphic that transitions from red at the top to orange at the bottom.

Aggregation pipeline

```
pipeline = [  
    {"$addField": {  
        "matched": filter if len(filter) > 0 else True  
    }},  
    {"$match": {"matched": True}},  
    {"$sort": {"created_on": -1}},  
    {"$group": {  
        '_id': {'year': {'$year': "$created_on"}, 'month': {'$month':  
"$created_on"}},  
        'events': {'$push': "$$ROOT"},  
        'count': {'$sum': 1},  
        'error_count': {'$sum': {'$cond': {'if': {'$and': [{'$eq':  
["$status", "ERROR"]}, {'$ne': ["$obsolete", True]}]},  
            'then': 1,  
            'else': 0}}}},  
    }  
    },  
    {"$sort": {"events.created_on": -1}},  
]
```

```
grouped_events = events.aggregate(pipeline=pipeline)
```

**MongoDB ne
conserve pas
l'ordre d'insertion !!!**



Capped Collection



Conserve l'ordre d'insertion

Obligation de définir une limite

```
events = db.create_collection("events",  
capped=True, size=5242880, max=5000)
```

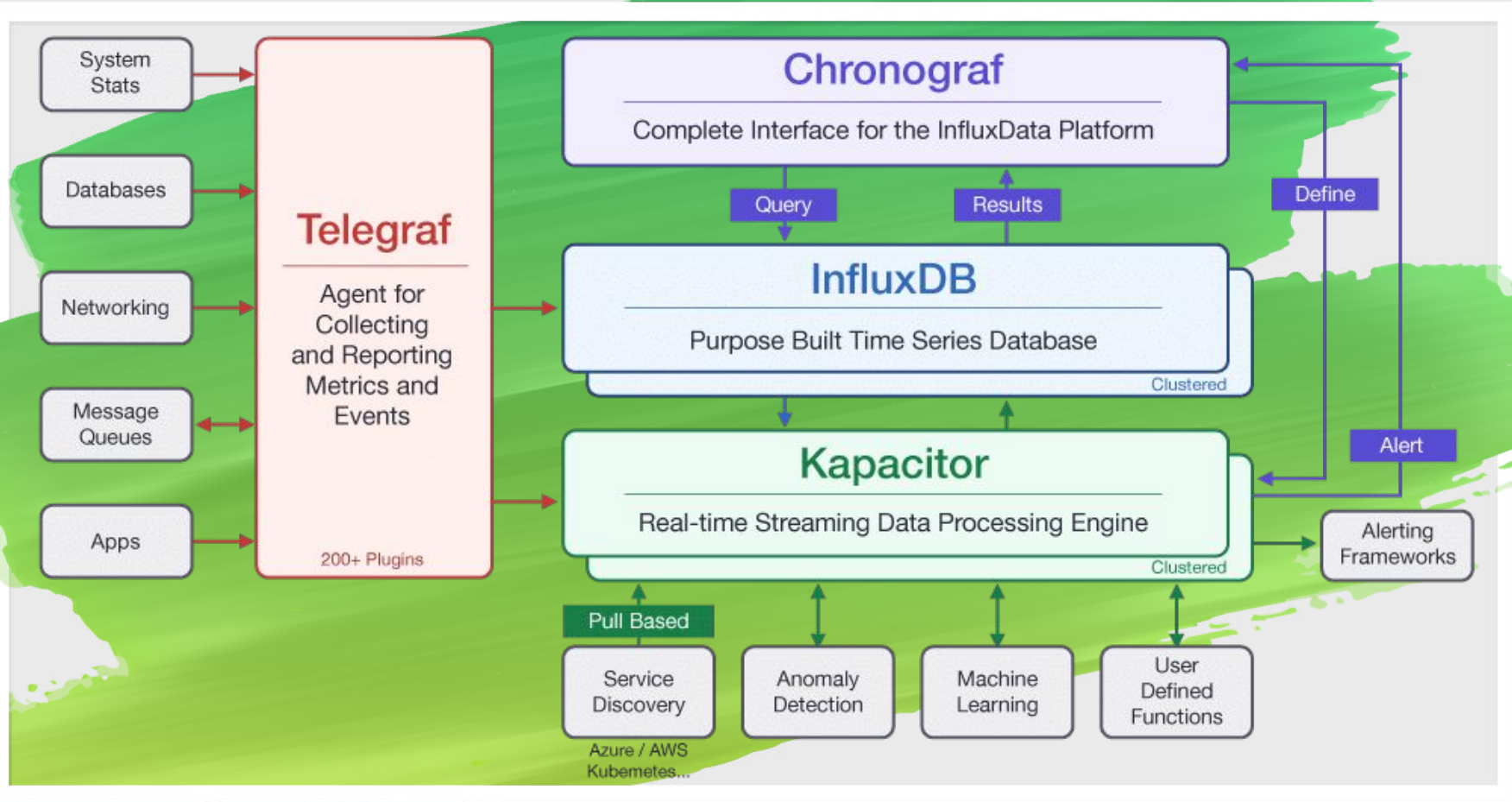


Une fois créé, un document ne peut pas voir sa taille augmenter

```
events.update_one(...)
```

InfluxDB

open-source time series database





<https://medium.com/@xaviergeerinck/how-to-visualize-metrics-with-grafana-and-chronograf-by-using-influxdb-450a6b796710>

The InfluxDB logo is presented within a white rectangular frame. The background of the frame is a vibrant green with a horizontal brushstroke texture. The text 'InfluxDB' is centered in a bold, white, sans-serif font.

InfluxDB

Tags are indexed and **fields** are not indexed

Field values are the actual data; they can be strings, floats, integers, or booleans

Timescale

Open-source time-series database
powered by PostgreSQL

PostgreSQL + JSON

Schéma

```
class HistoryEvent(models.Model):  
  
    created_on = models.DateTimeField(...)  
    type = models.CharField(...)  
    pid = models.CharField(...)  
    col = models.CharField(...)  
    status = models.CharField(...)  
  
    data = JSONField(default={})
```

Schéma

```
class HistoryEvent(models.Model):  
  
    created_on = models.DateTimeField(...)  
    type = models.CharField(...)  
    pid = models.CharField(...)  
    col = models.CharField(...)  
    status = models.CharField(...)  
  
    data = JSONField(default={})
```

Django ORM

```
event = HistoryEvent(type='matching',
                    pid='AIF_2007__57_7',
                    col='AIF',
                    status='ERROR',
                    data= {
                        'message': "Internal Error",
                        'ids_count': 3,
                        'articles': [
                            {'pid': 'AIF_2007__57_7_2143_0',
                             'ids': [{'type': 'zbl', 'id': 'zbl3', 'seq': 5}]
                            },
                            {'pid': 'AIF_2007__57_7_2143_0',
                             'ids': [
                                 {'type': 'zbl', 'id': 'zbl4', 'seq': 8},
                                 {'type': 'mr', 'id': 'mr2', 'seq': 12}]
                            }
                        ]
                    }
                )
```

Django ORM

```
>>> Dog.objects.create(name='Rufus', data={
...     'breed': 'labrador',
...     'owner': {
...         'name': 'Bob',
...         'other_pets': [{
...             'name': 'Fishy',
...         }],
...     },
... })
```

```
>>> Dog.objects.filter(data__owner__name='Bob')
```

Agrégation

```
data = HistoryEvents.objects
    .filter(**kwargs)
    .order_by('-pk')
    .annotate(month=TruncMonth('created_on'))
```

```
# On crée les groupes à la main
for event in data:
    grouped_events.append(...)
```