



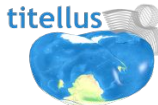
GeoNetwork
open source

GeoNetwork

Going to micro-services

campto**camp**[™]
INNOVATIVE SOLUTIONS
BY OPEN SOURCE EXPERTS

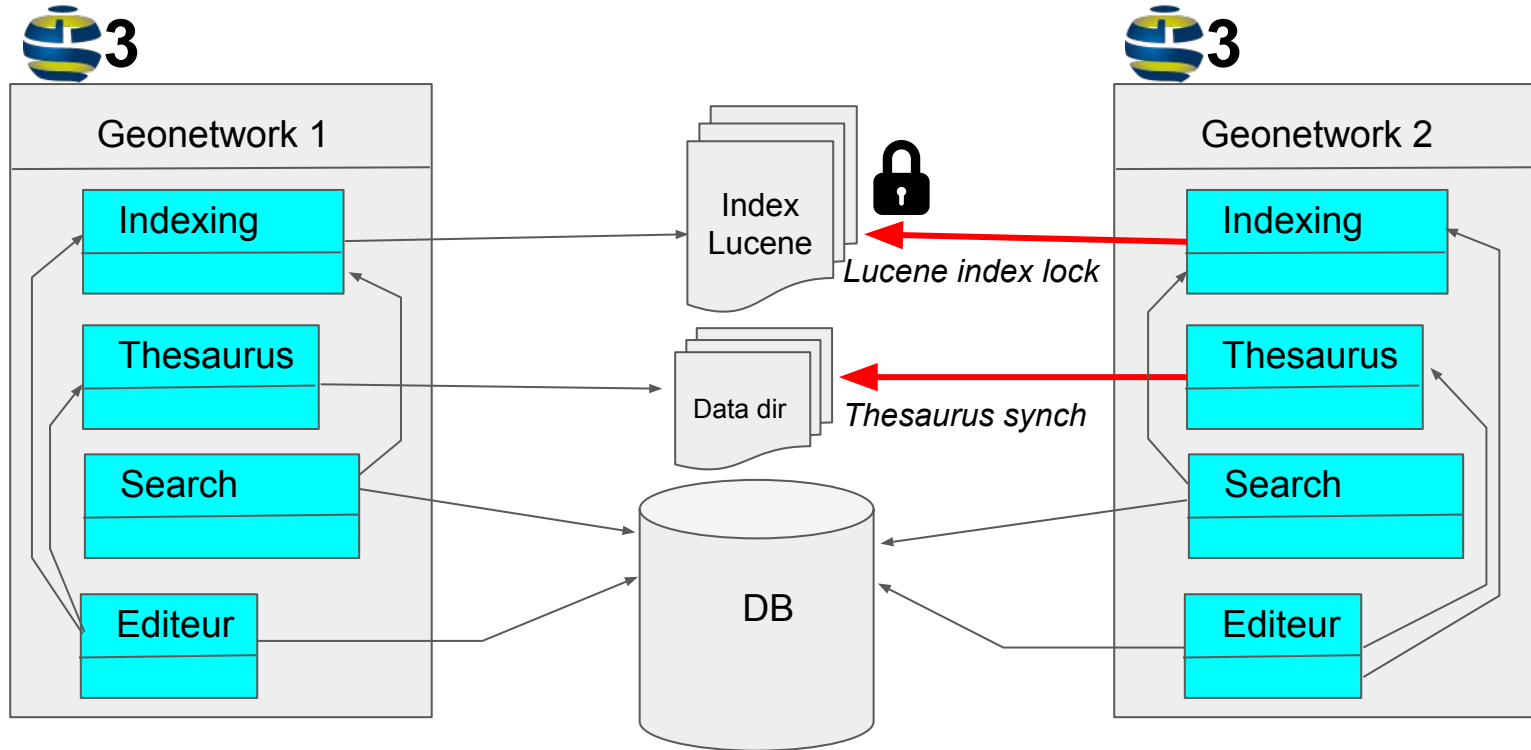
GEOCAT
GOVERNMENT GEOGRAPHIC DATA PUBLISHING



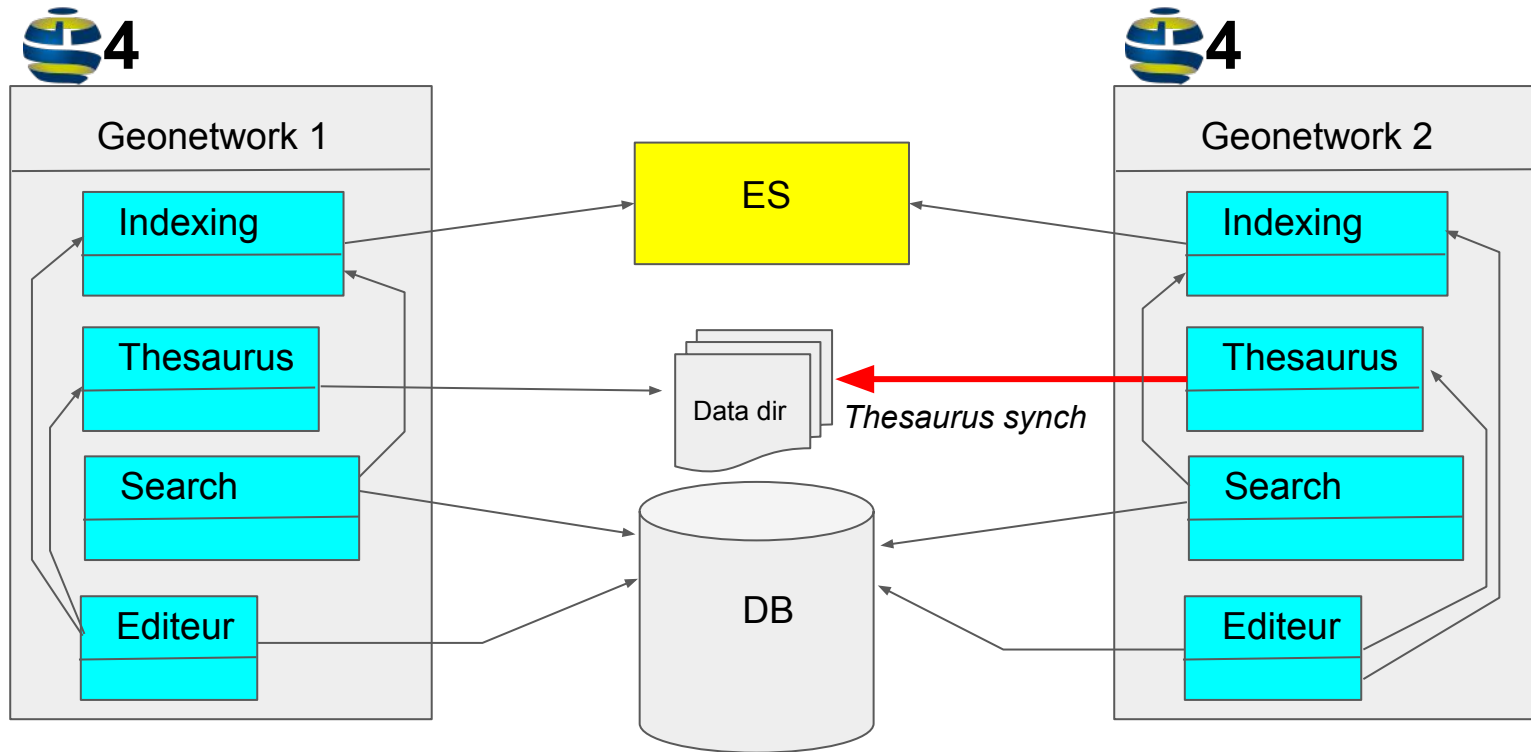
Microservices for resilience

- Moving to Elasticsearch is a first step towards the "microservices" approach
- If multiple instances write to the same lucene index, they run into a file lock. Problem solved if GeoNetwork does not manage the index by itself anymore
- It is not the Geonetwork business anymore to manage the lucene index => less code to maintain

