# Elasticsearch from the Bottom Up

Alex Brasetvik
alex@found.no
@alexbrasetvik

# Elasticsearch from the Bottom Up

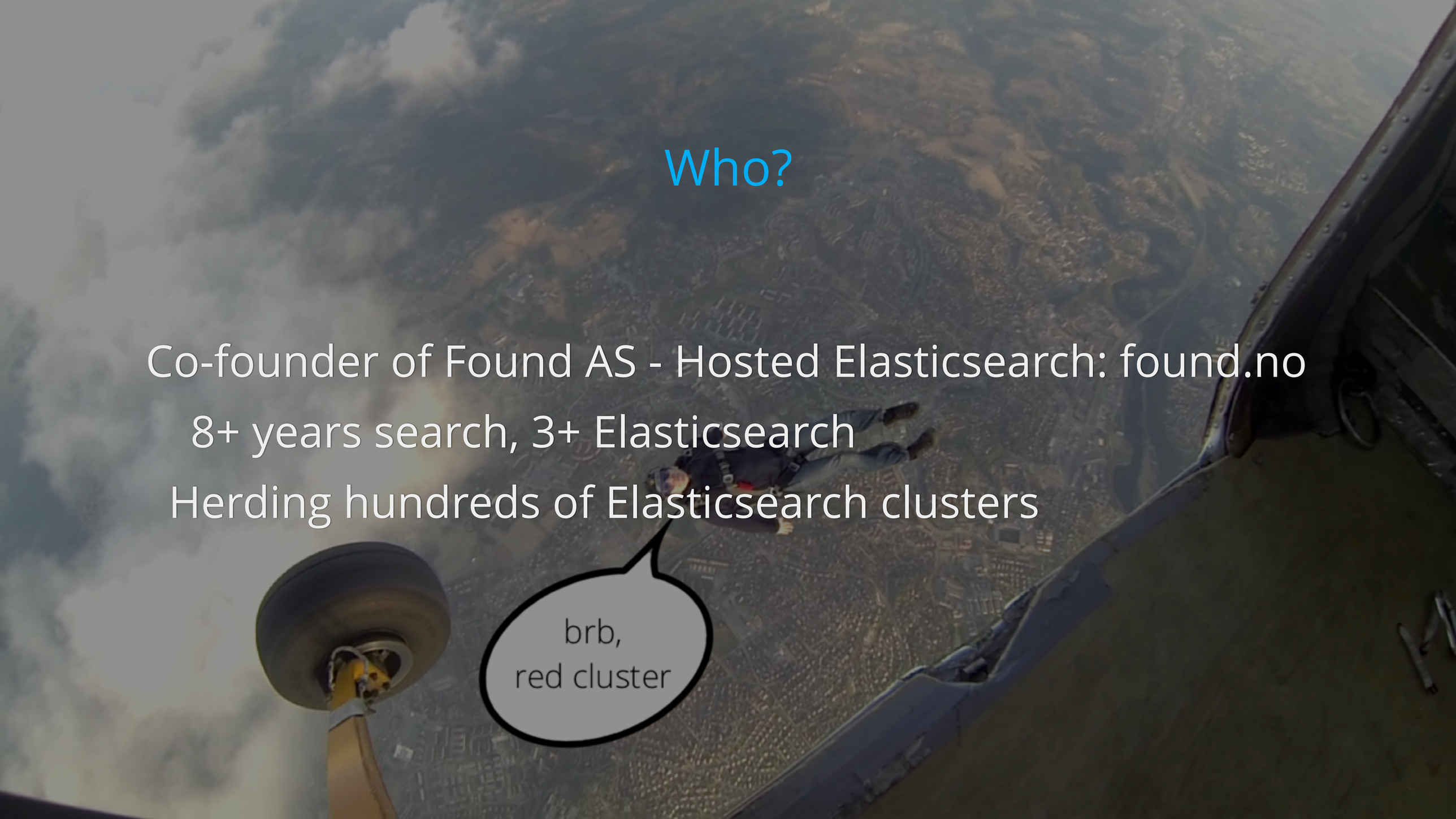
Alex Brasetvik
alex@found.no
@alexbrasetvik

# Who?

Co-founder of Found AS - Hosted Elasticsearch: found.no

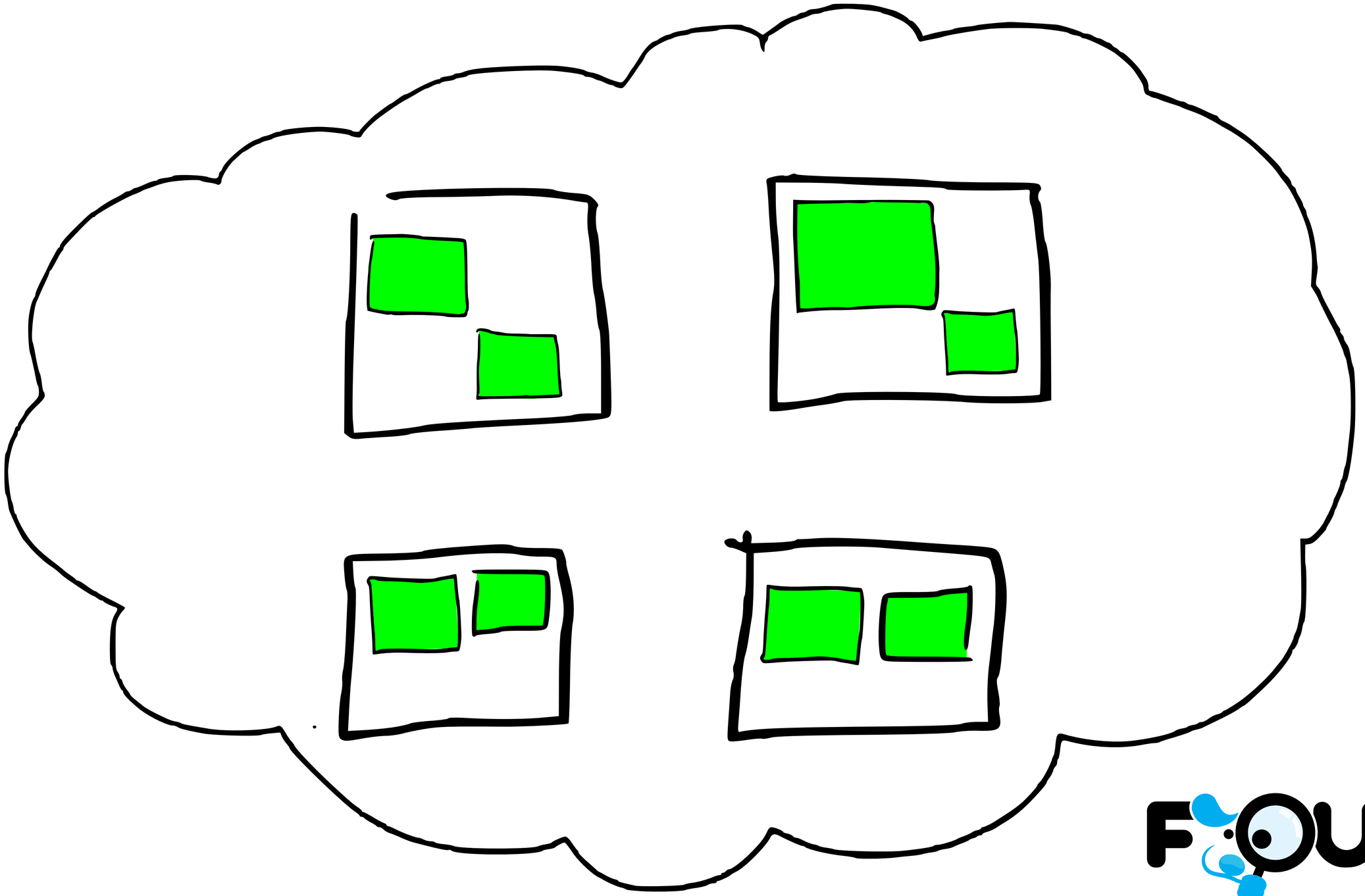
8+ years search, 3+ Elasticsearch

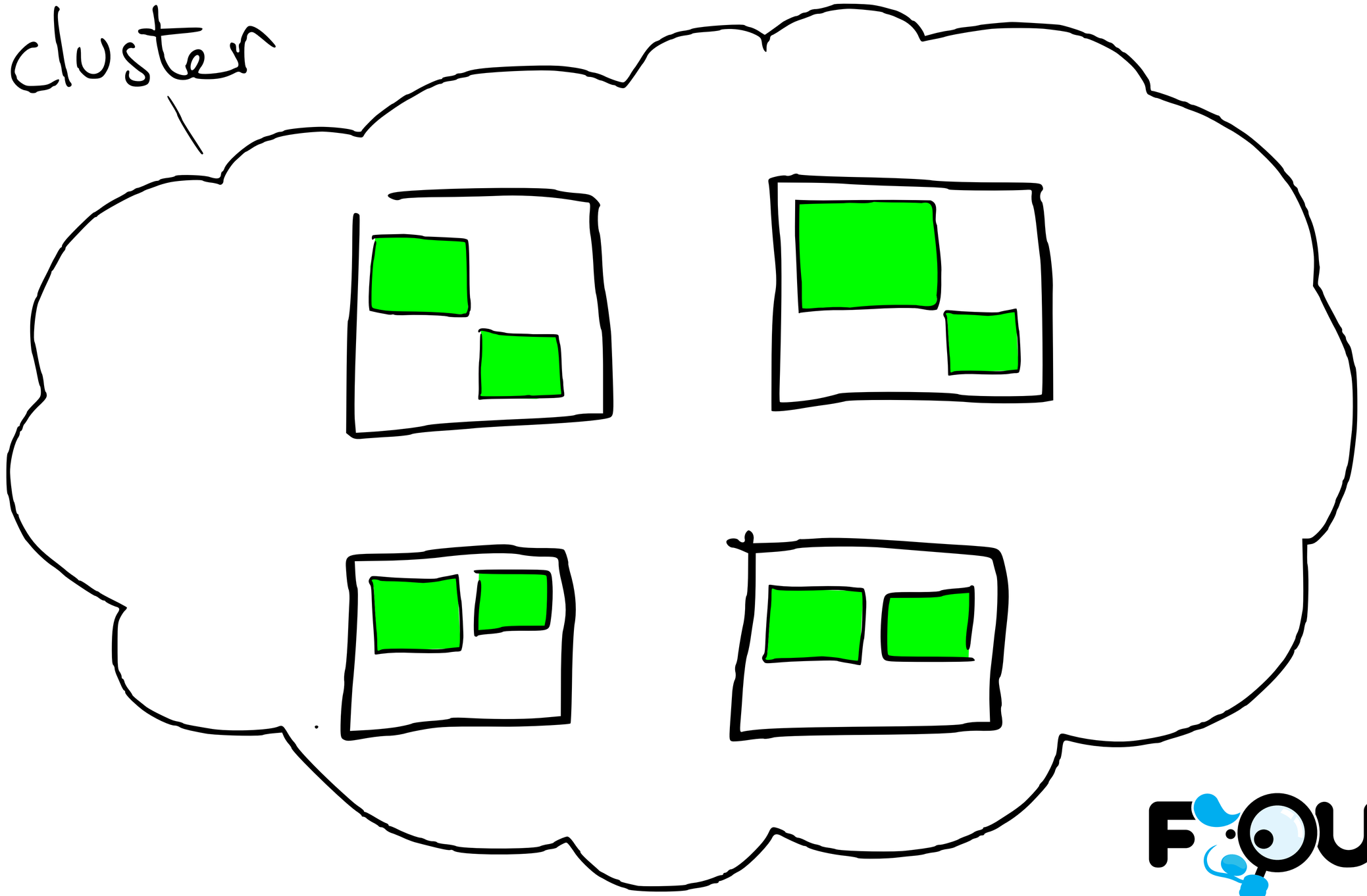
Herding hundreds of Elasticsearch clusters

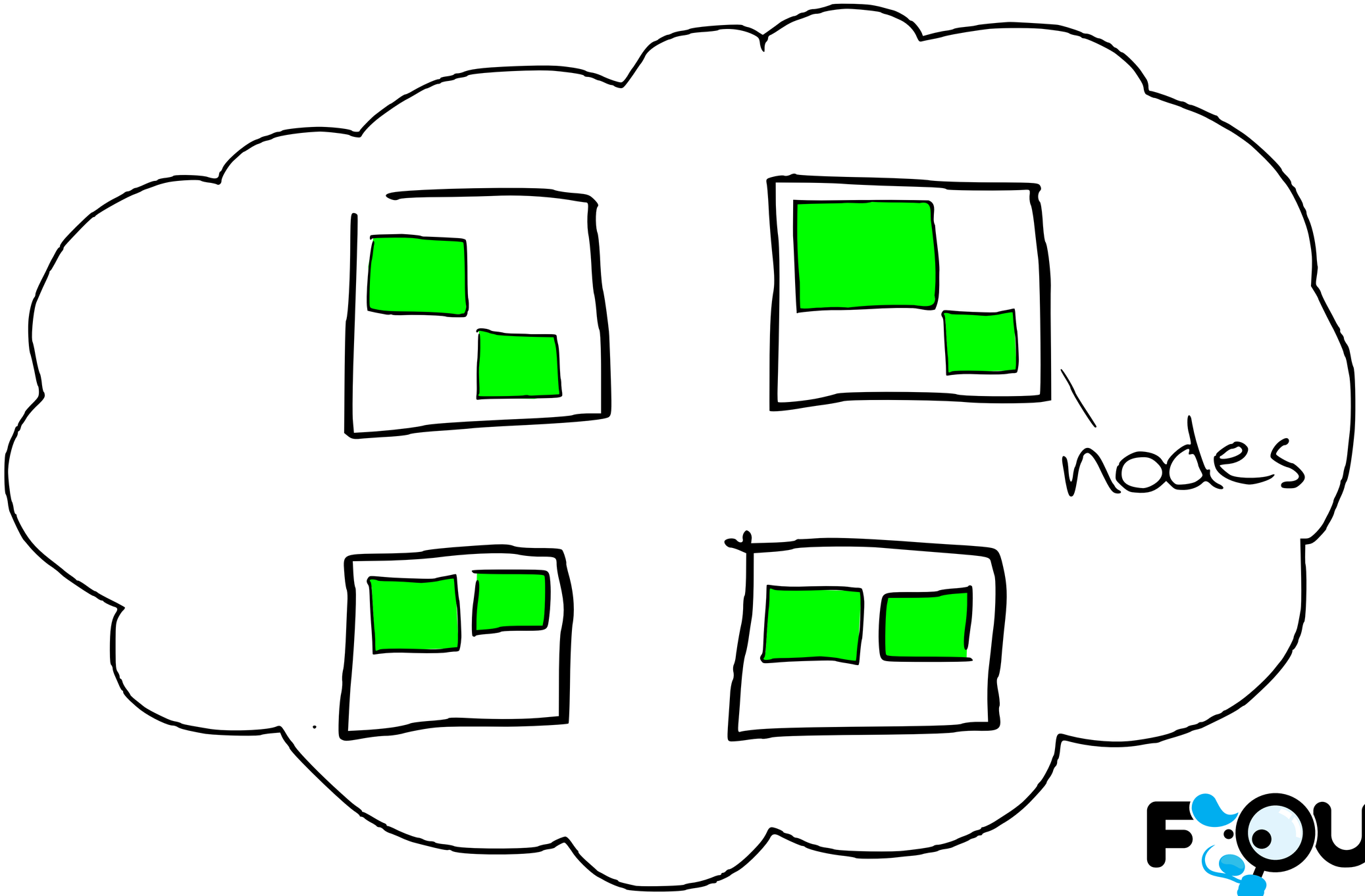


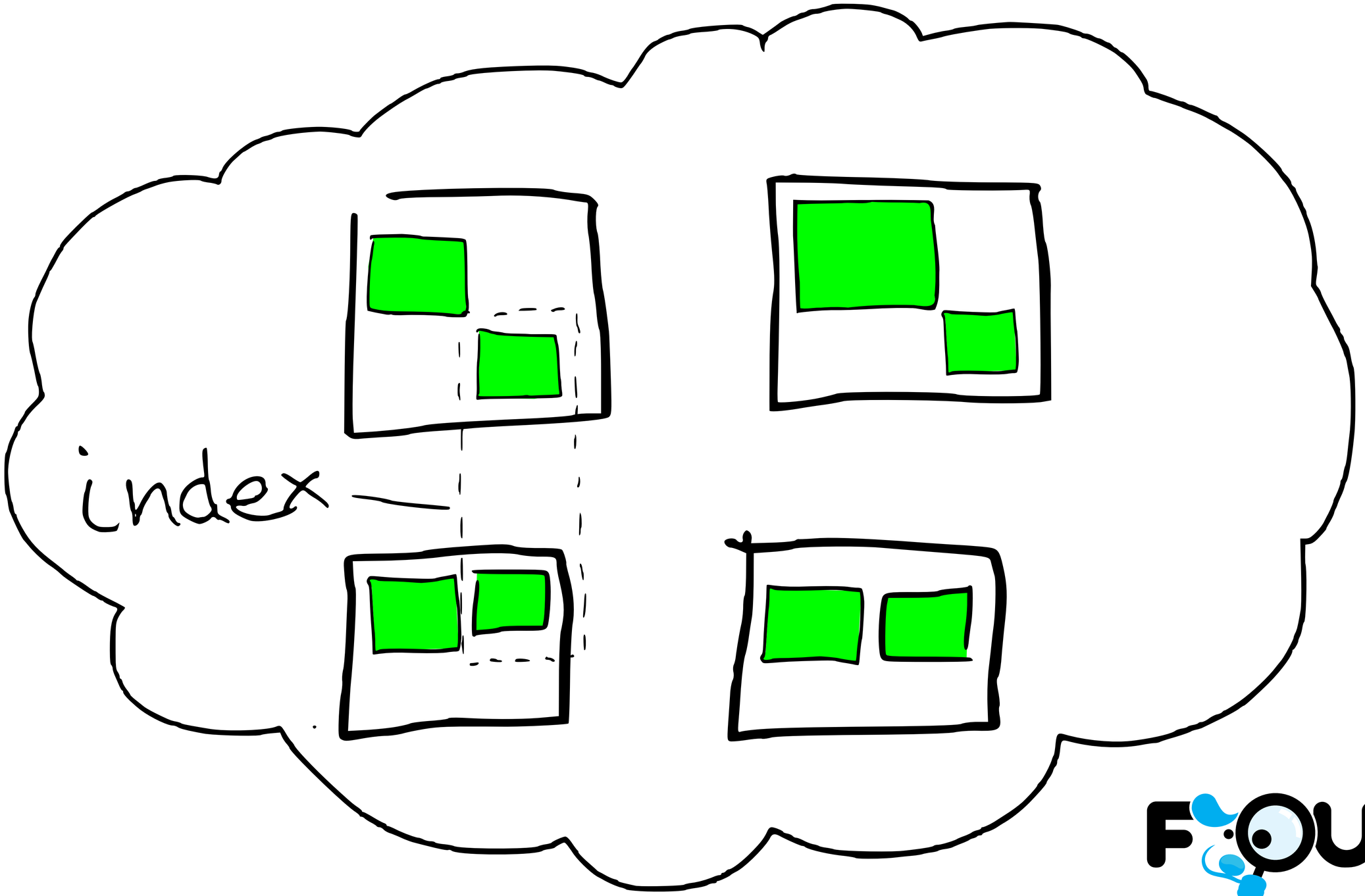
# Motivation

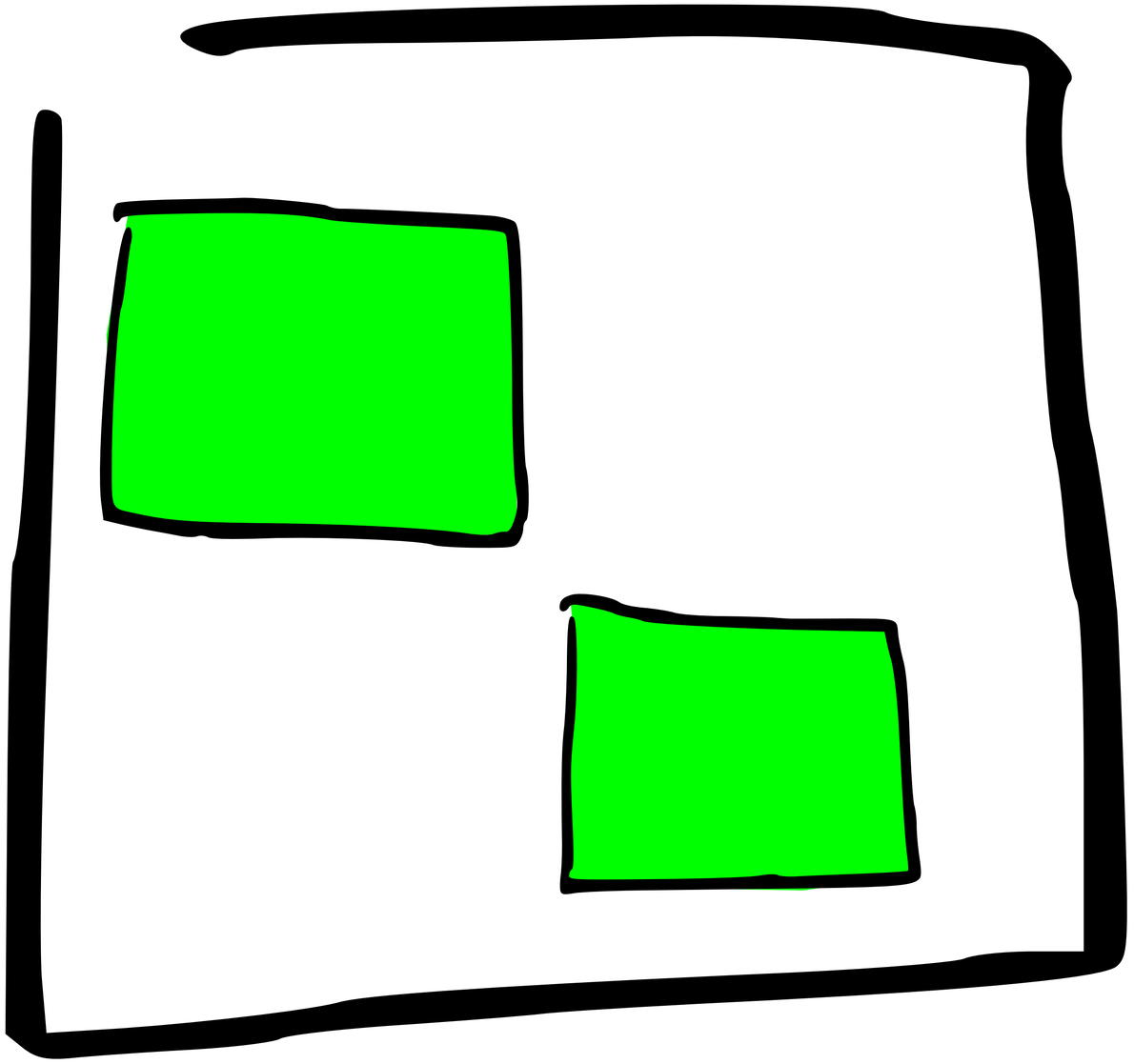
- Why isn't my search for `*foo-bar*` matching “`foo-bar`”?
- Why can adding more documents shrink the index?
- Why is Elasticsearch using so much memory?
- Why can a distributed aggregation be inaccurate?