Clustering issues in version 3



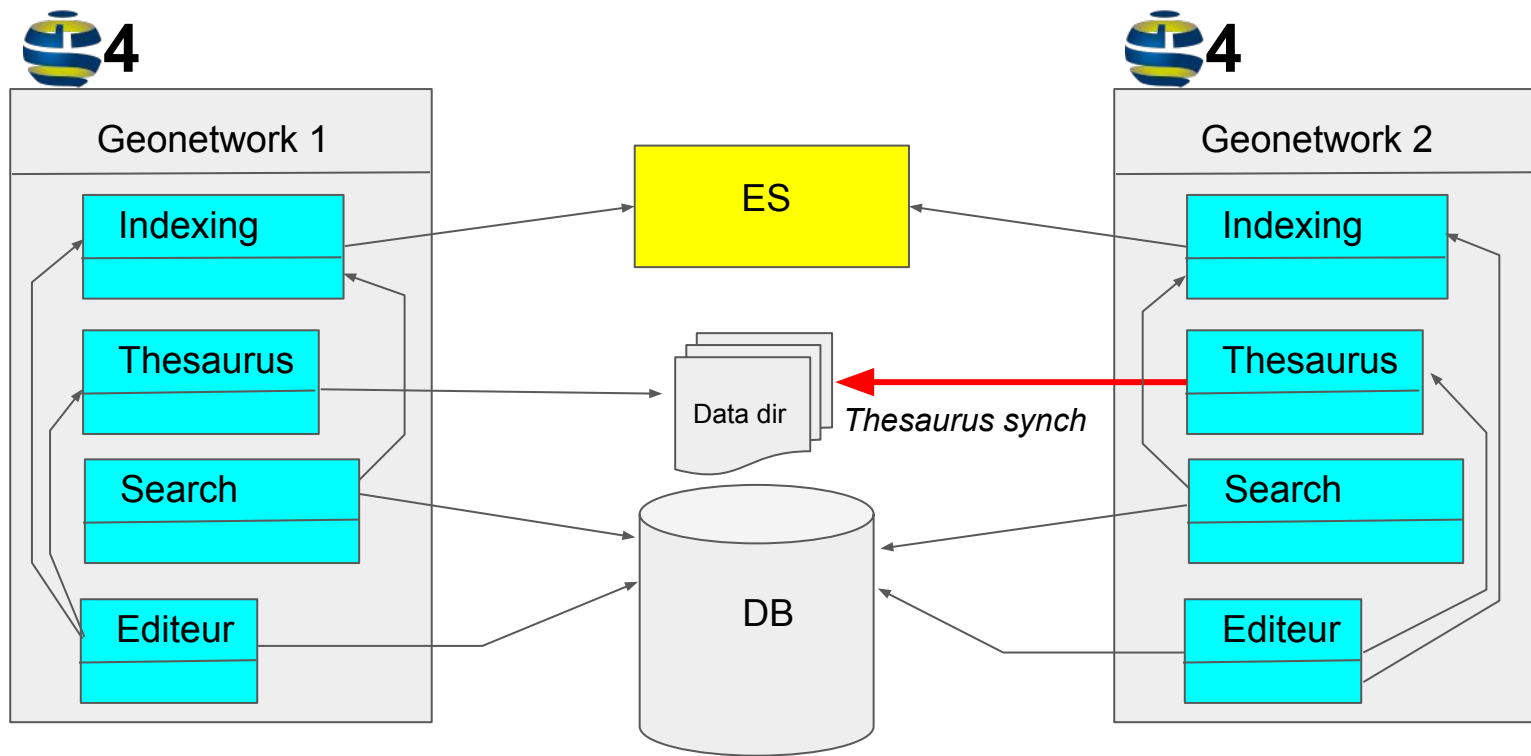
Version 4 solves the index lock



Version 4 solves the index lock ...

But still some other issues

(eg. thesaurus, formatter cache)



"100% availability"

eg. geocat.ch 🐱

Can the GeoNetwork search (and CSW) be 100% available, even if the editor is down / very busy ?

=> How to split Geonetwork even more ?

=> How to scale search functions ?

Harvesting should not alter main app performances

eg. Ifremer/Sextant 

How to split the application and have background tasks running independently ?

(background tasks = harvesting, validation, processing, link analysis, indexing ...)

Some tasks are too resource-consuming

eg. every actions related to MD selection

- How to stream large responses ?
- Shall we move some actions to asynchronous processes running in the background ?
- If so, which ones ? MEF-PDF-CSV export, ...

Modularity

- Projects may only require harvesting and search features (eg. a national INSPIRE portal)
- Some components could be turned off (and free some resources)
 - Editor
 - Thesaurus
 - Map viewer
 - ...

Microservices can
help solving these
issues

What's next?

What is GeoNetwork ?

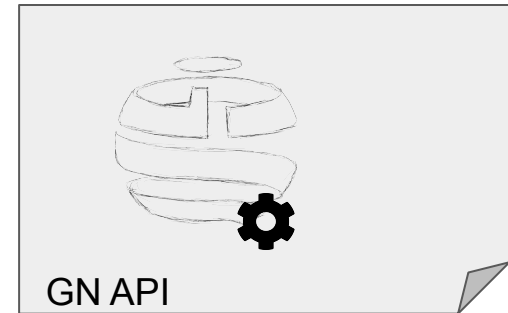
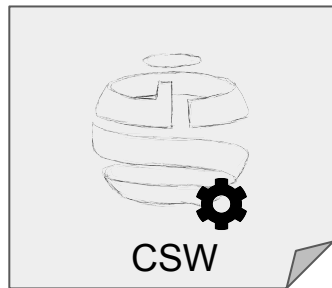
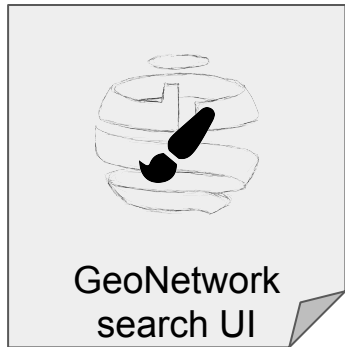
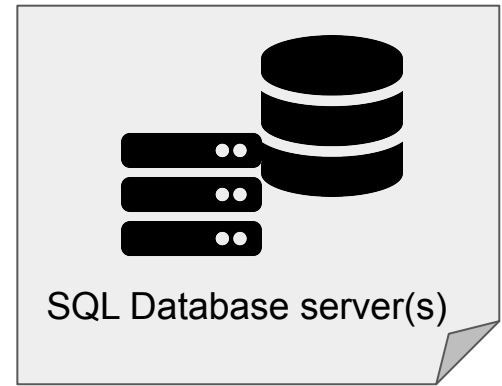
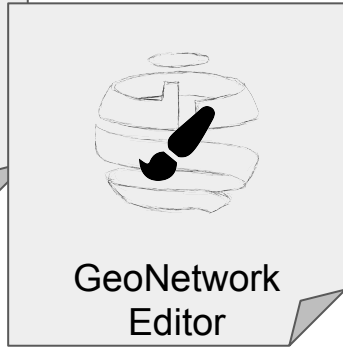
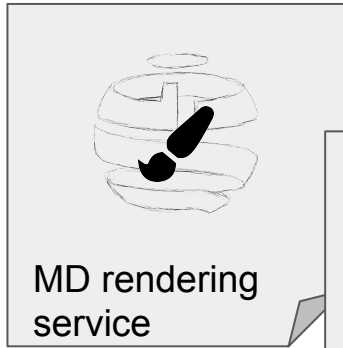


GeoNetwork
open source

Technically speaking, the geonetwork webapp embeds:

- A relational database (h2, PostgreSQL, ...)
- A thesaurus repository (RDF store)
- An indexation system (Lucene)
- A web UI based on AngularJS, allowing to:
 - Search
 - Edit metadatas
 - Visualize spatial datas
- Some harvesters to crawl remote services with different protocols
- And other features ...

Could it be more like ... ?



Each component

- could be instantiated alone ... (**modularity**)
- ... on its own resources (**horizontal scalability, response time more predictable**)
- Several times (**availability**)
- Not deployed if not needed (**resource-savvy**)

*"A microservice is a **small, loosely coupled, distributed service.***

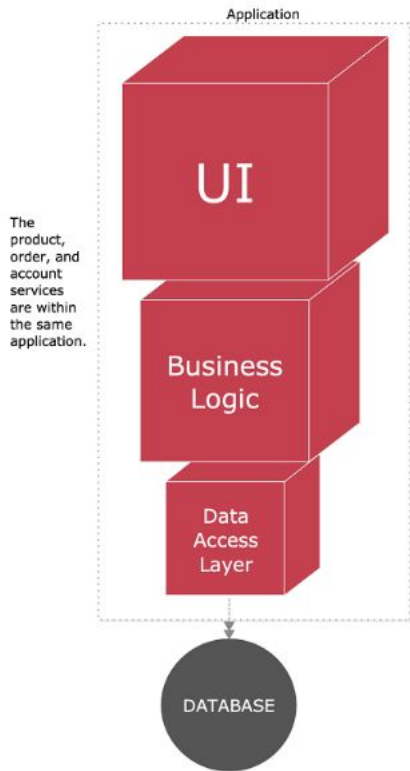
*Microservices allow you to take an **extensive application** and decompose it into **easy-to-manage components with narrowly-defined responsibilities.**"*

Challenges

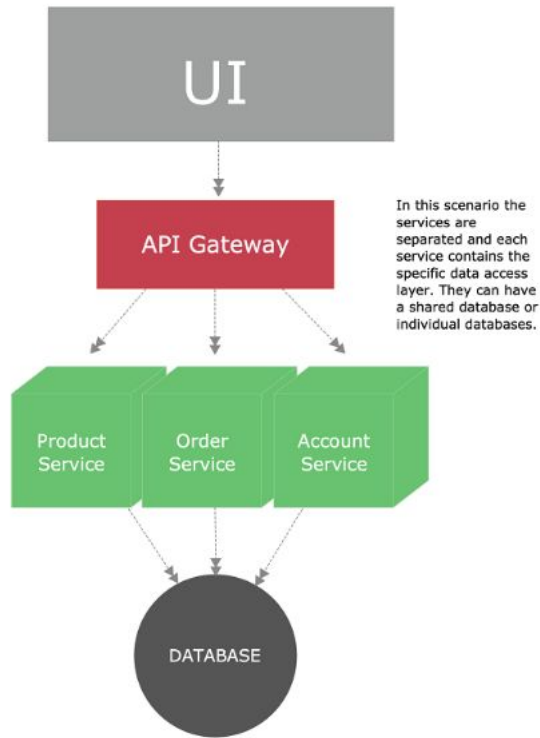
- From 1 monolithic webapp to several ones, which have to be taken care of separately ~ from 1 single server to a distributed system
- Having a good monitoring solution becomes crucial to:
 - Detect faults fast
 - Act consequently
- It will also be a good opportunity to do some cleanup:
 - Simplifying access controls
 - Remove unused features
 - Removing last parts relying on the former Java framework (Jeeves)
 - Ensure compatibility with newer versions of Java
 - Dependency upgrades