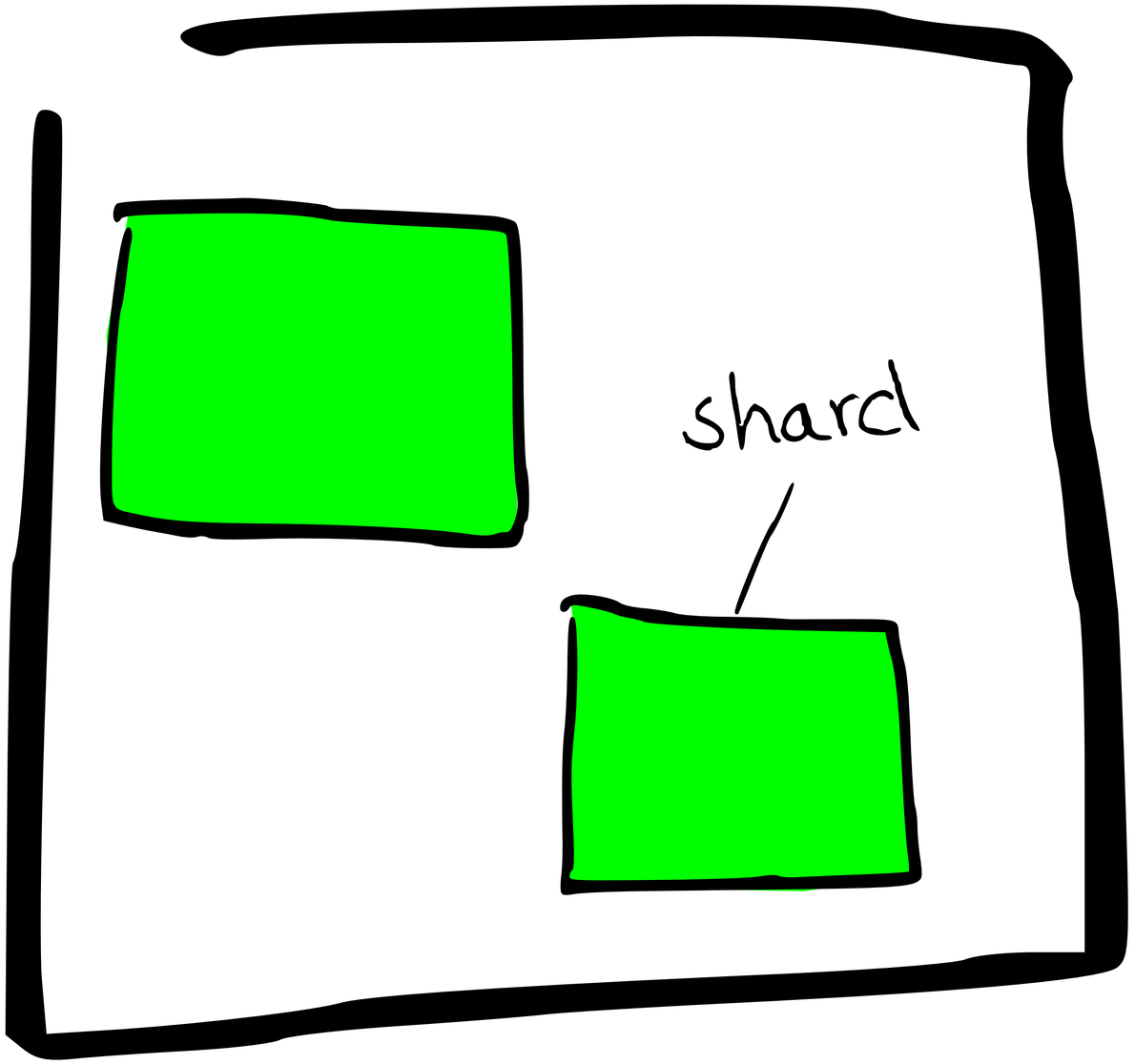


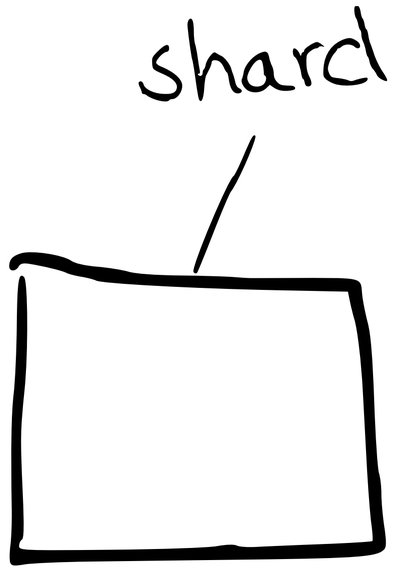

cluster


nodes

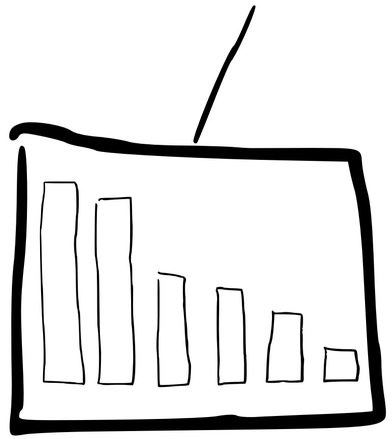

index

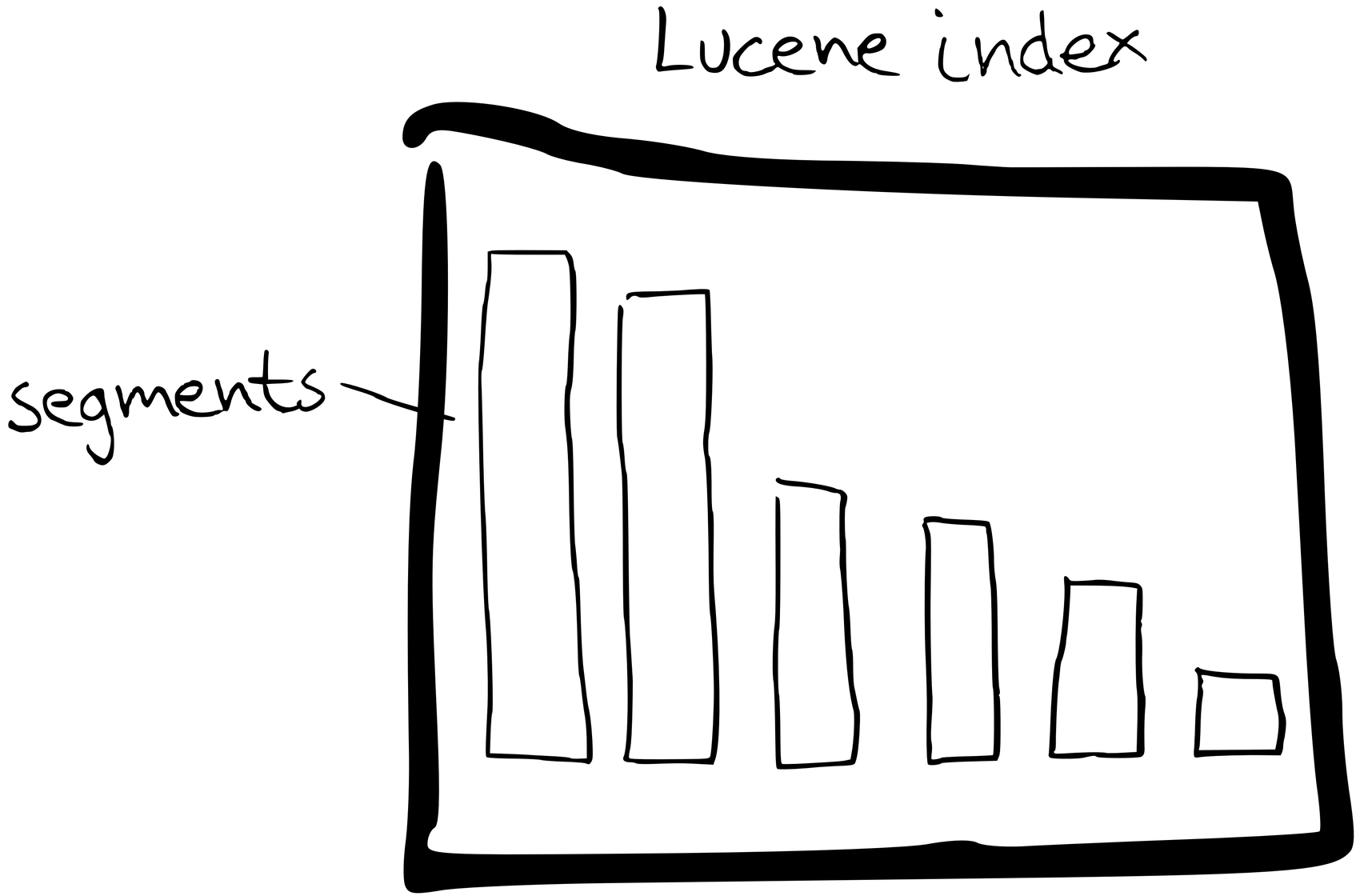


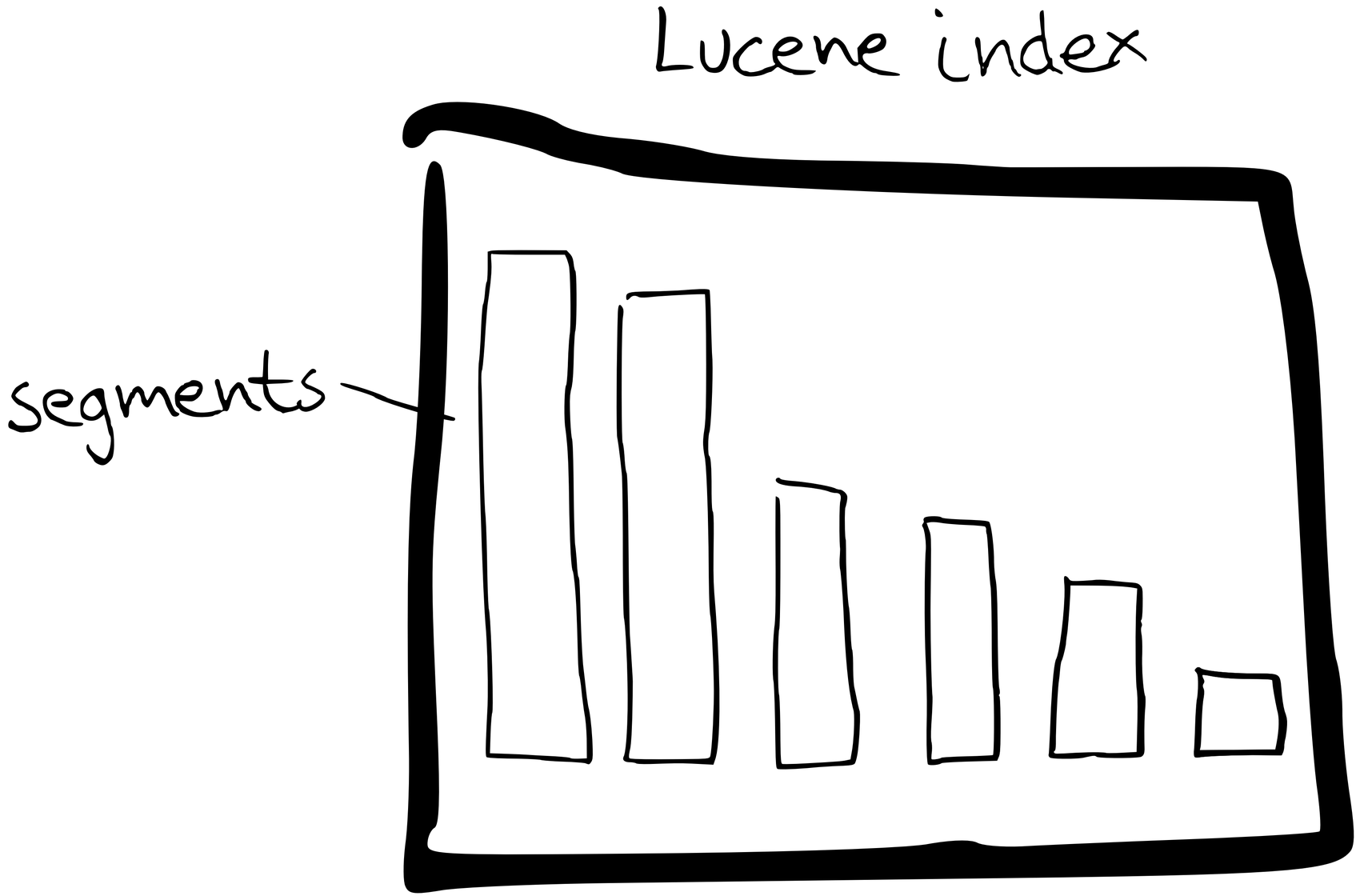

shard

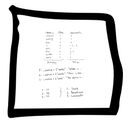

shard

Lucene index




Lucene index
segments


Lucene index
segments



| term | freq | documents |
|---|---|---|
| choice | 1 | 3 |
| coming | 1 | 1 |
| fury | 1 | 2 |
| is | 3 | 1, 2, 3 |
| ours | 1 | 2 |
| the | 2 | 2, 3 |
| winter | 1 | 1 |
| yours | 1 | 3 |

Dictionary | Postings

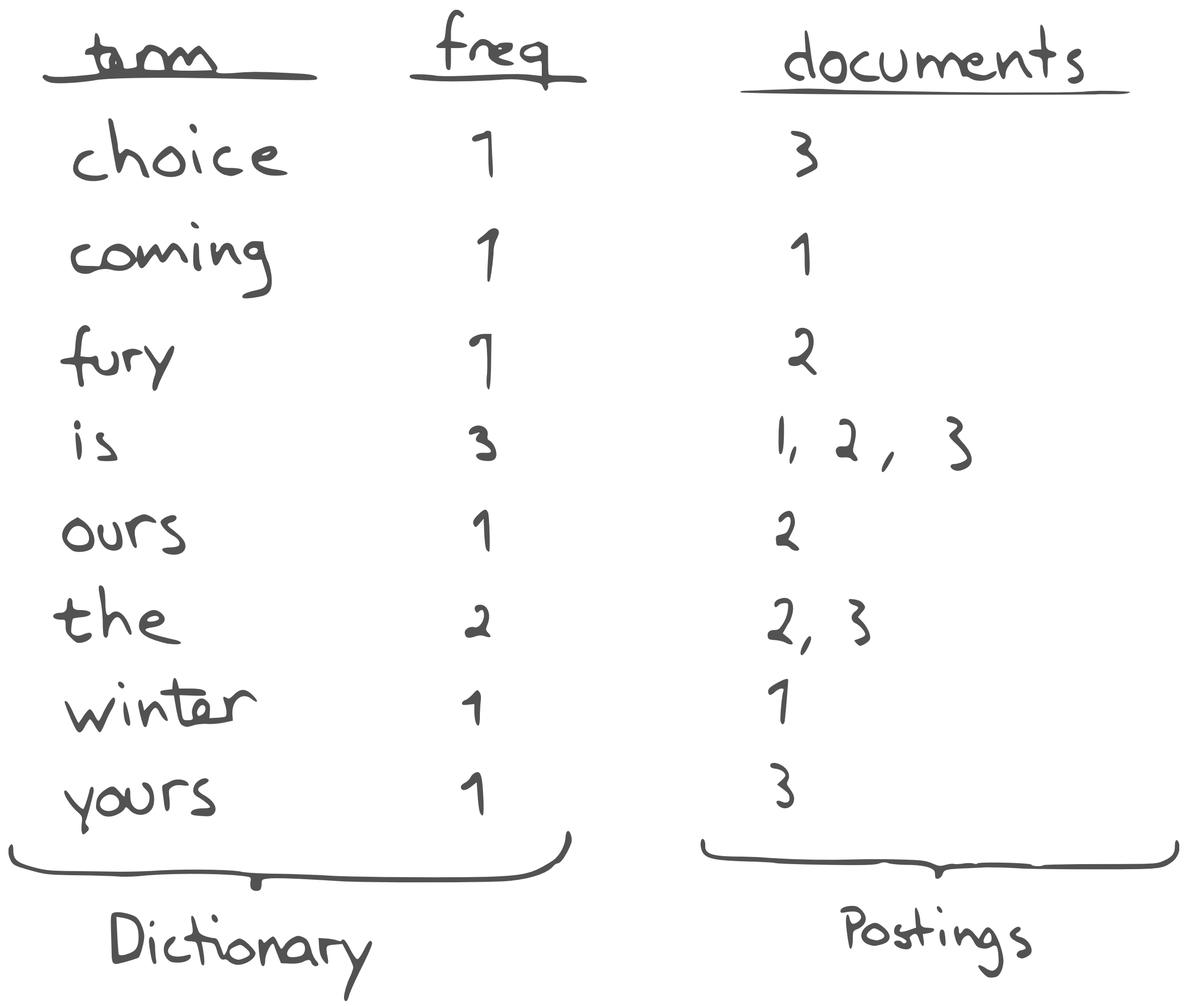
1: _source = {"words": "Winter is ...

2: _source = {"words": "Ours is the ...

3: _source = {"words": "The choice is ...

1: 14
2: 42
3: 33

1: Stark
2: Baratheon
3: Lonmouth


term
freq
documents
choice 1 3
coming 1 1
fury 1 2
is 3 1, 2, 3
ours 1 2
the 2 2, 3
winter 1 1
yours 1 3
Dictionary
Postings

1: Winter is coming.

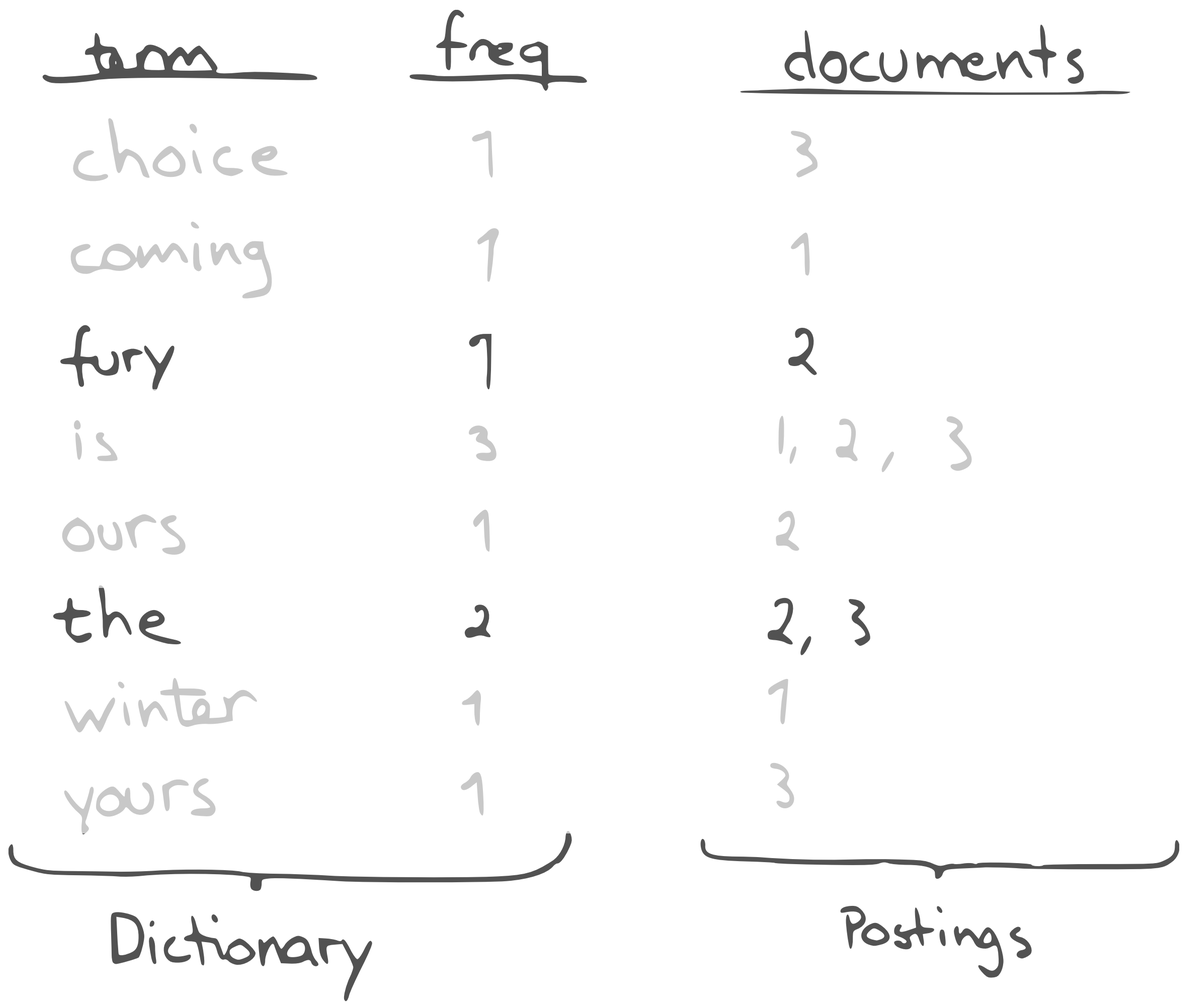
2: Ours is the fury.

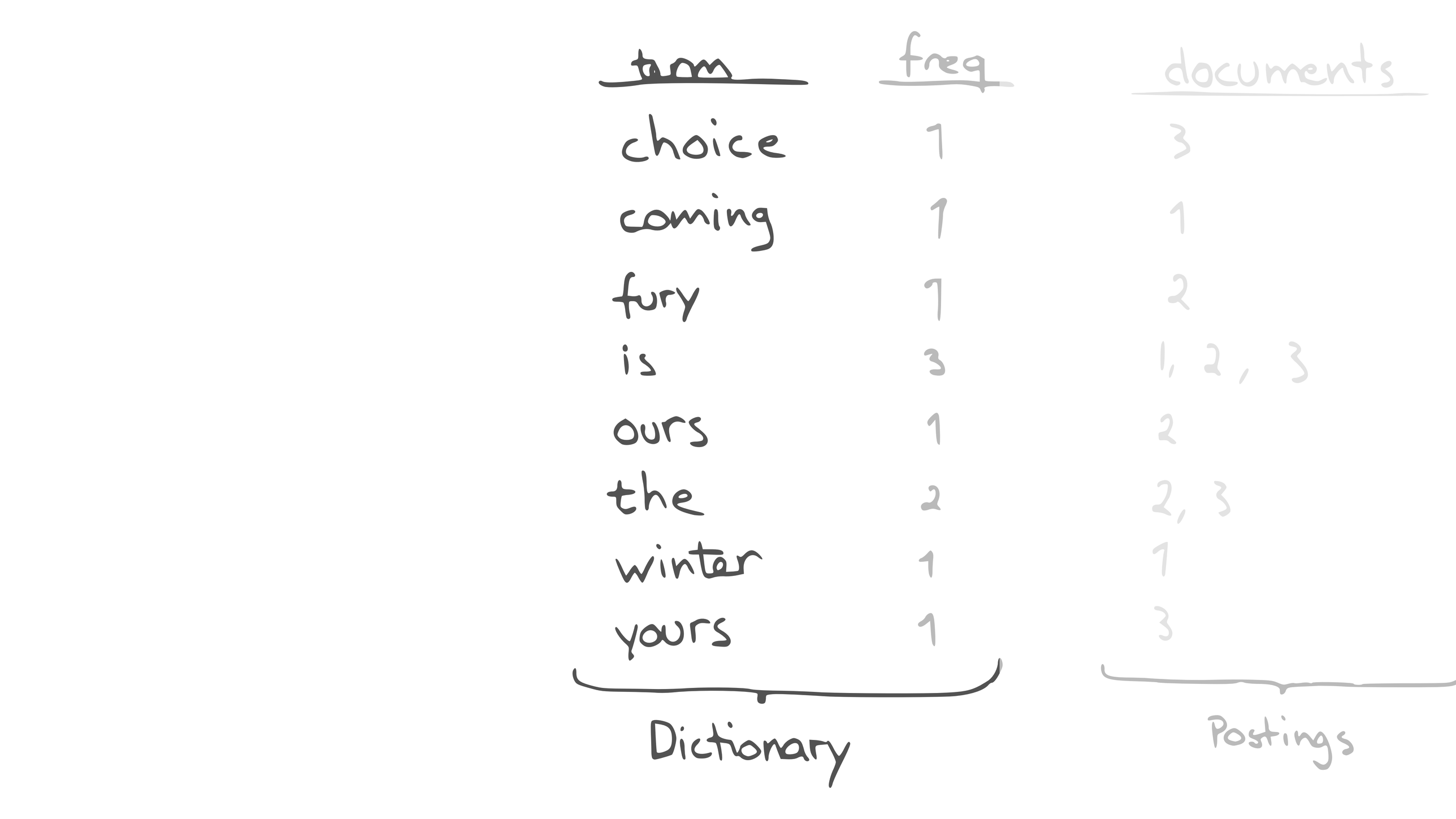
3: The choice is yours.

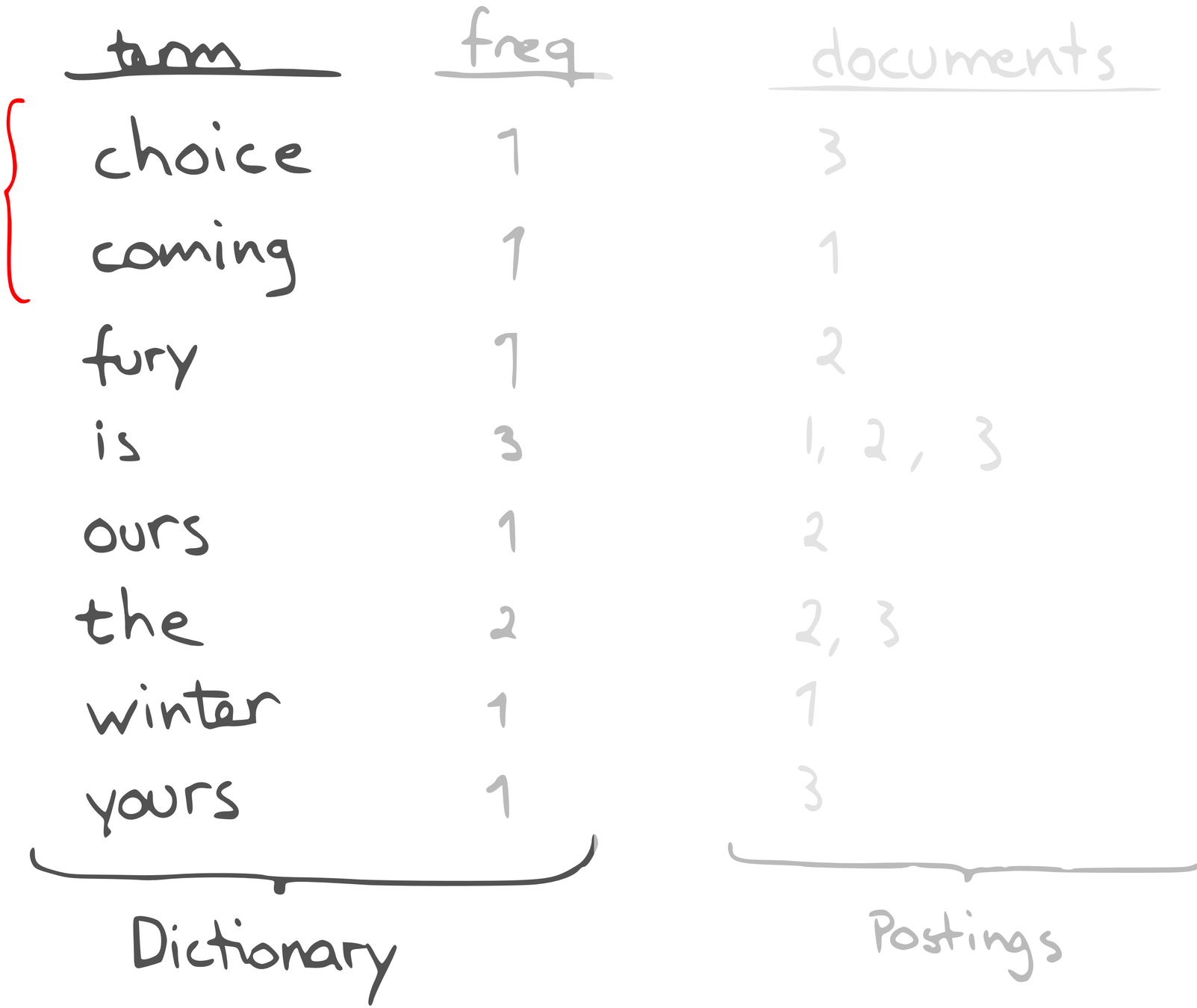
| term | freq | documents |
|---|---|---|
| choice | 1 | 3 |
| coming | 1 | 1 |
| fury | 1 | 2 |
| is | 3 | 1, 2, 3 |
| ours | 1 | 2 |
| the | 2 | 2, 3 |
| winter | 1 | 1 |
| yours | 1 | 3 |

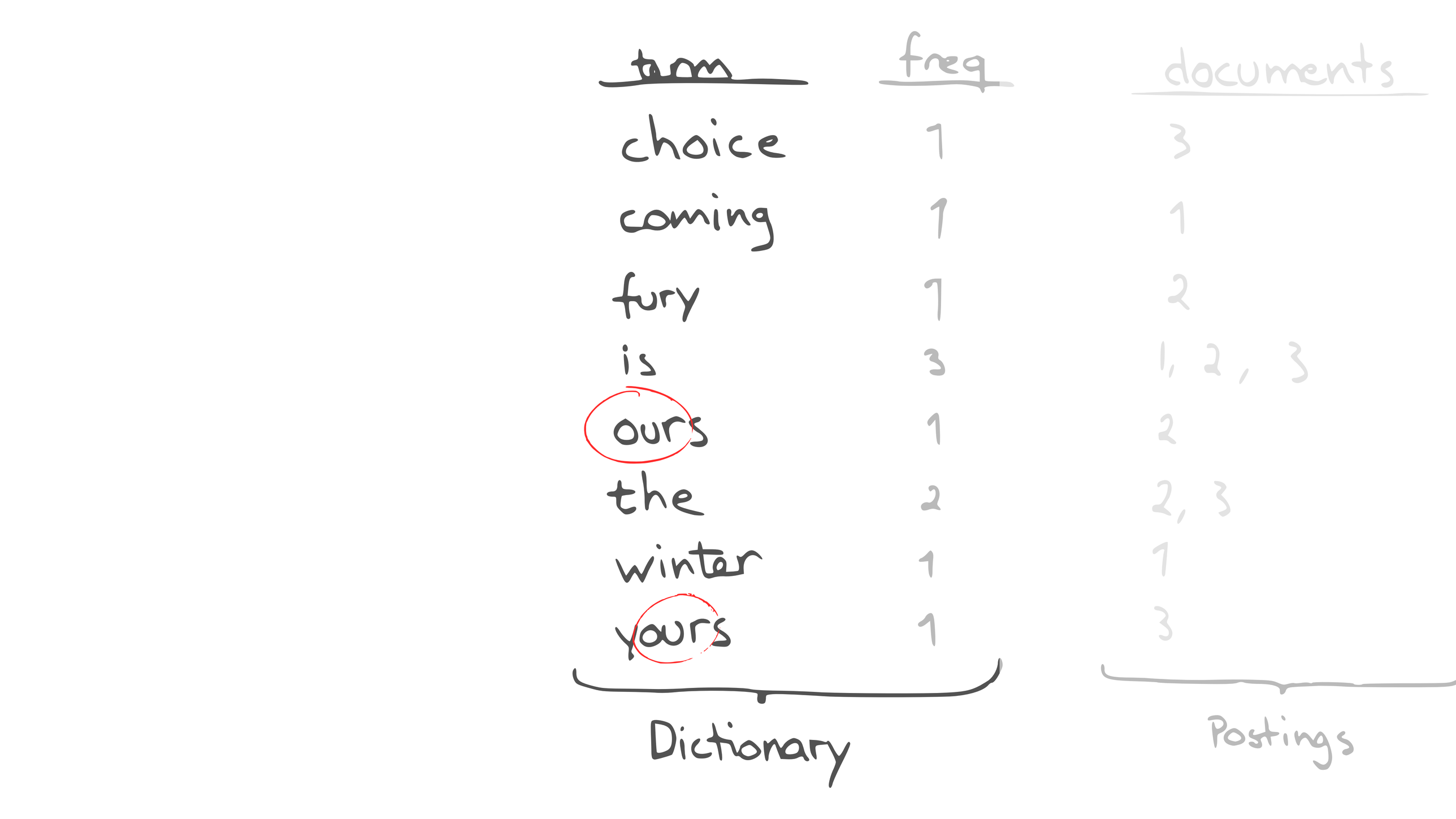
Dictionary

Postings


term
freq
documents
choice 1 3
coming 1 1
fury 1 2
is 3 1, 2, 3
ours 1 2
the 2 2, 3
winter 1 1
yours 1 3
Dictionary
Postings


term
freq
documents
choice
1
3
coming
1
1
fury
1
2
is
3
1, 2, 3
ours
1
2
the
2
2, 3
winter
1
1
yours
1
3
Dictionary
Postings

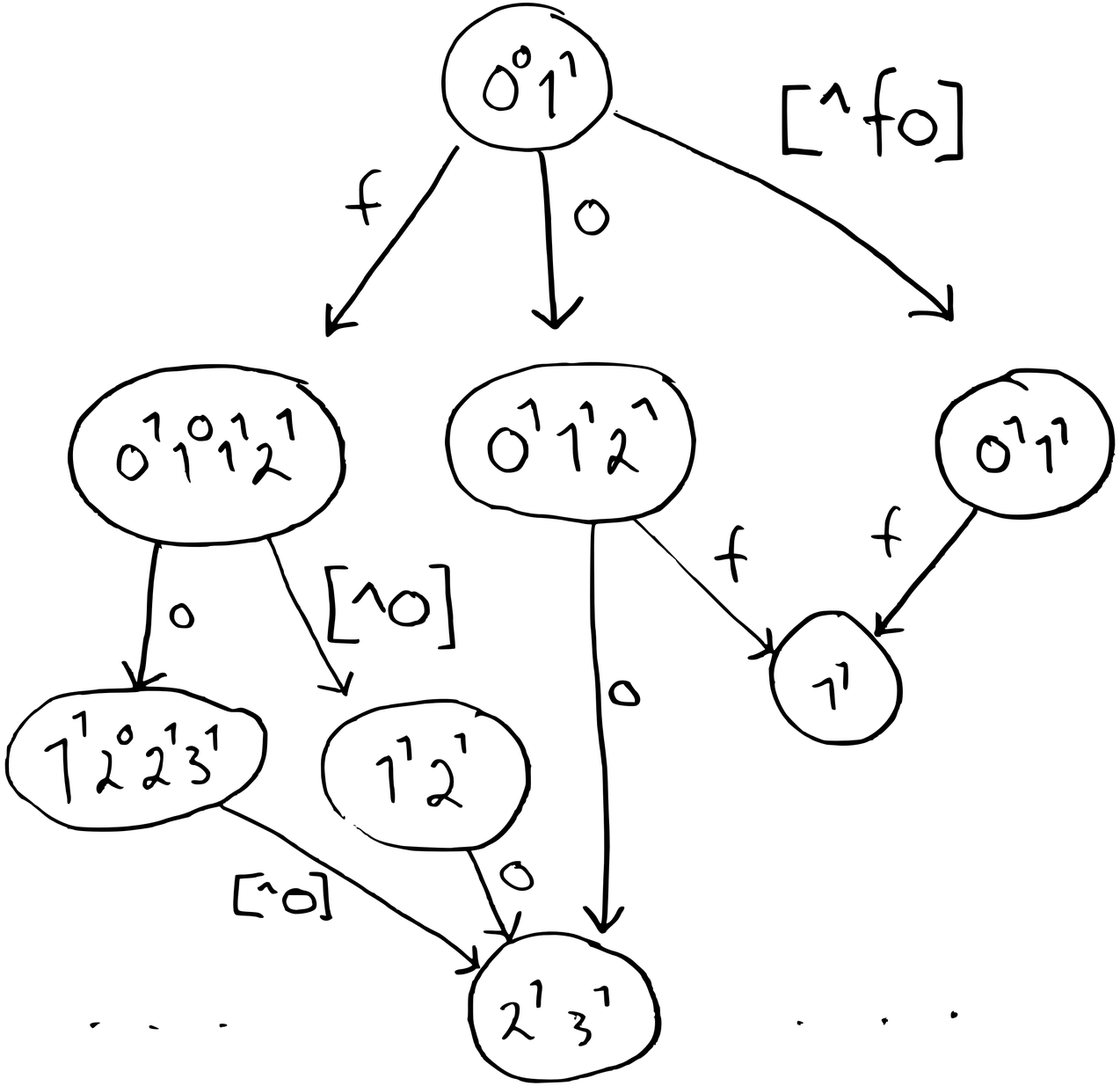

term
freq
documents
choice
1
3
coming
1
1
fury
1
2
is
3
1, 2, 3
ours
1
2
the
2
2, 3
winter
1
1
yours
1
3
Dictionary
Postings


term
freq
documents
choice
1
3
coming
1
1
fury
1
2
is
3
1, 2, 3
ours
1
2
the
2
2, 3
winter
1
1
yours
1
3
Dictionary
Postings

*suffix → xiffus*

(60.6384, 6.5017) → u4u8gyykk

123 → {1-hundreds, 12-tens, 123} (approx.)


0°1¹
[^fo]
f
o
0¹1°1¹2¹
0¹1¹2¹
0¹1¹
o
[^o]
f
f
1¹2°2¹3¹
1¹2¹
1¹
o
[^o]
o
2¹3¹

| term | freq | documents |
|---|---|---|
| choice | 1 | 3 |
| coming | 1 | 1 |
| fury | 1 | 2 |
| is | 3 | 1, 2, 3 |
| ours | 1 | 2 |
| the | 2 | 2, 3 |
| winter | 1 | 1 |
| yours | 1 | 3 |

Dictionary

Postings

# Stored Fields

1: _source = {"words": "Winter is ...

2: _source = {"words": "Ours is the ...

3: _source = {"words": "The choice is ...