Monolithic vs microservices architecture

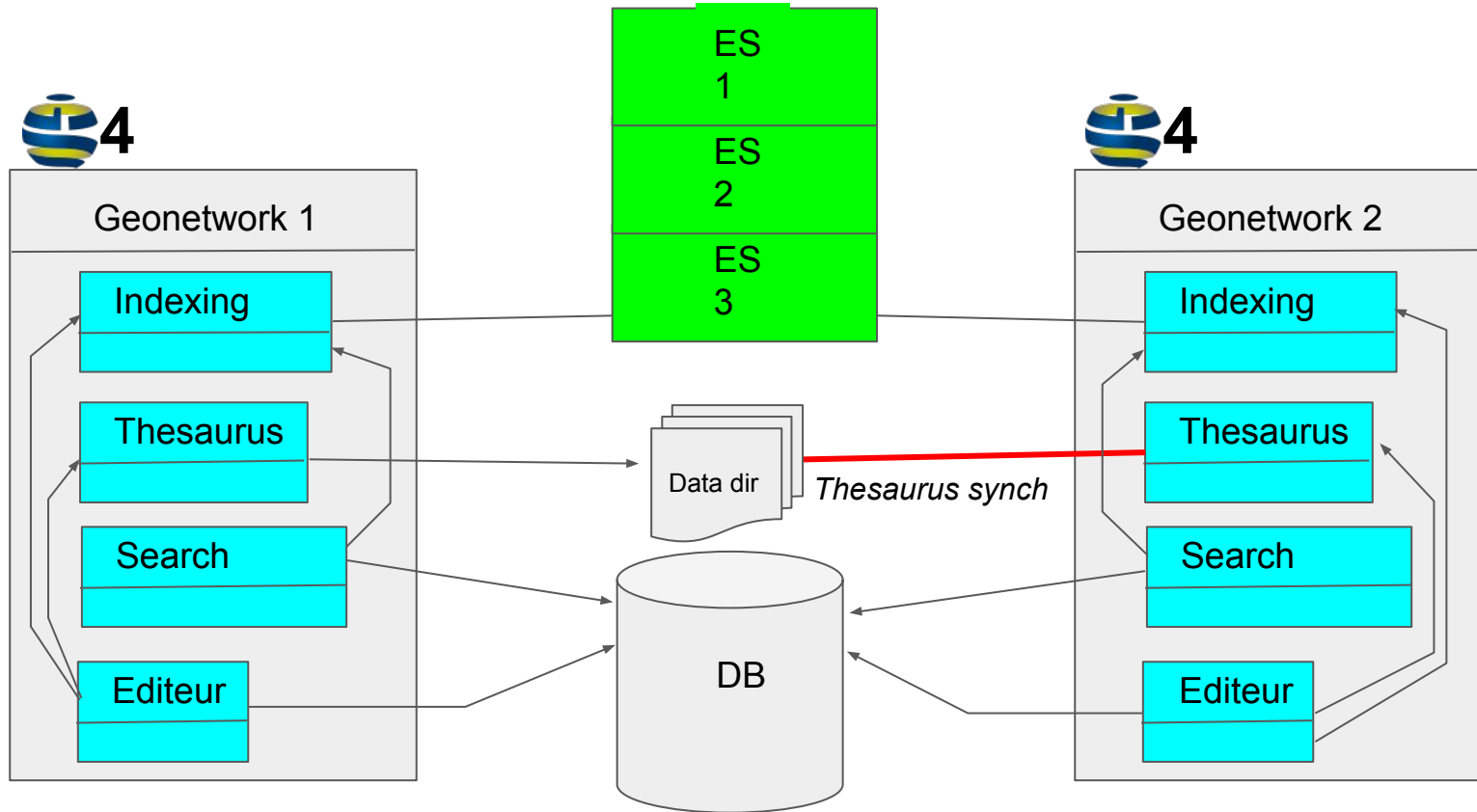
Monolithic Architecture



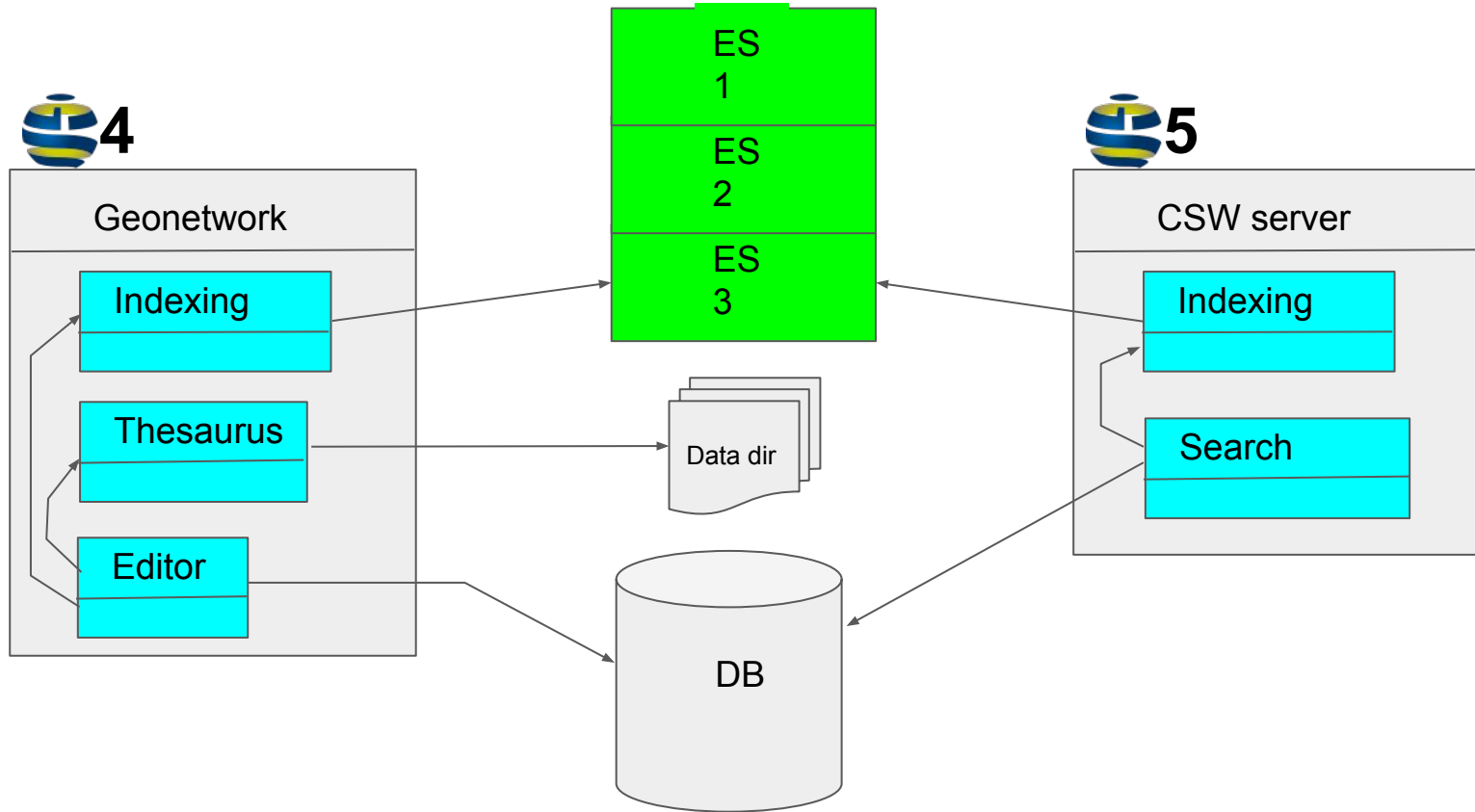
Microservices Architecture