# Document Values ( Field Cache )

1 : 14
2 : 42
3 : 33

1 : Stark
2 : Baratheon
3 : Lonmouth

| term | freq | documents |
|---|---|---|
| choice | 1 | 3 |
| coming | 1 | 1 |
| fury | 1 | 2 |
| is | 3 | 1, 2, 3 |
| ours | 1 | 2 |
| the | 2 | 2, 3 |
| winter | 1 | 1 |
| yours | 1 | 3 |

Dictionary | Postings

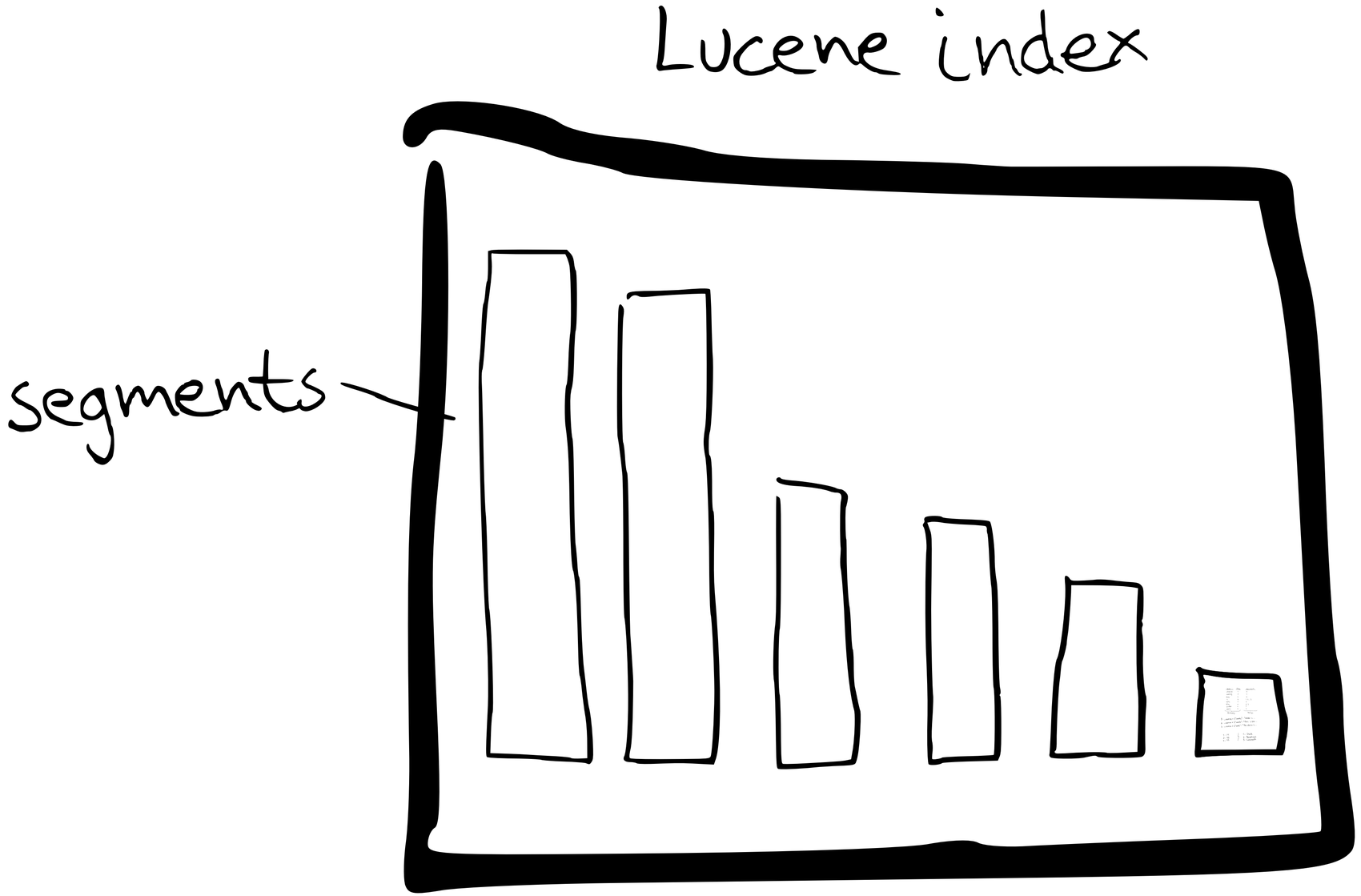
1: _source = {"words": "Winter is ...

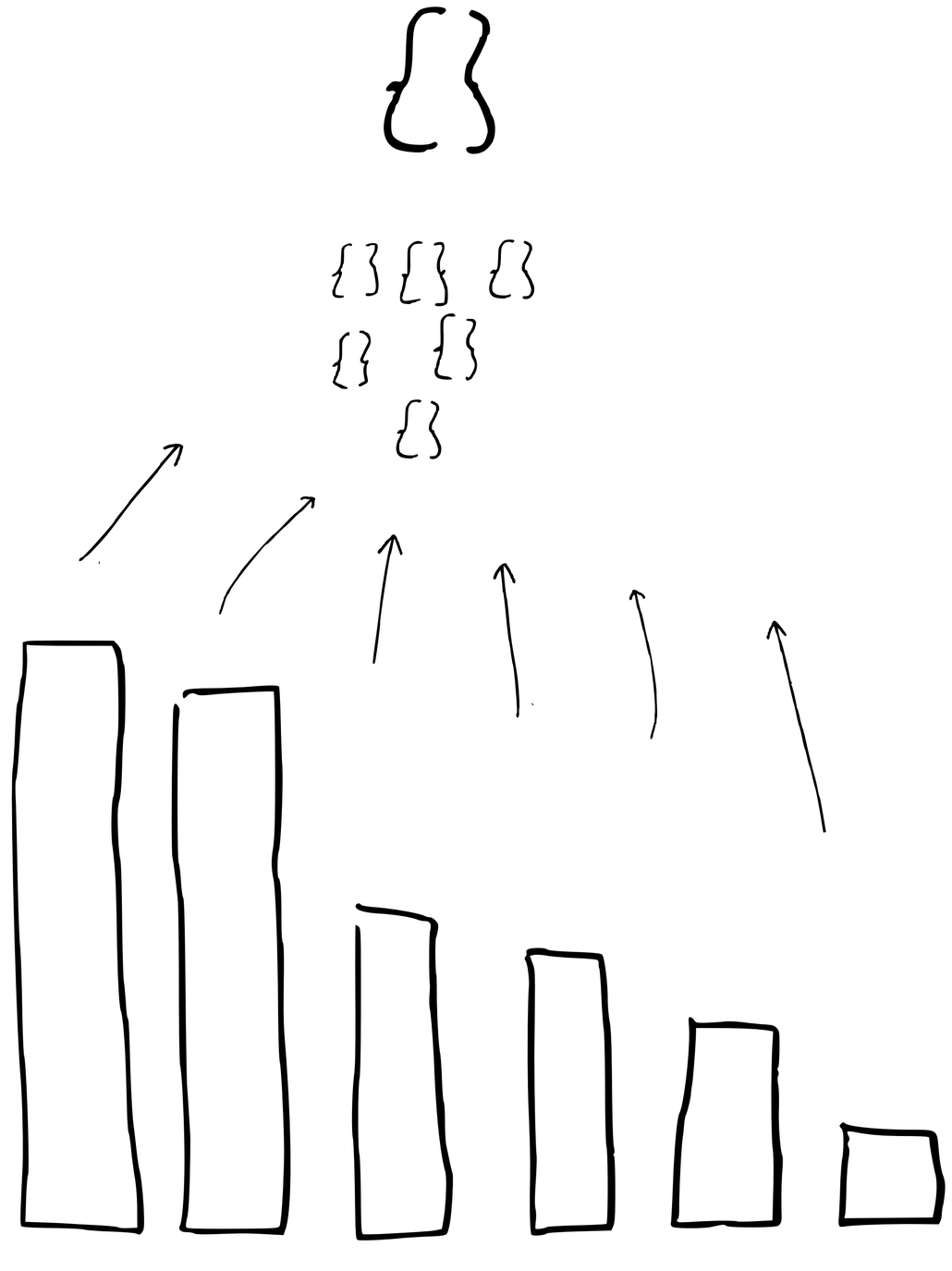
2: _source = {"words": "Ours is the ...

3: _source = {"words": "The choice is ...

1: 14
2: 42
3: 33

1: Stark
2: Baratheon
3: Lonmouth


Lucene index
segments



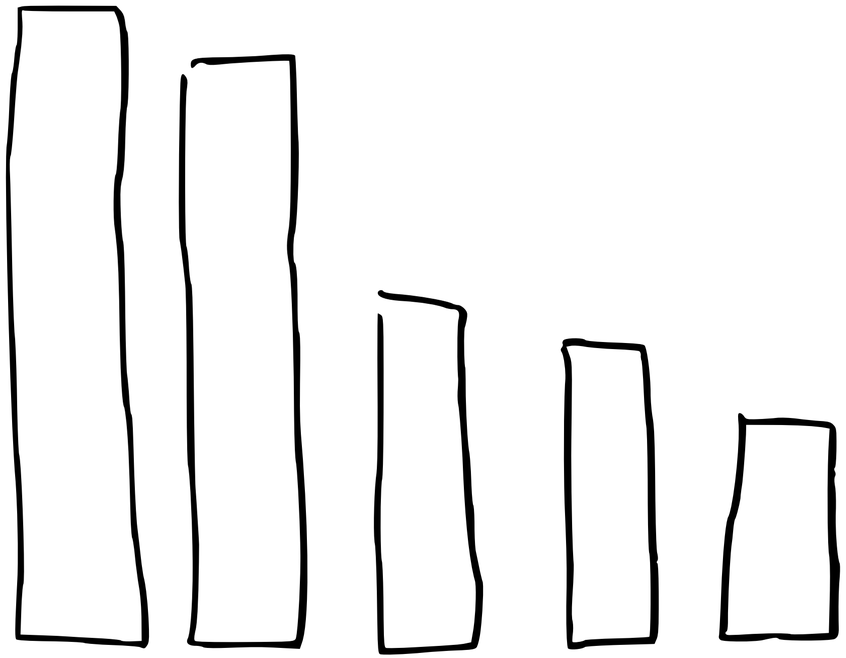
# Segments are immutable

# Deletes?

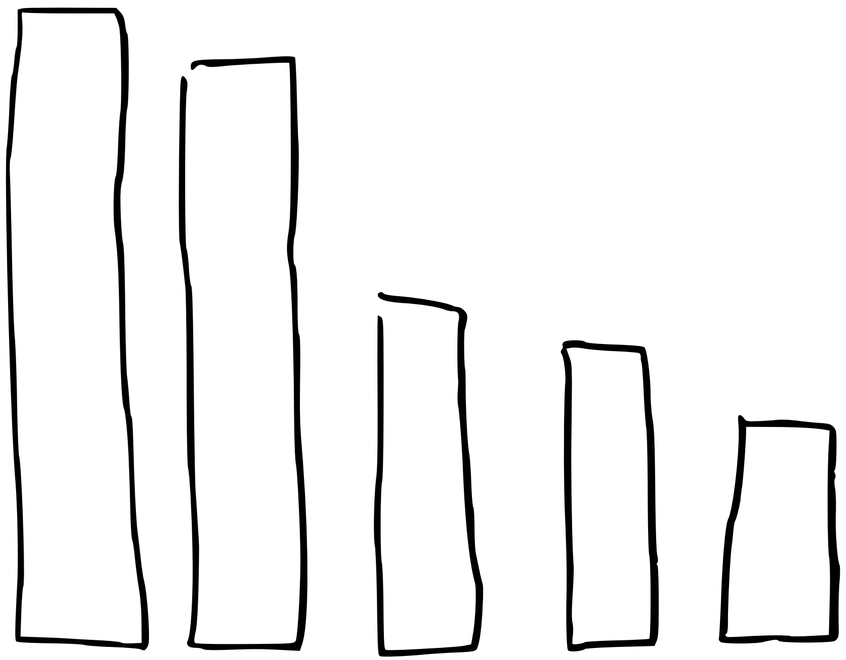
# *Compress* all the things!

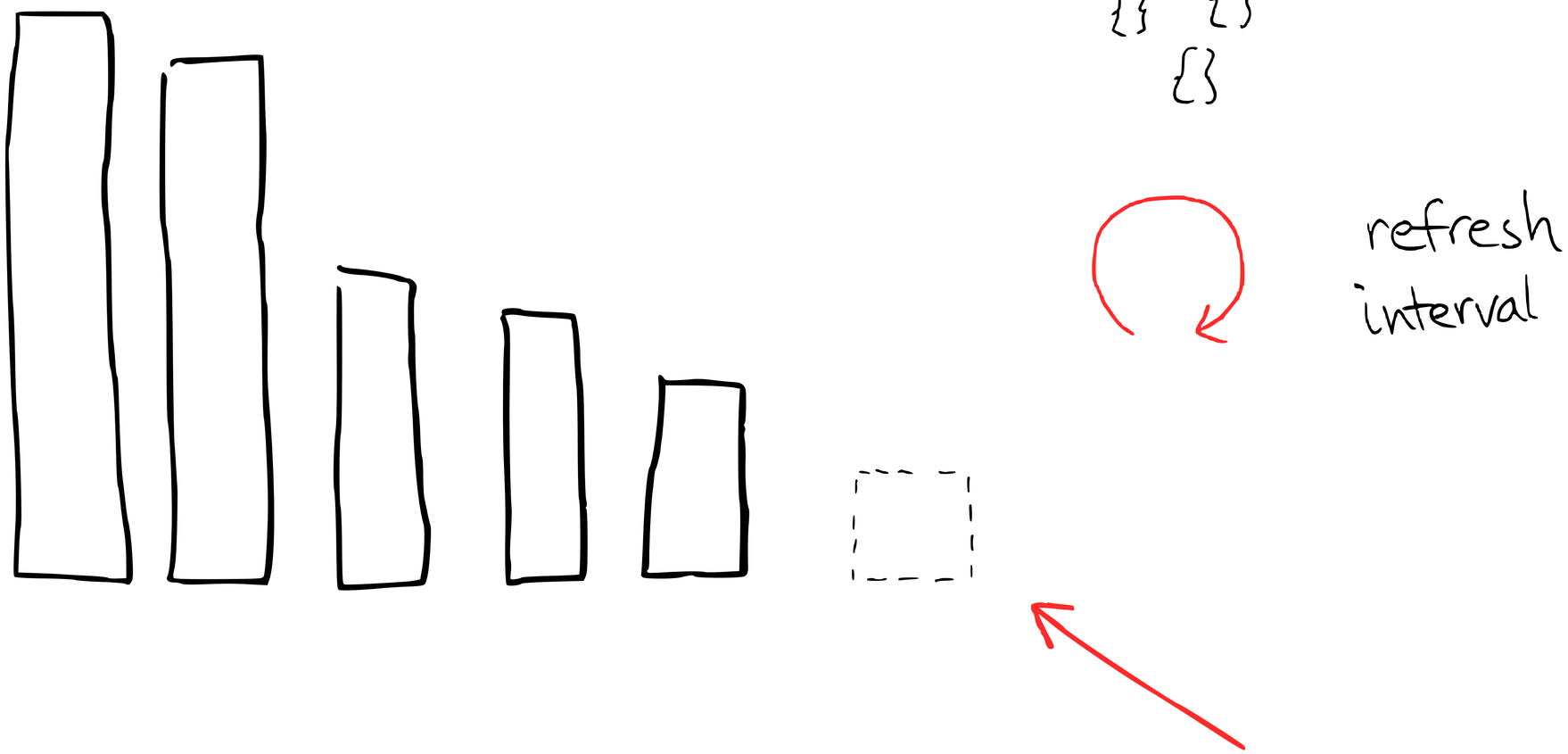
# *Cache* all the things!

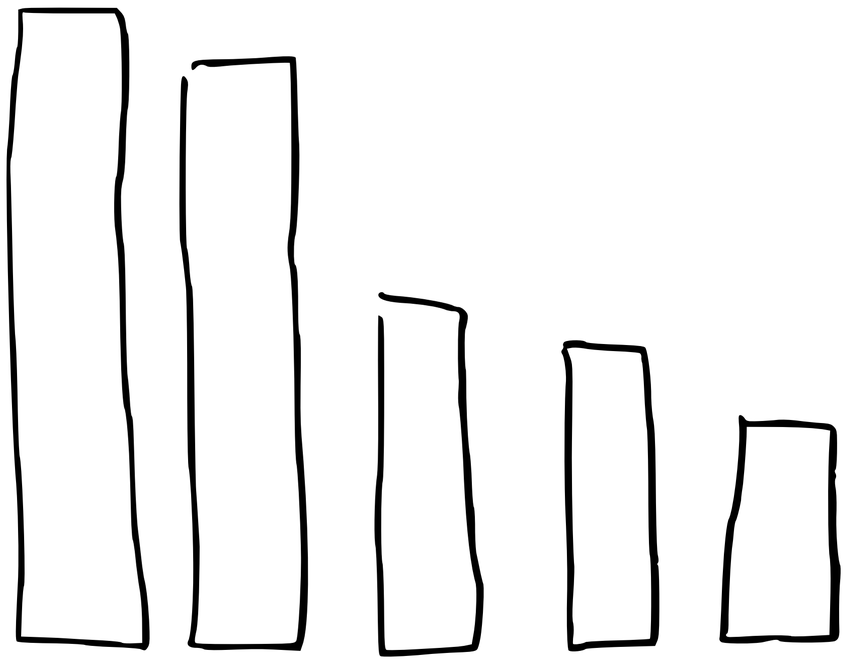
buffer

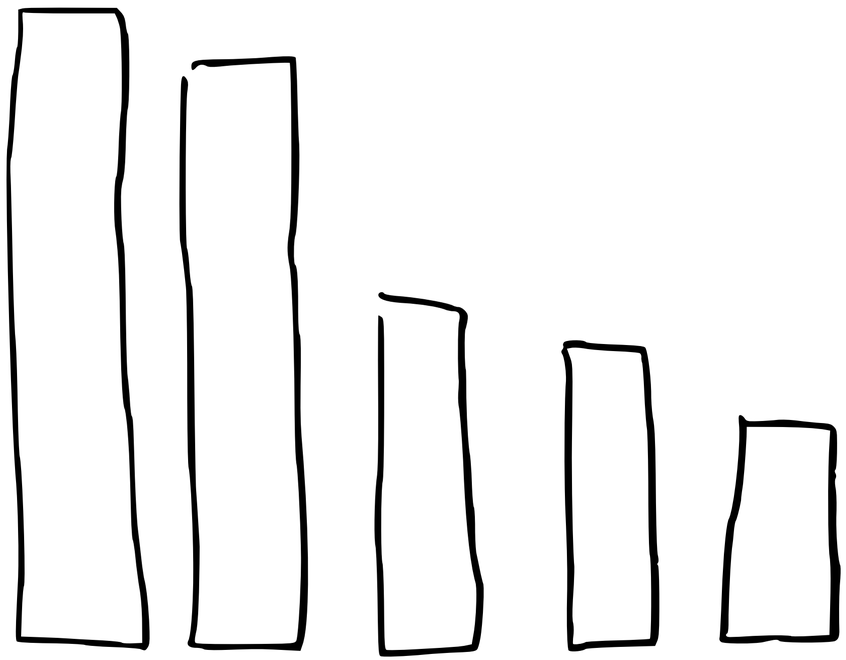


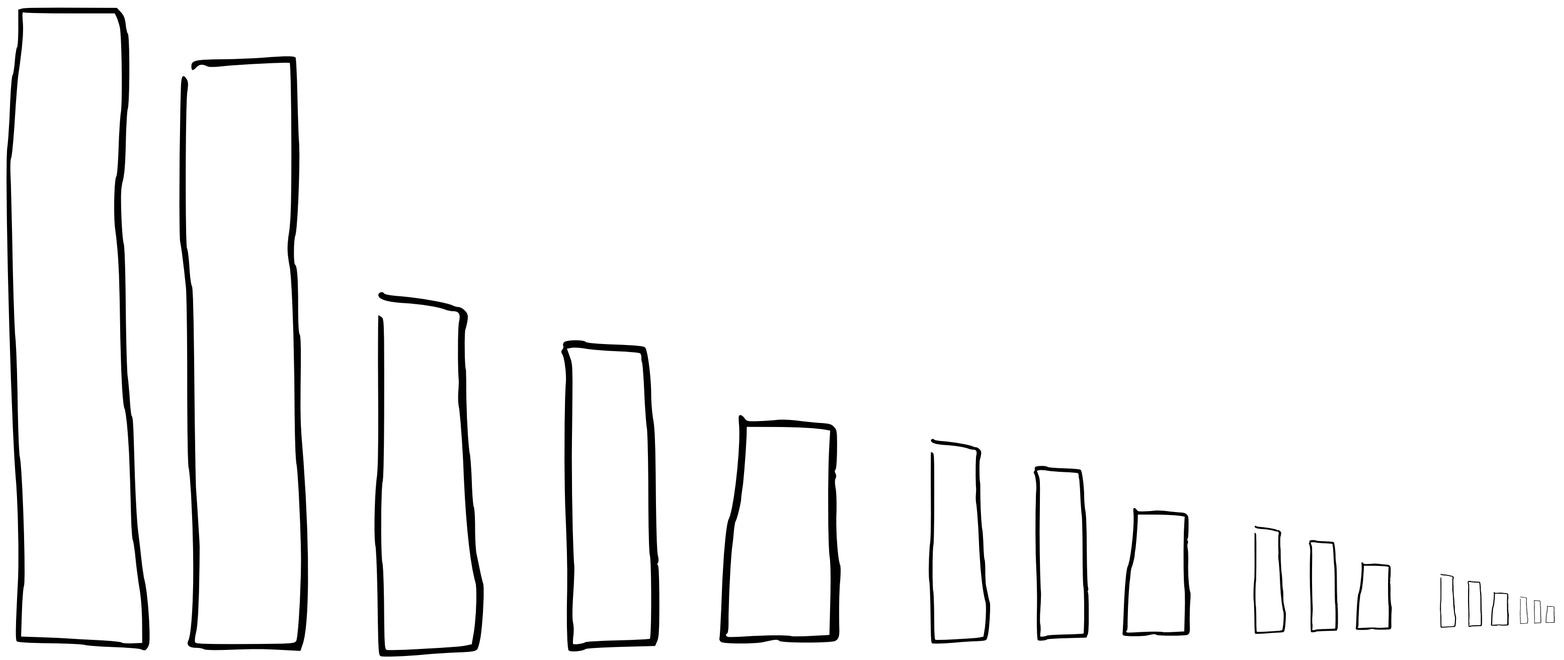
buffer

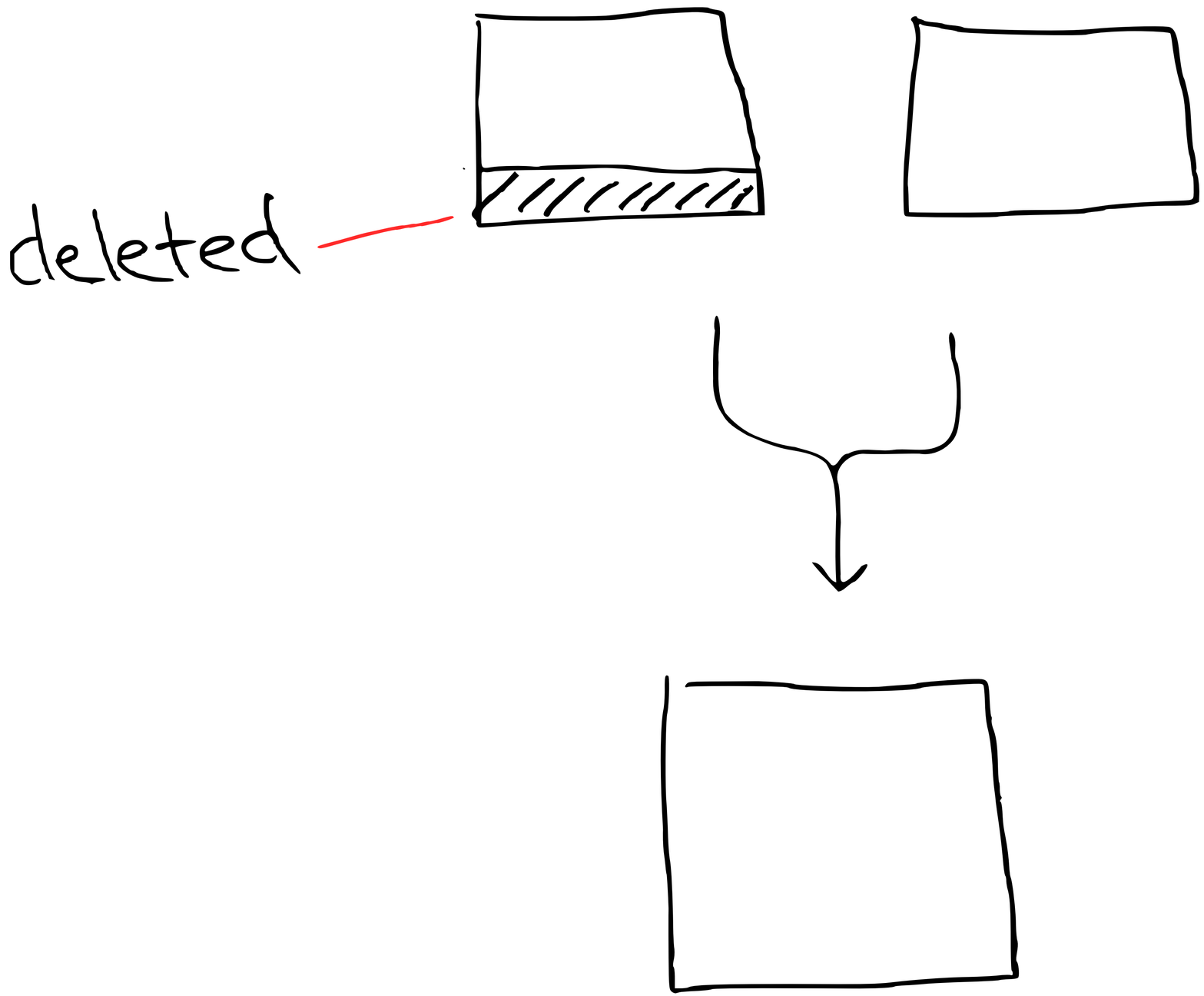


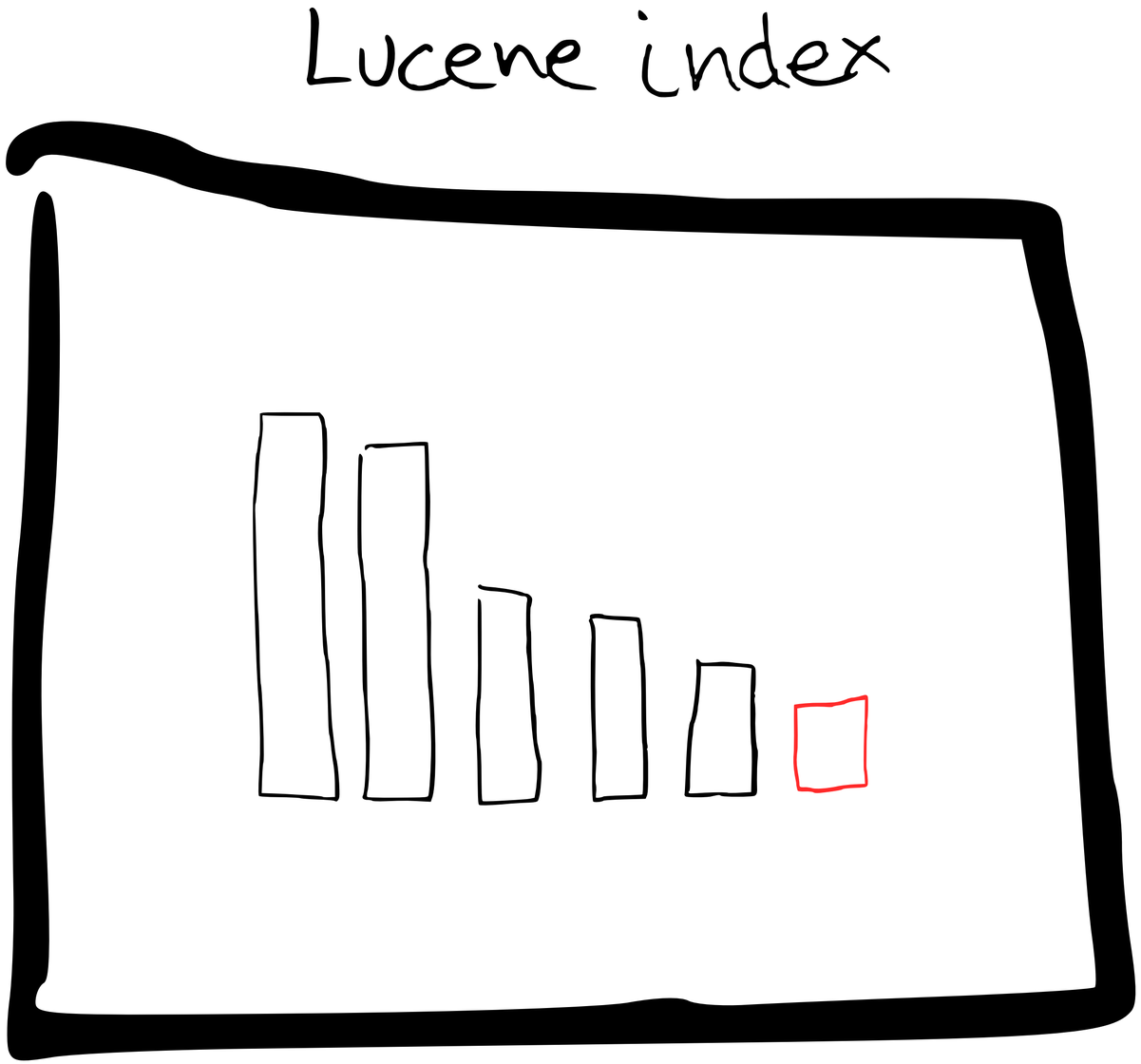

buffer
{} {} {}
{} {}
{}
refresh
interval

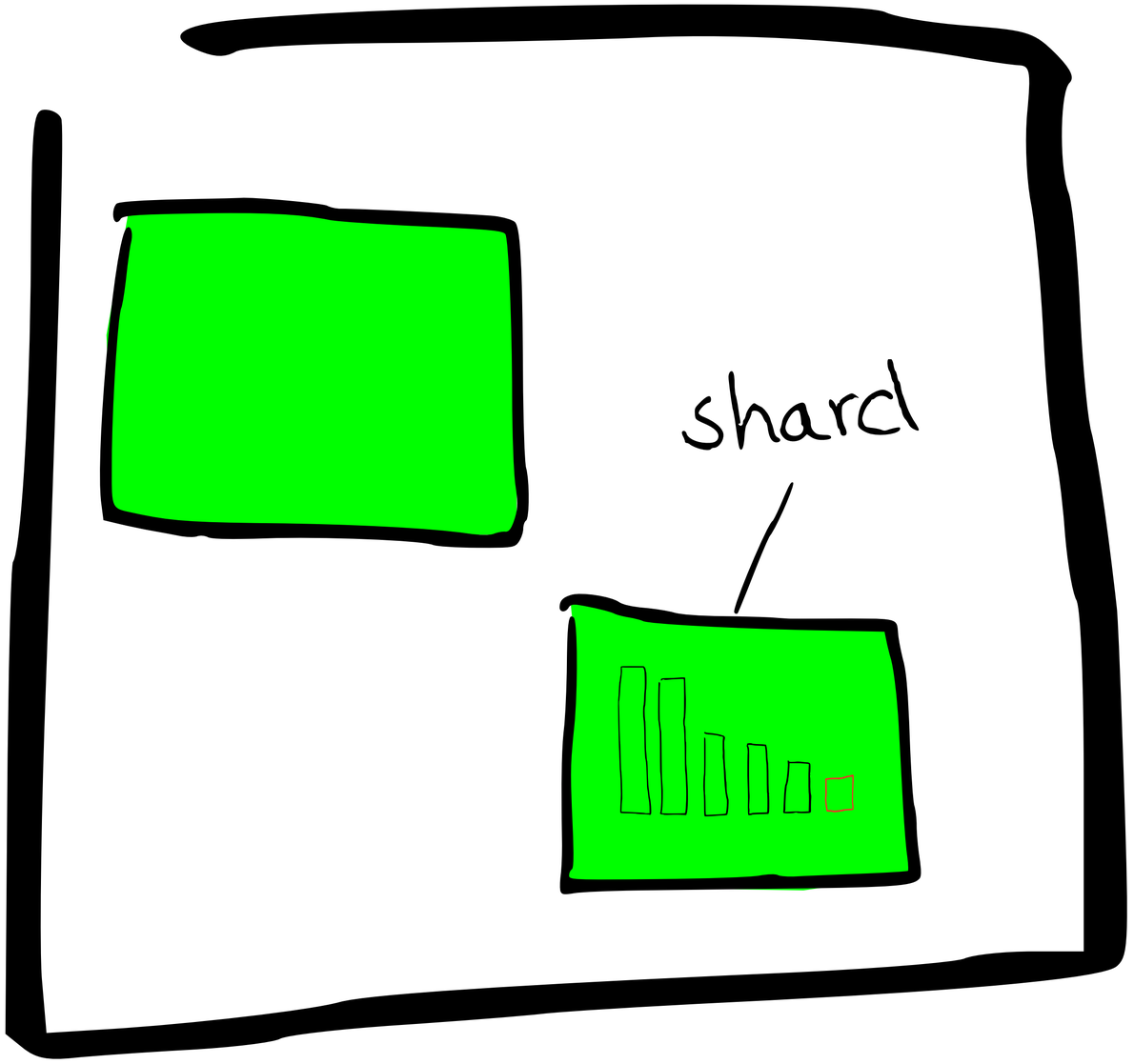


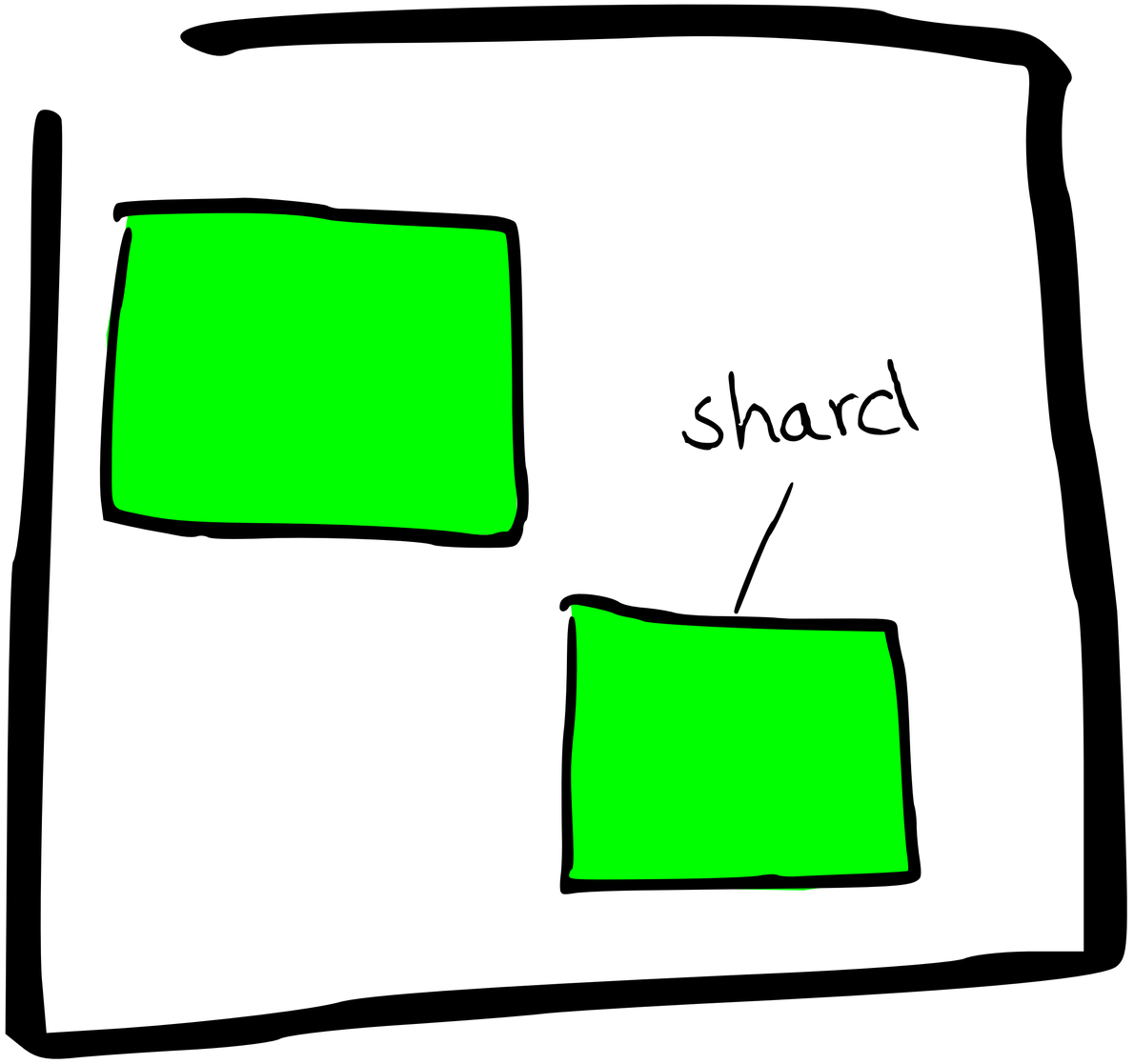


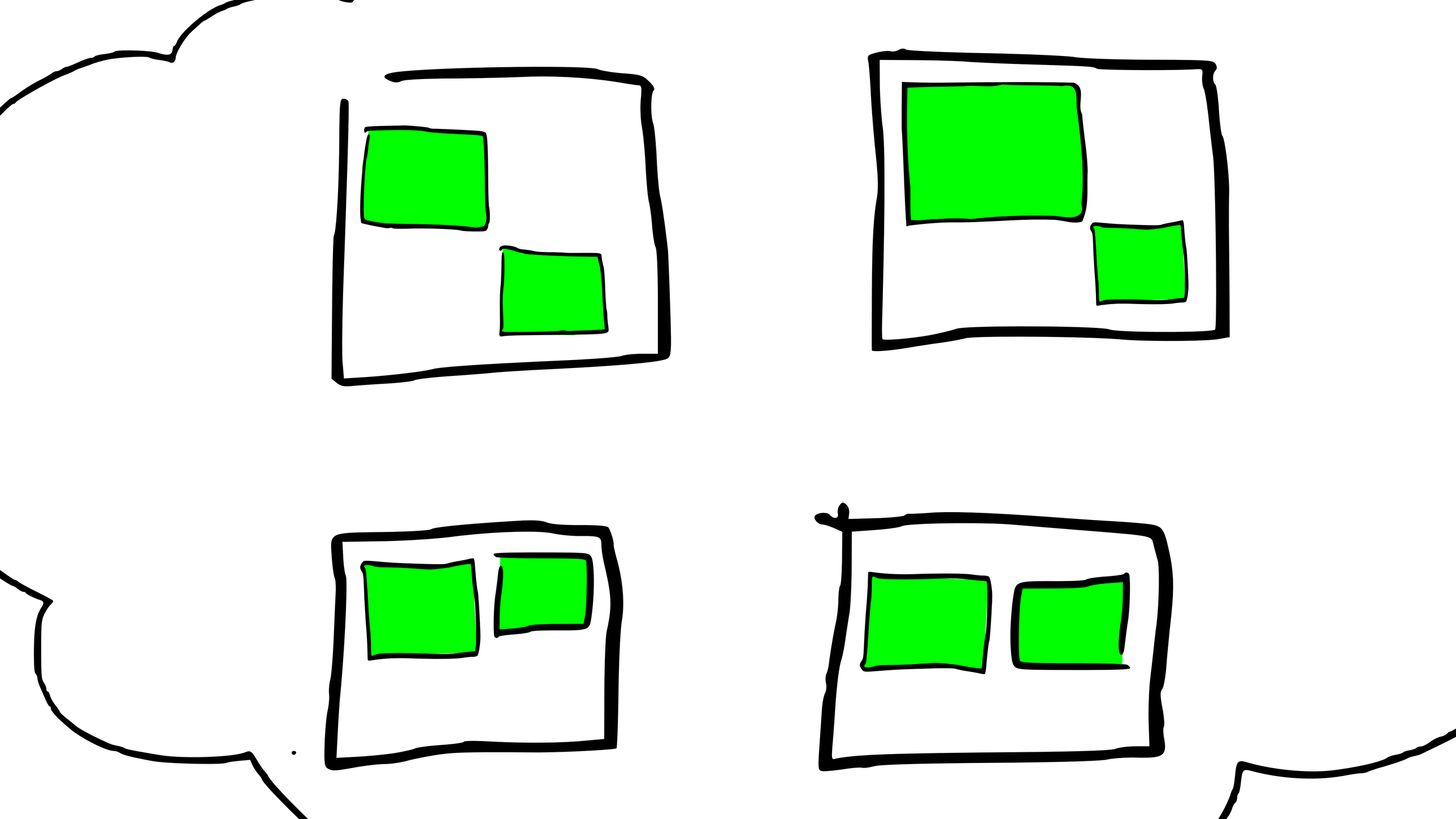


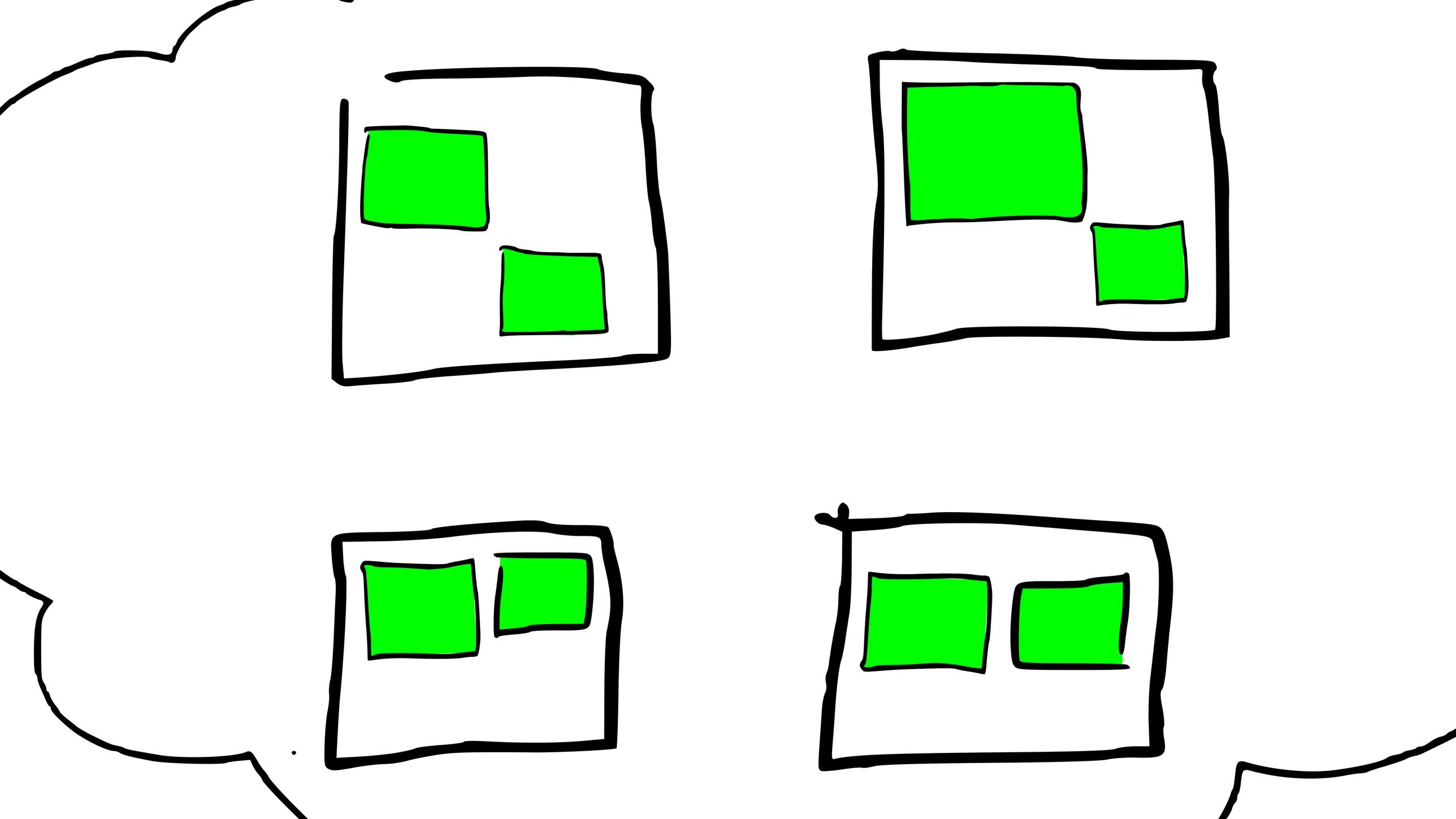

deleted

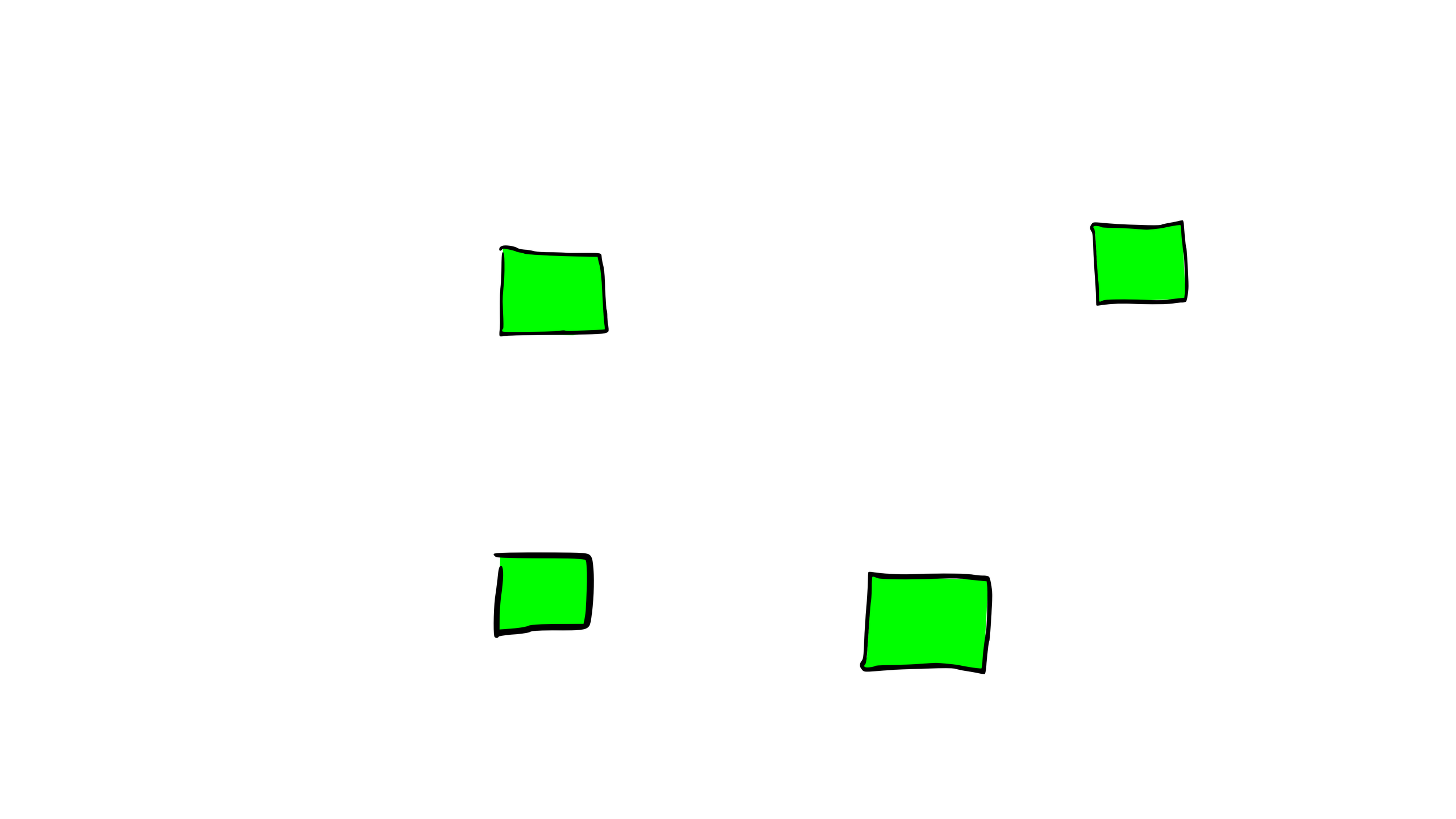

Lucene index

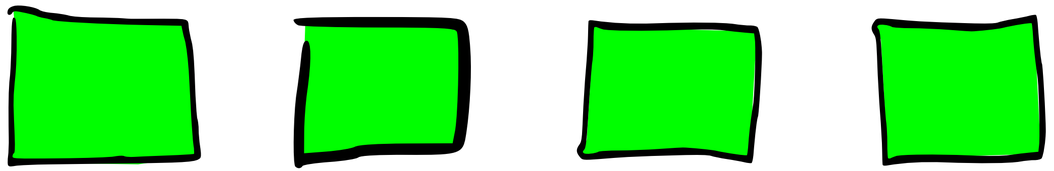

shard

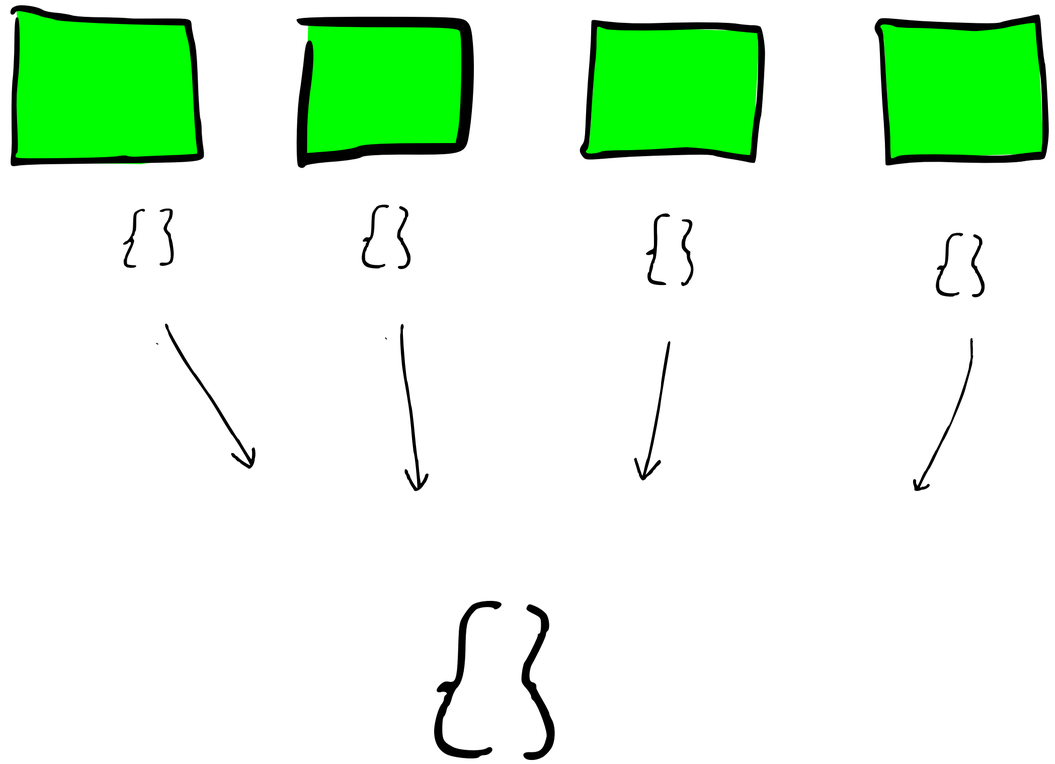

shard