Cluster the index



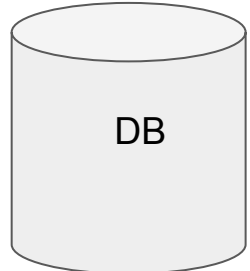
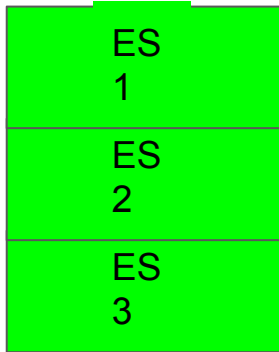
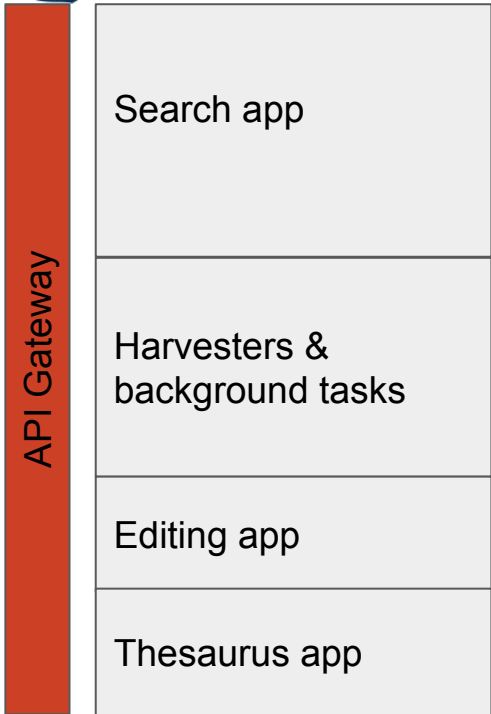
GeoNetwork Modularity