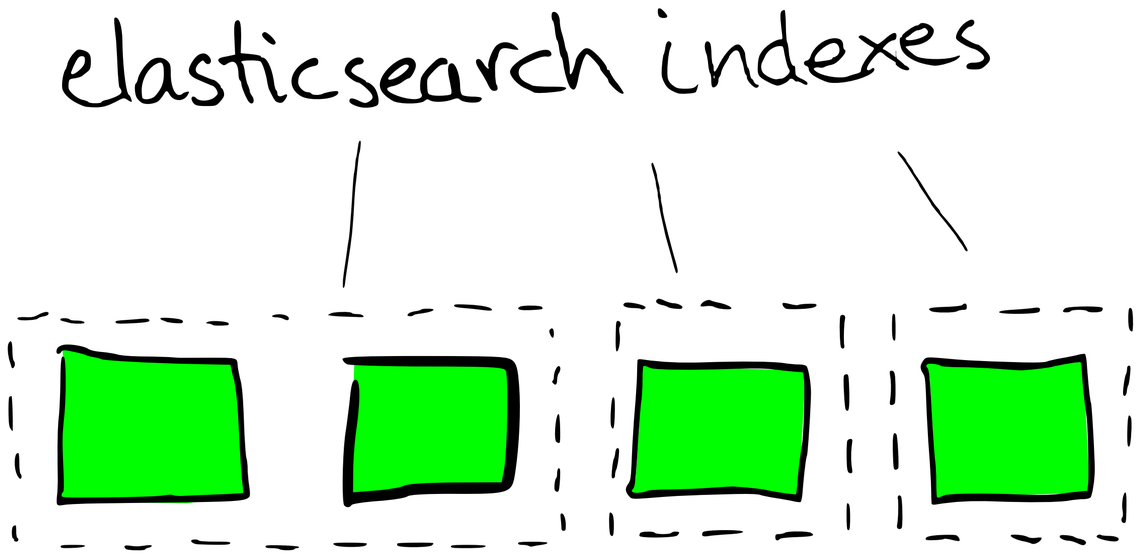




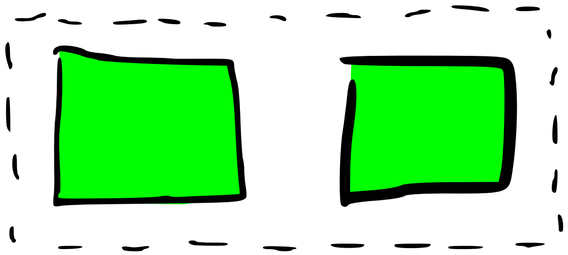




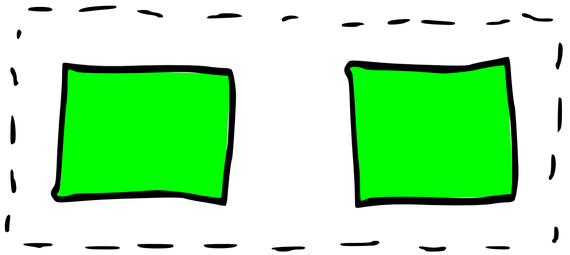



elasticsearch indexes

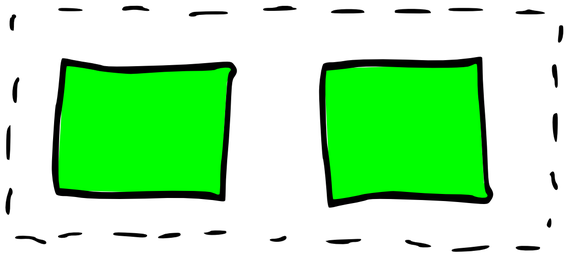
logs-2014-07-24

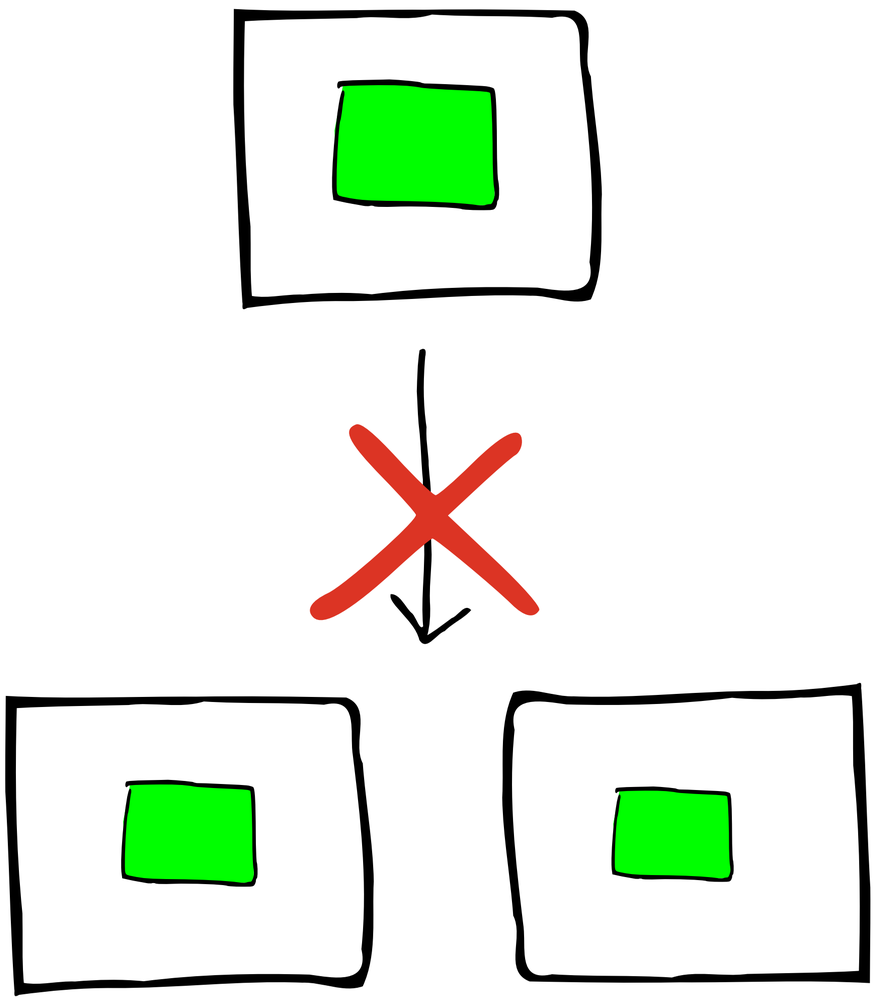


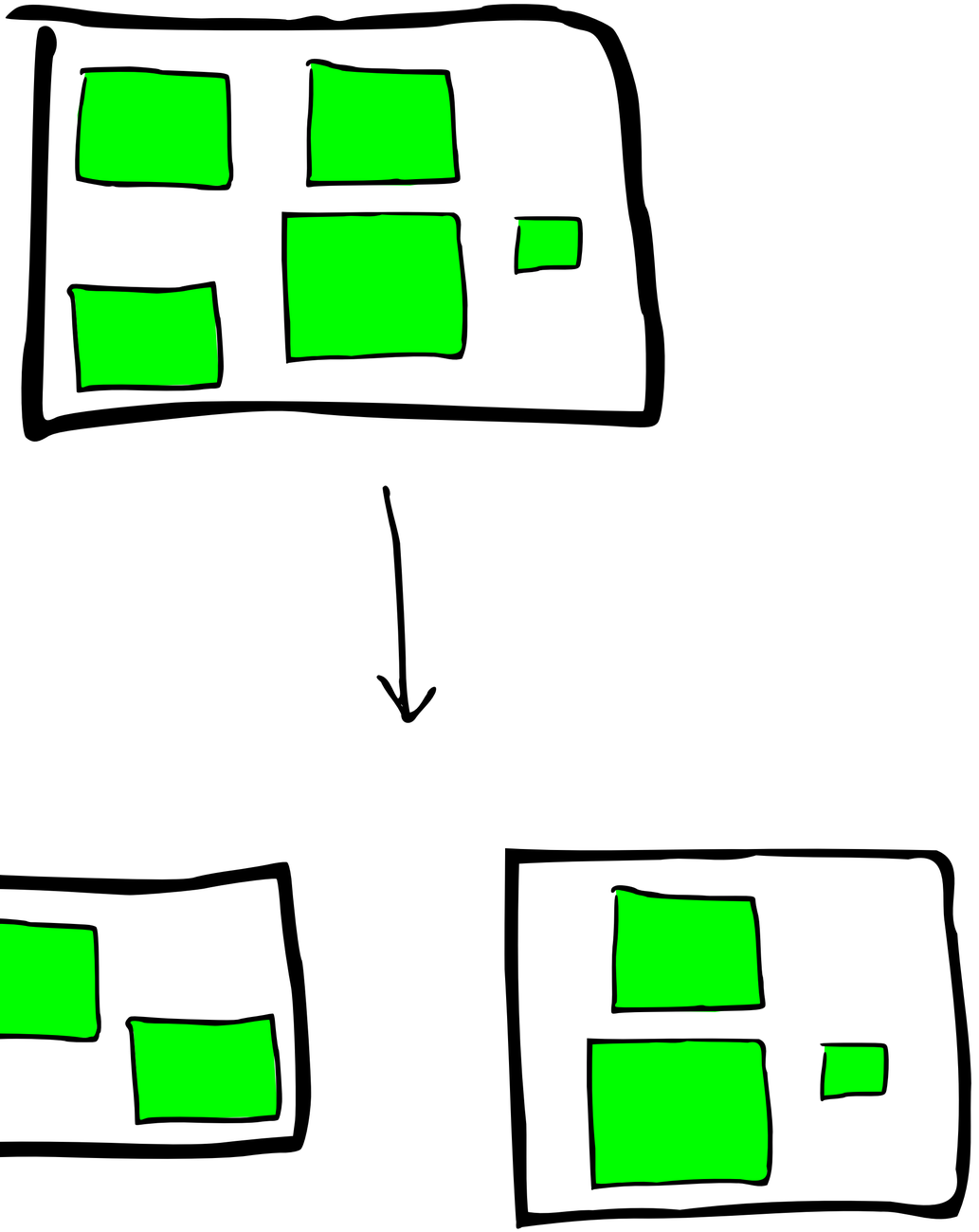
logs-2014-07-23

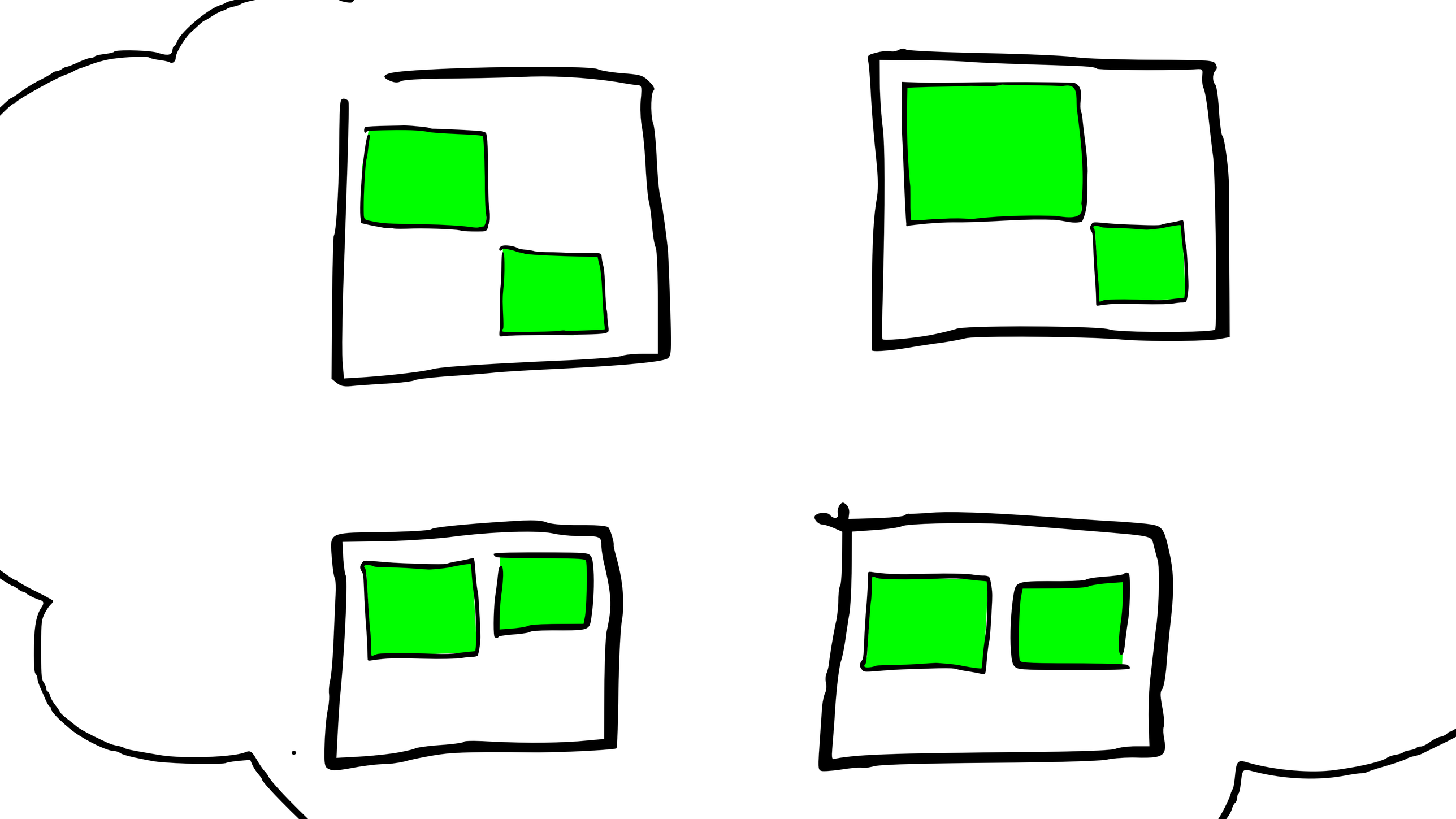


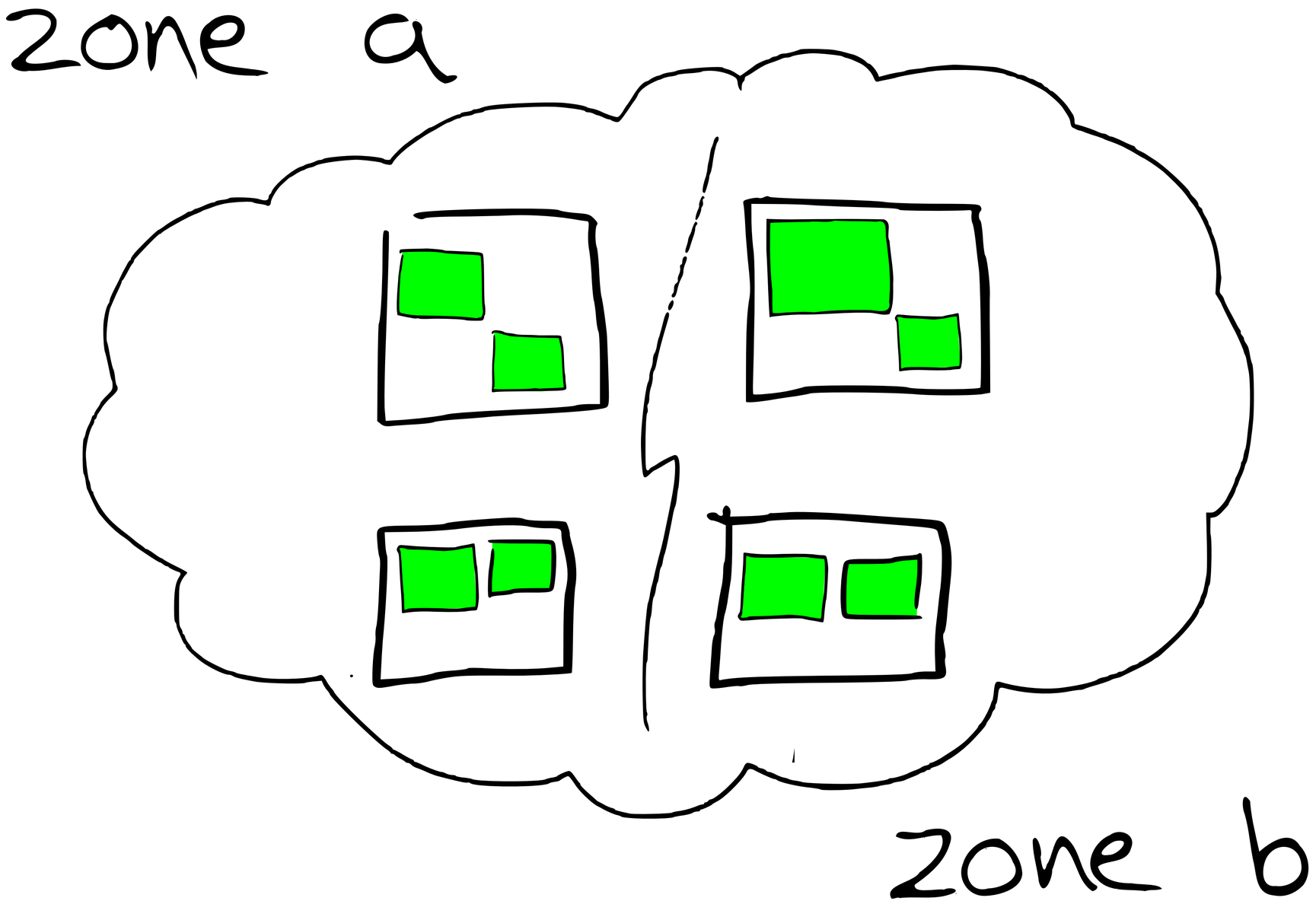
logs-2014-07-22

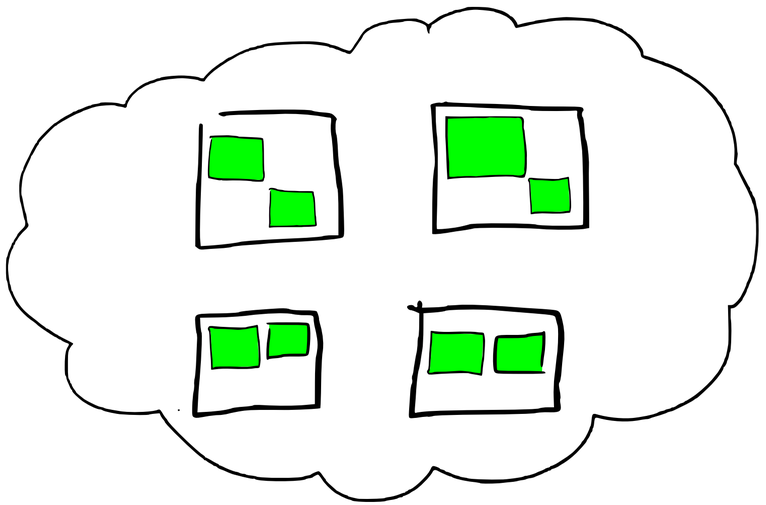









zone a
zone b

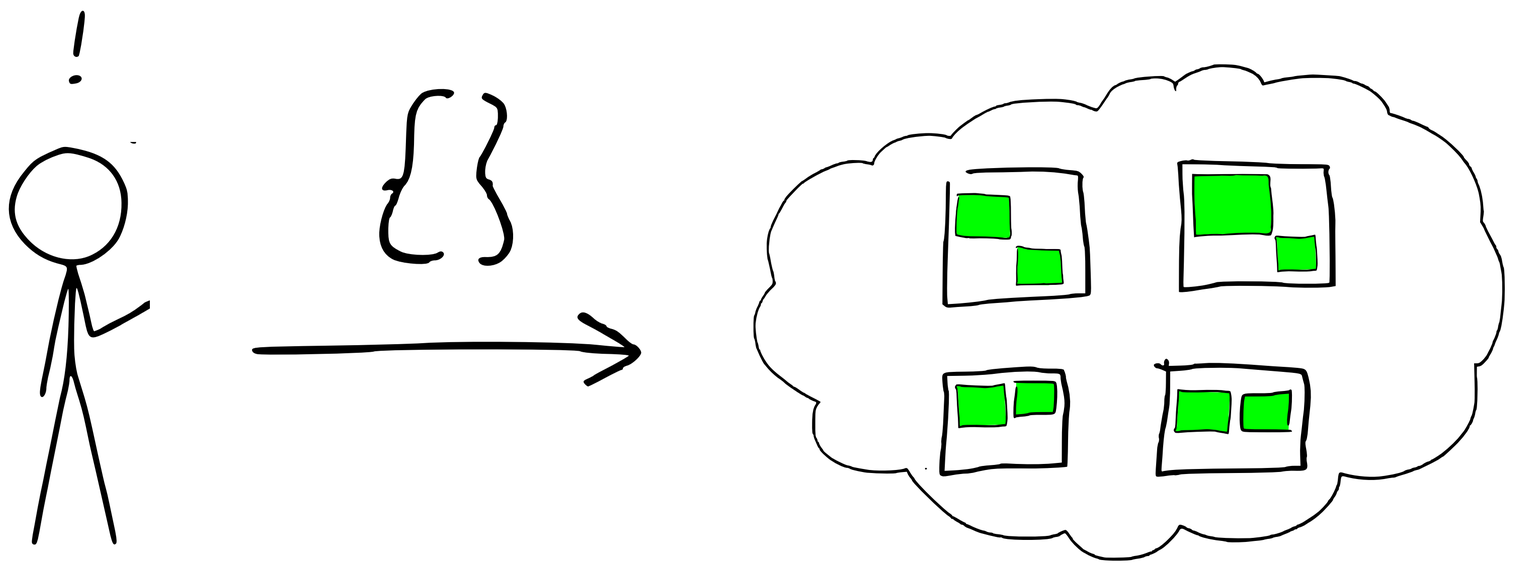


mappings

shard routing table

master node

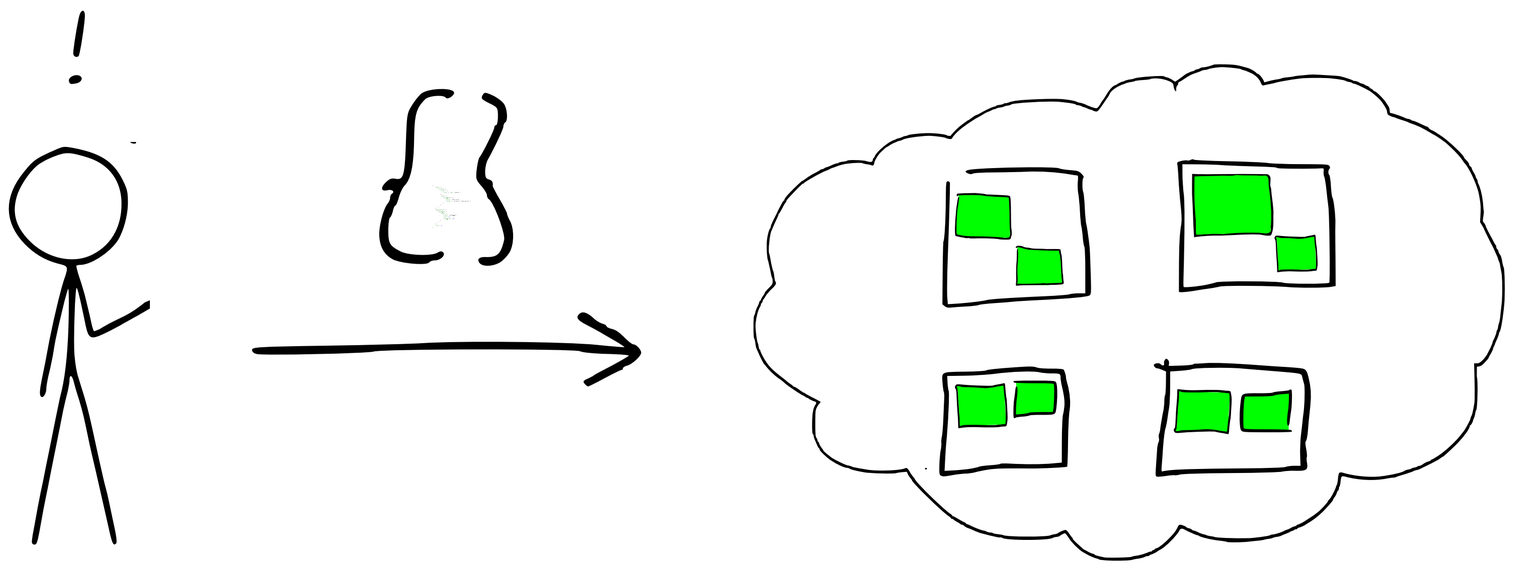
etc.

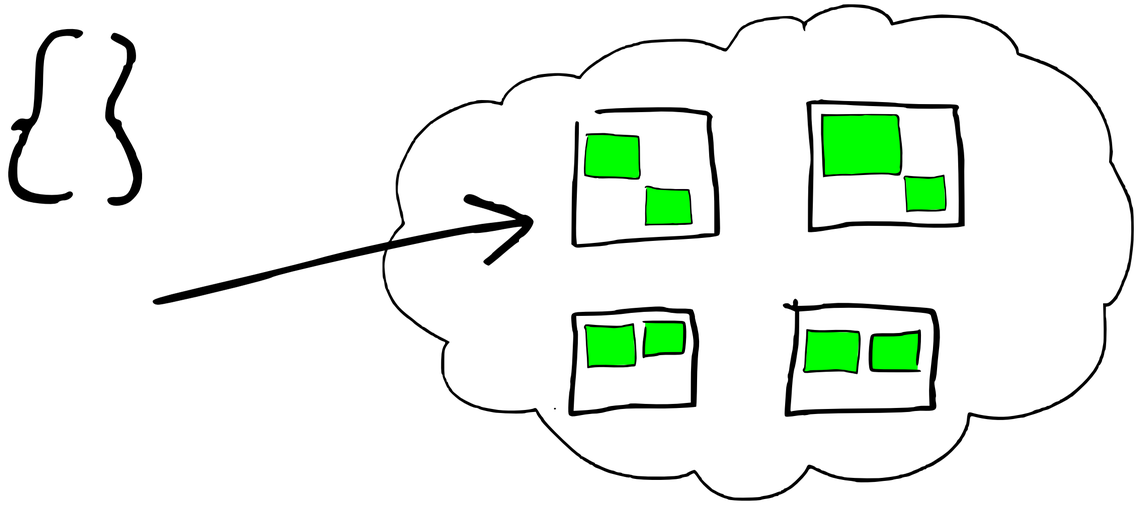


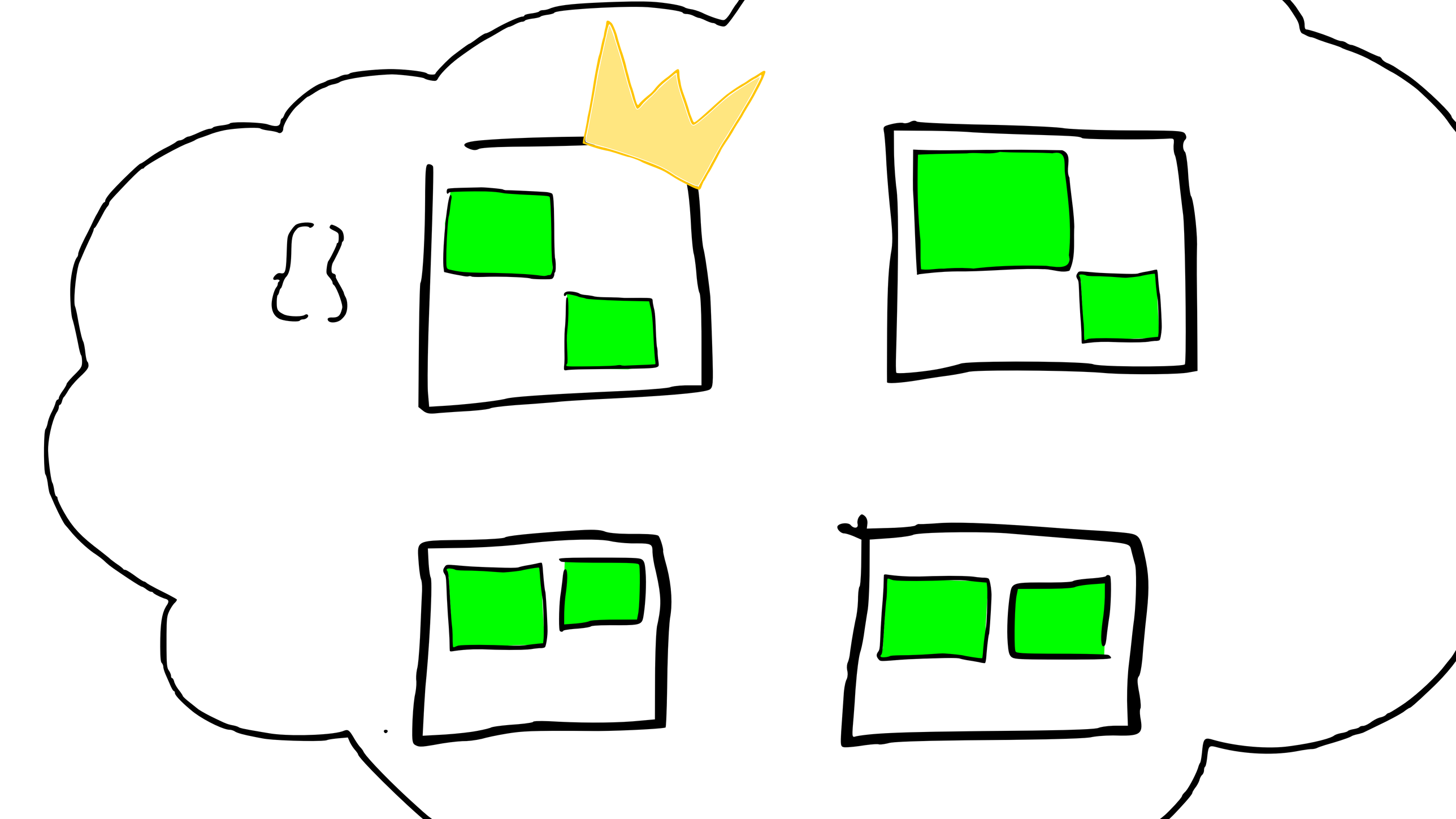
```
{
    "query": {
        "filtered": {
            "filter": {
                "term": { "tag": "python" }
            },
            "query": {
                "multi_match": {
                    "query": "Holy Grail",
                    "fields": [ "title^5", "description" ]
                }
            }
        }
    },
    "aggregations": {
        "author_id": {
            "terms": {
                "field": "author_id",
                "size": 10,
                "shard_size": 100
            }
        }
    },
    "size": 10
}
```