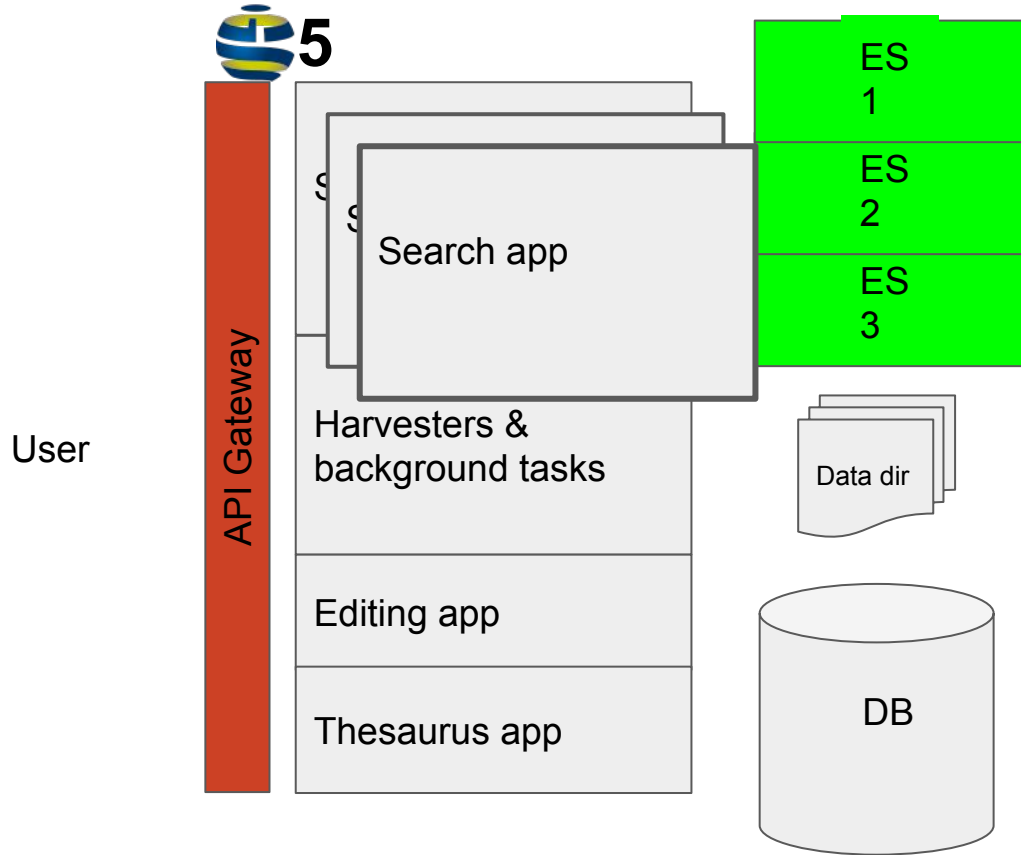
“Microservices”



User



Scalability



A suggested "roadmap" / Conception & study

- What / how to split things ? Which features should be kept ? What could be abandoned ? What could be a microservice in itself ?
"GeoNetwork features tour"
- Does the data model (SQL Database) require a rework, having the microservices approach in mind ?
- Identify the "locks" (file lock on lucene indexes was an example)
- Setting up message queueing for microservices interactions & communications: state of the art ?
- Do we need to reorganize the project's code layout (repository) ?

A suggested "roadmap" / Development

- Updating libraries / Improving new Java versions compatibility
- Begin with a simple one: splitting the harvesting logic and/or the CSW server into a separate service outside of the main webapp
- The "API gateway" brick (schema on the previous slides)
- Iterating over the other identified parts

Conclusion

- Some issues & Nice-to-have in GeoNetwork have been identified
- Microservices can be an answer to some of them
- Going to microservices will require investigations & conception
- Elasticsearch was a first step to the "Microservices journey"



GeoNetwork
open source

Thank you

Rest API

**Simple view formatting
microservice**

es index

"uuid": '9fa99621-f639-436b-b5b2...'
"title": 'Zonage d'assainissement ...'
"abstract": 'Document réglementa...'
...

"uuid": '2294-4e48-9c0f-fc83c5478...'
"title": 'Museum von dem Museum...'
"abstract": 'Interaktive Karte der M...'
...

"uuid": '4bd598c6-7677-44bc-928f...'
"title": 'Prescription linéaire du P...'
"abstract": 'Le Code de l'urbanism...'
...

Rest API

**Searching
microservice**

es index

"uuid": '9fa99621-f639-436b-b5b2...'
"title": 'Zonage d'assainissement ...'
"abstract": 'Document réglementa...'
...

"uuid": '2294-4e48-9c0f-fc83c5478...'
"title": 'Museum von dem Museum...'
"abstract": 'Interaktive Karte der M...'
...

"uuid": '4bd598c6-7677-44bc-928f...'
"title": 'Prescription linéaire du P...'
"abstract": 'Le Code de l'urbanism...'
...

Rest API

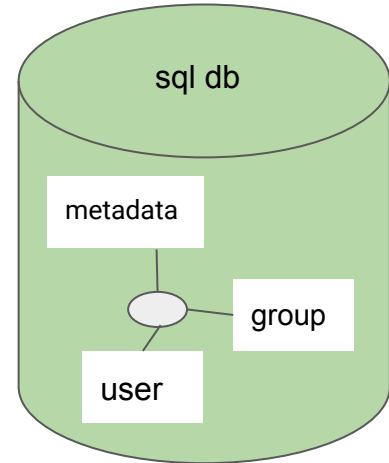
**Full view formatting
microservice**

sql db

metadata

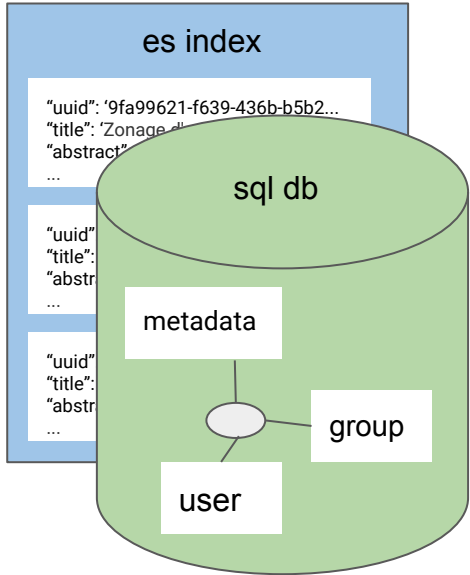
group

user



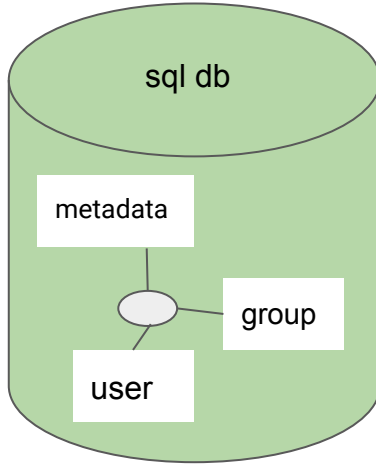
Rest API

Indexing
microservice



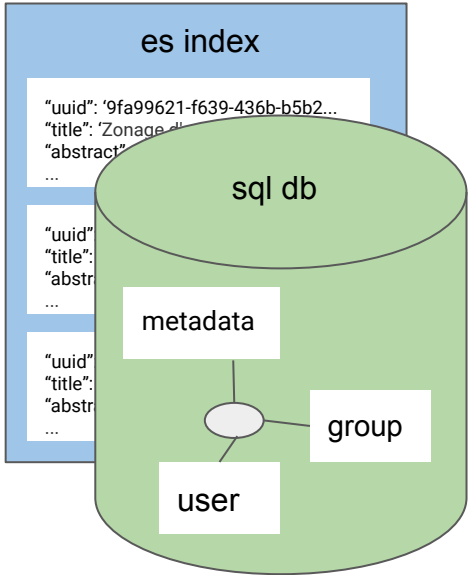
Rest API

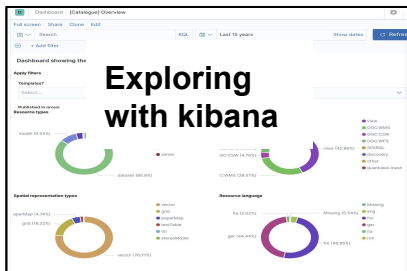
Editing / validating
microservice



Rest API

Being harvested
microservice





es index

```

{
  "uid": "9fa99621-f639-436b-b5b2...",
  "title": "Zonage d'assainissement ...",
  "abstract": "Document réglementa...",
  ...
}

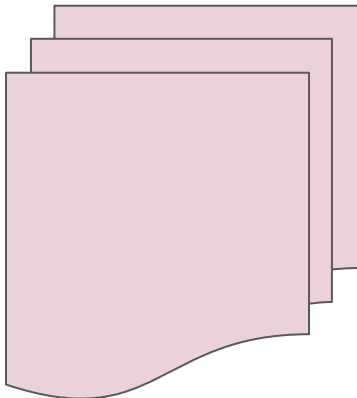
{
  "uid": "2294-4e48-9c0f-fc83c5478...",
  "title": "Museum von dem Museum...",
  "abstract": "Interaktive Karte der M...",
  ...
}

{
  "uid": "4bd598c6-7677-44bc-928f...",
  "title": "Prescription linéaire du P...",
  "abstract": "Le Code de l'urbanism...",
  ...
}

```

Rest API

Caching & retrieving formatted view microservice



Simple view formatting microservice

es index

```
"uuid": "9fa99621-f639-436b-b5b2-..."  
"title": "Zonage d'assainissement ..."  
"abstract": "Document réglementa..."  
...
```

```
"uuid": "2294-4e48-..."  
"title": "Museum vor..."  
"abstract": "Interakt..."  
...
```

```
"uuid": "4bd598c6-7..."  
"title": "Prescription..."  
"abstract": "Le Code..."  
...
```

Searching microservice

es index

```
"uuid": "9fa99621-f639-436b-b5b2-..."  
"title": "Zonage d'assainissement ..."  
"abstract": "Document réglementa..."  
...
```

Caching & retrieving formatted view microservice

md1

extent

```
title: extent1  
periode: date1 - date2  
periode: date3 - date4
```

extent

```
title: extent2  
emprise: geom1  
emprise: geom2
```

md2

extent

```
title: extent1  
emprise: geom1  
emprise: geom2
```

extent

```
title: extent2  
periode: date1 - date2  
periode: date3 - date4
```

Same index

```
extentTitle: extent1  
extentTitle: extent2  
periode: date1 - date2  
periode: date3 - date4  
emprise: geom1  
emprise: geom2
```

Emprise

- photos 1974 (FR)

Étendue spatiale



Étendue temporelle

Date de création

1974-01-01

Période

Tue Jan 01 1974 00:00:00 GMT+0100 ▶ Tue Jan 01 1974 00:00:00 GMT+0100