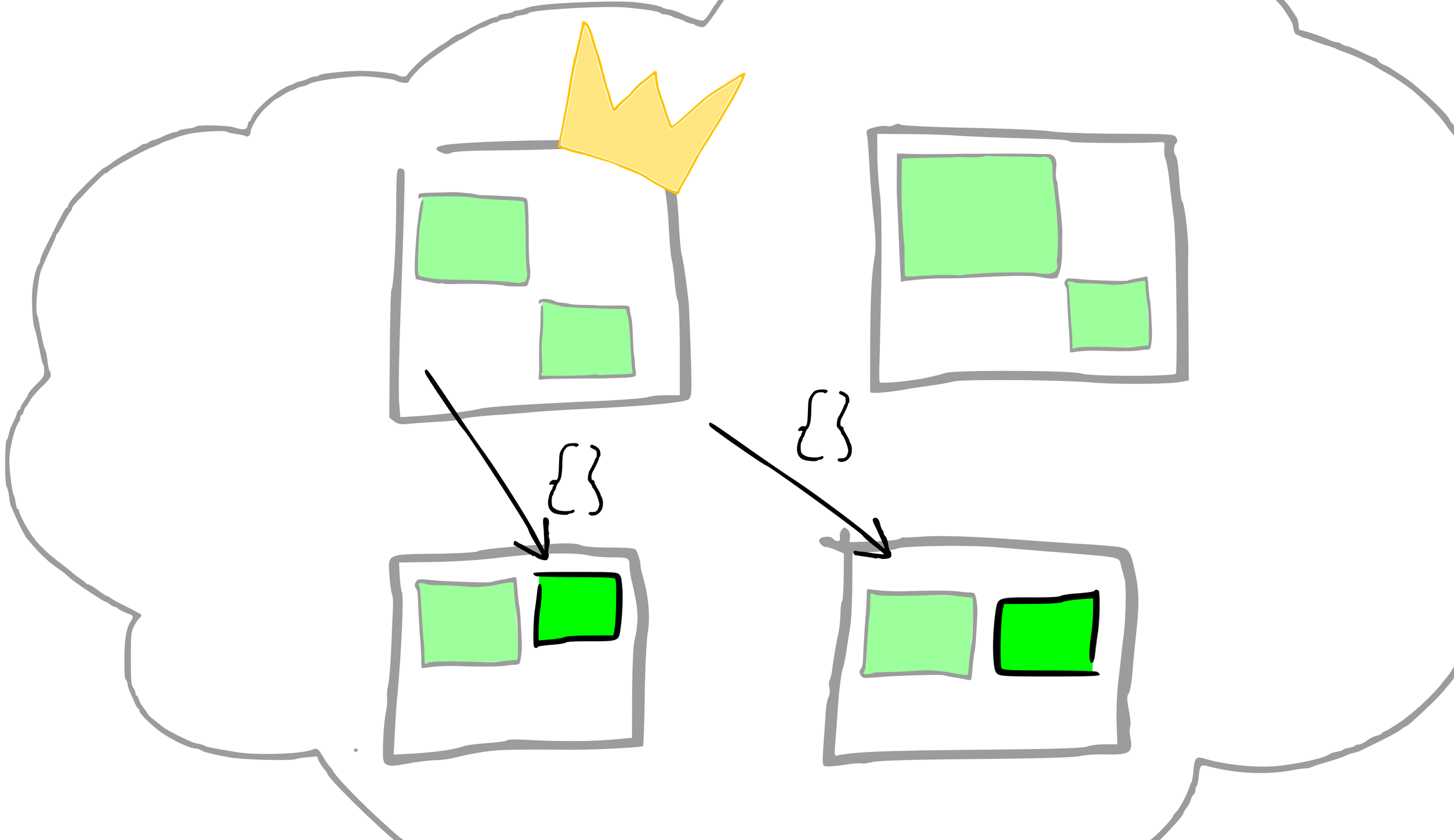
```
{
    "query": {
        "filtered": {
            "filter": {
                "term": { "tag": "python" }
            },
            "query": {
                "multi_match": {
                    "query": "Holy Grail",
                    "fields": [ "title^5", "description" ]
                }
            }
        }
    },
```

```
    "aggregations": {
        "author_id": {
            "terms": {
                "field": "author_id",
                "size": 10,
                "shard_size": 100
            }
        }
    },
    "size": 10
}
```

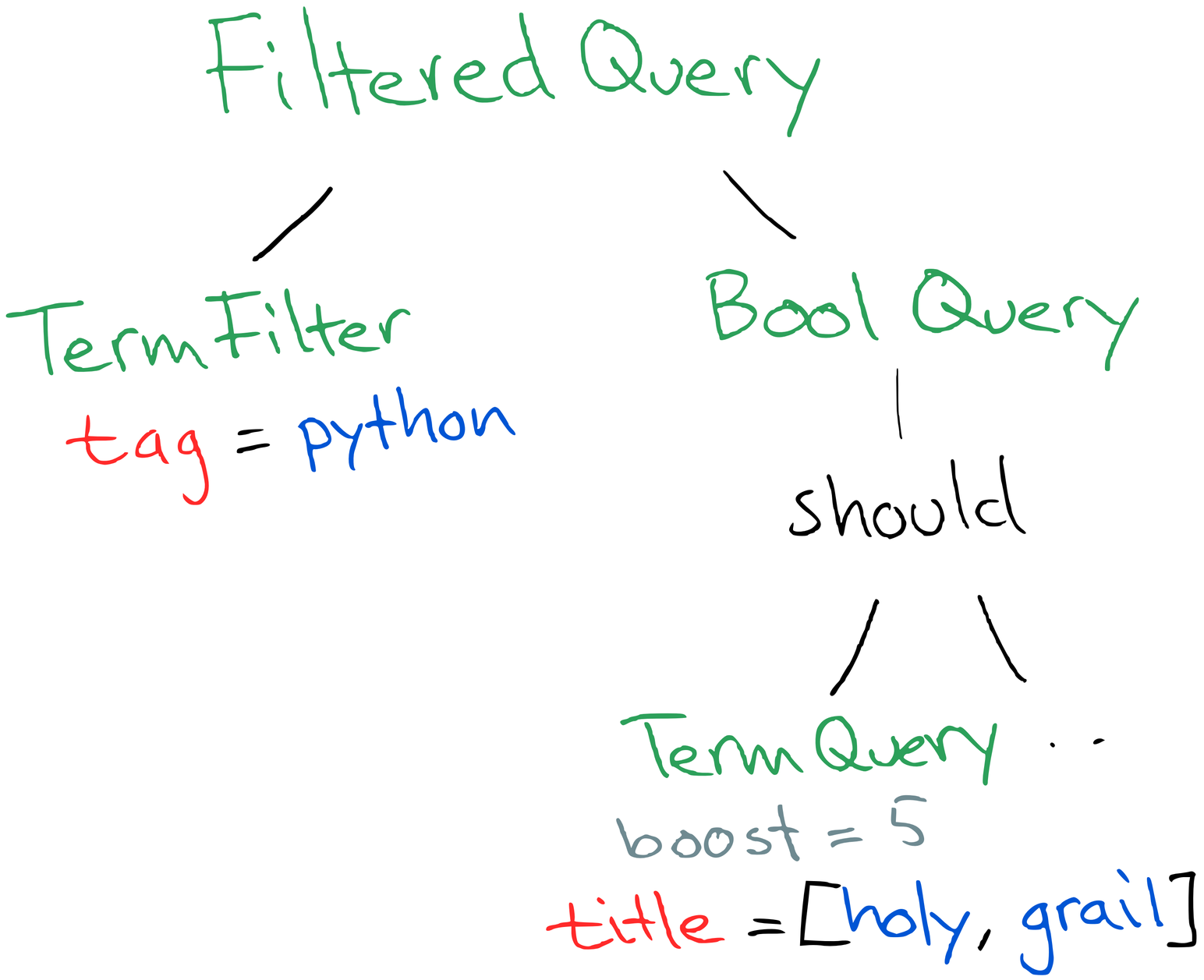


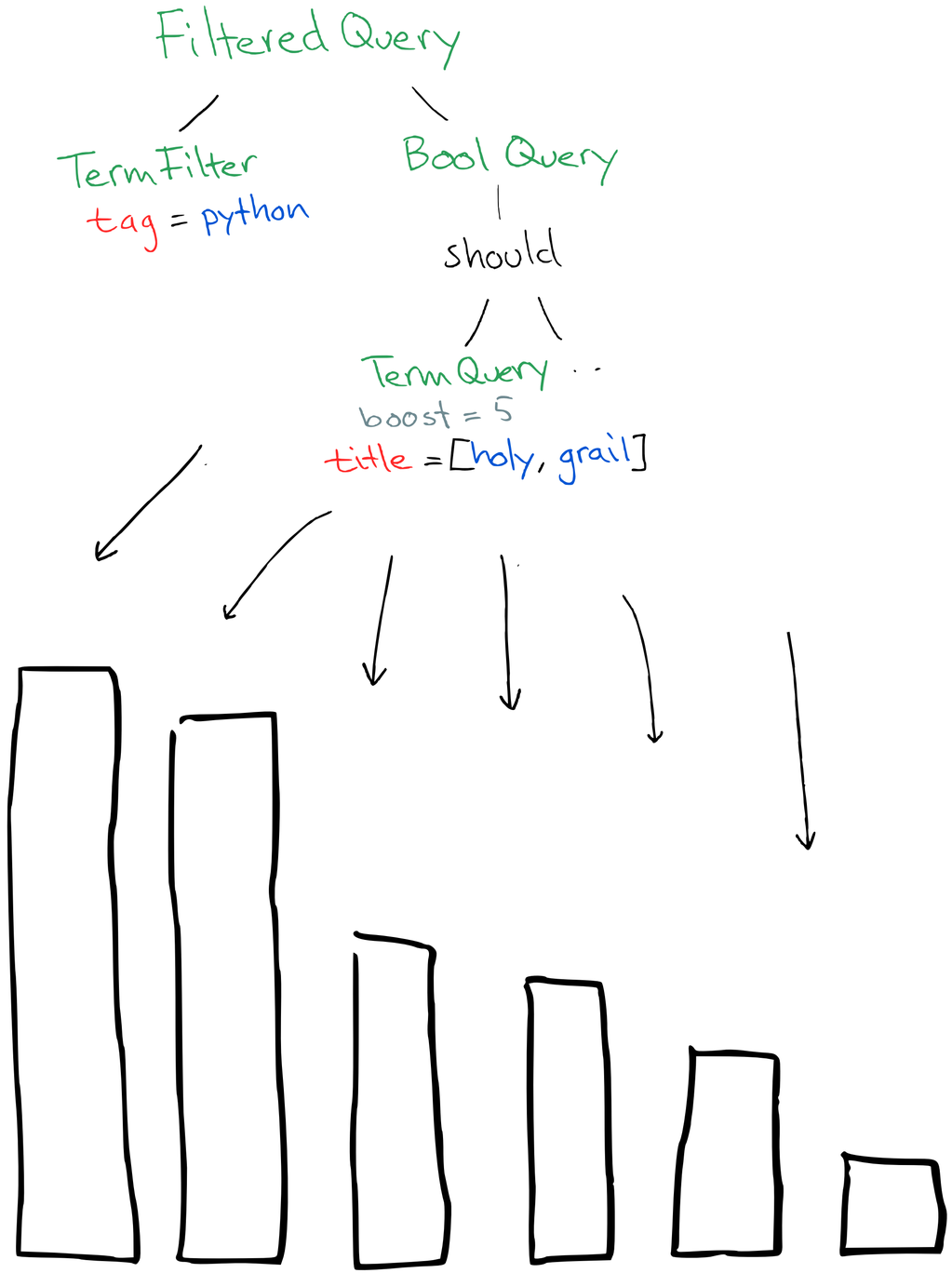


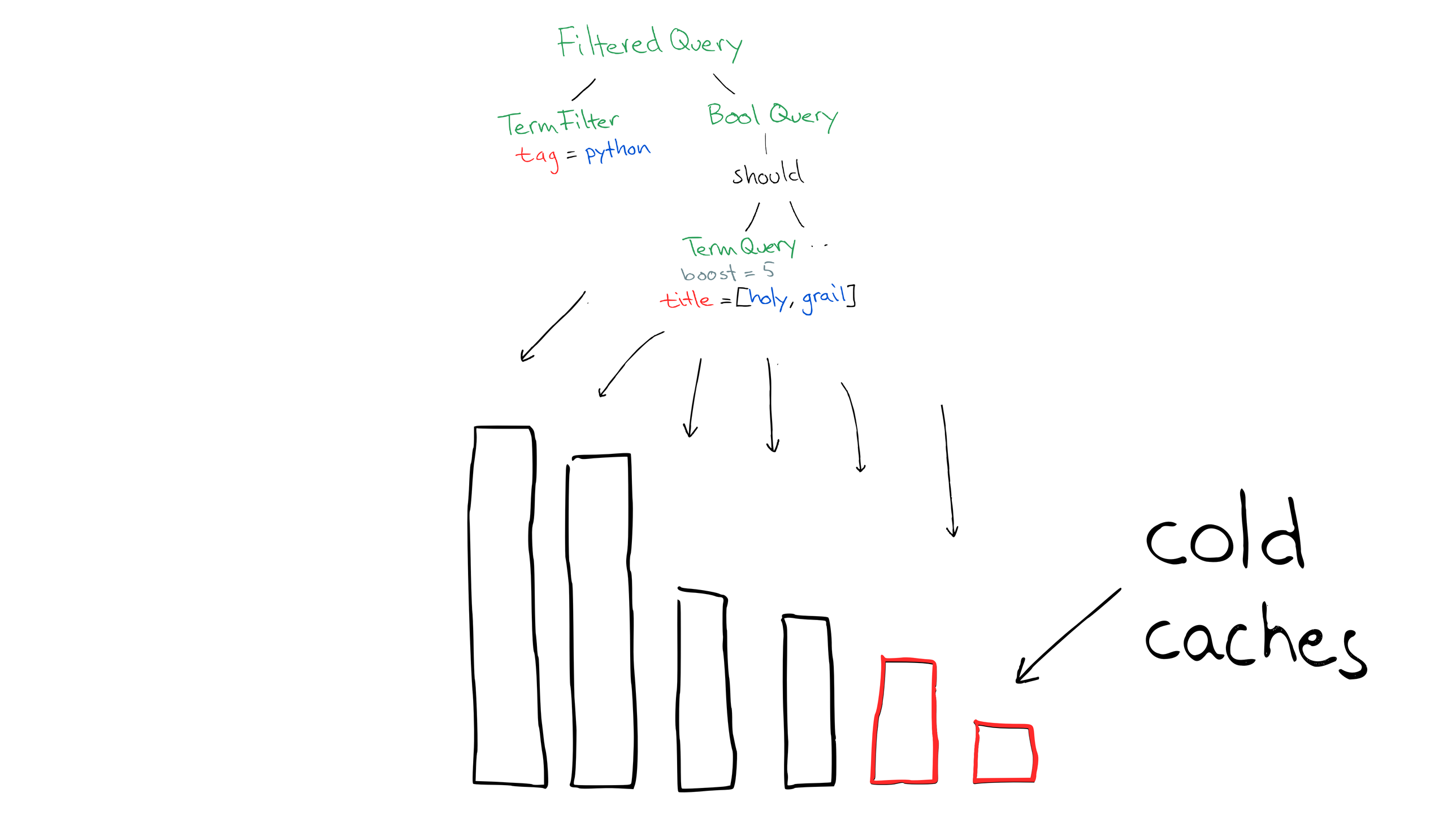


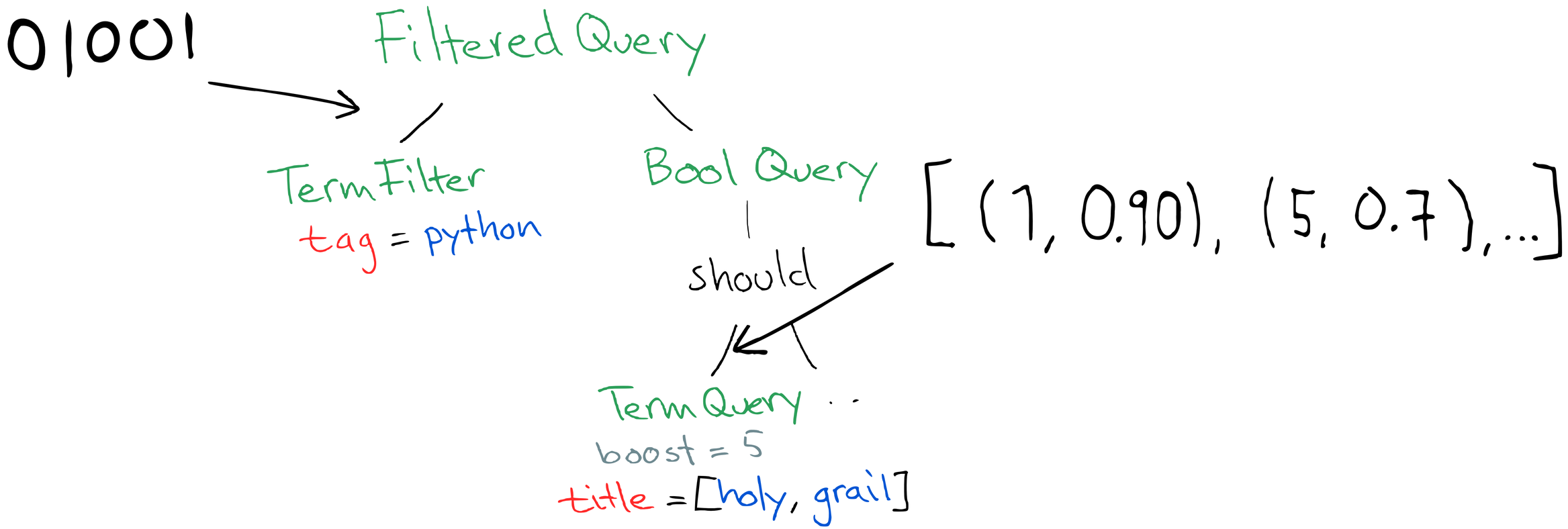


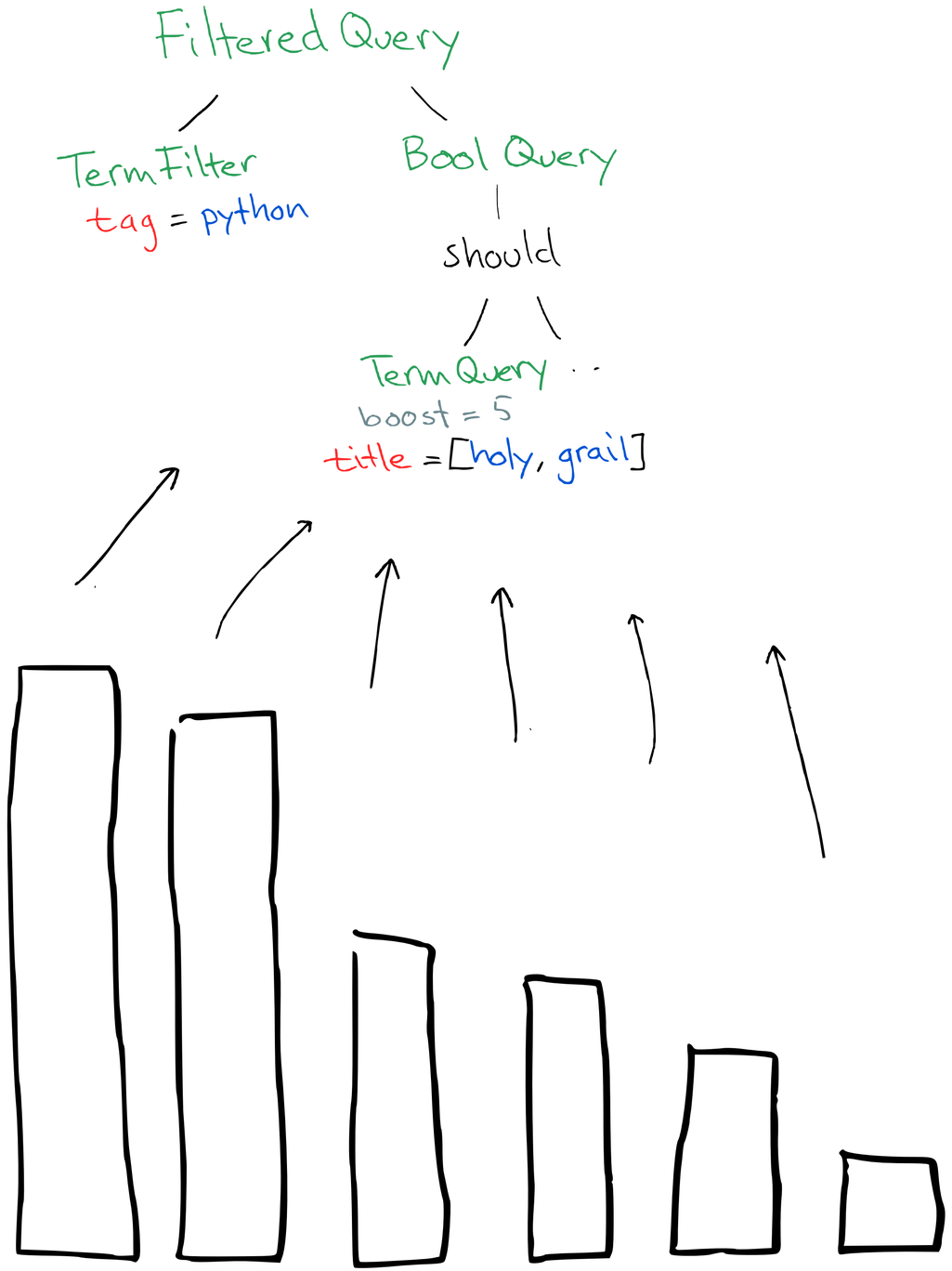
```
"query": {
    "filtered": {
        "filter": {
            "term": { "tag": "python" }
        },
        "query": {
            "multi_match": {
                "query": "Holy Grail",
                "fields": [ "title^5", "description" ]
            }
        }
    }
},
```

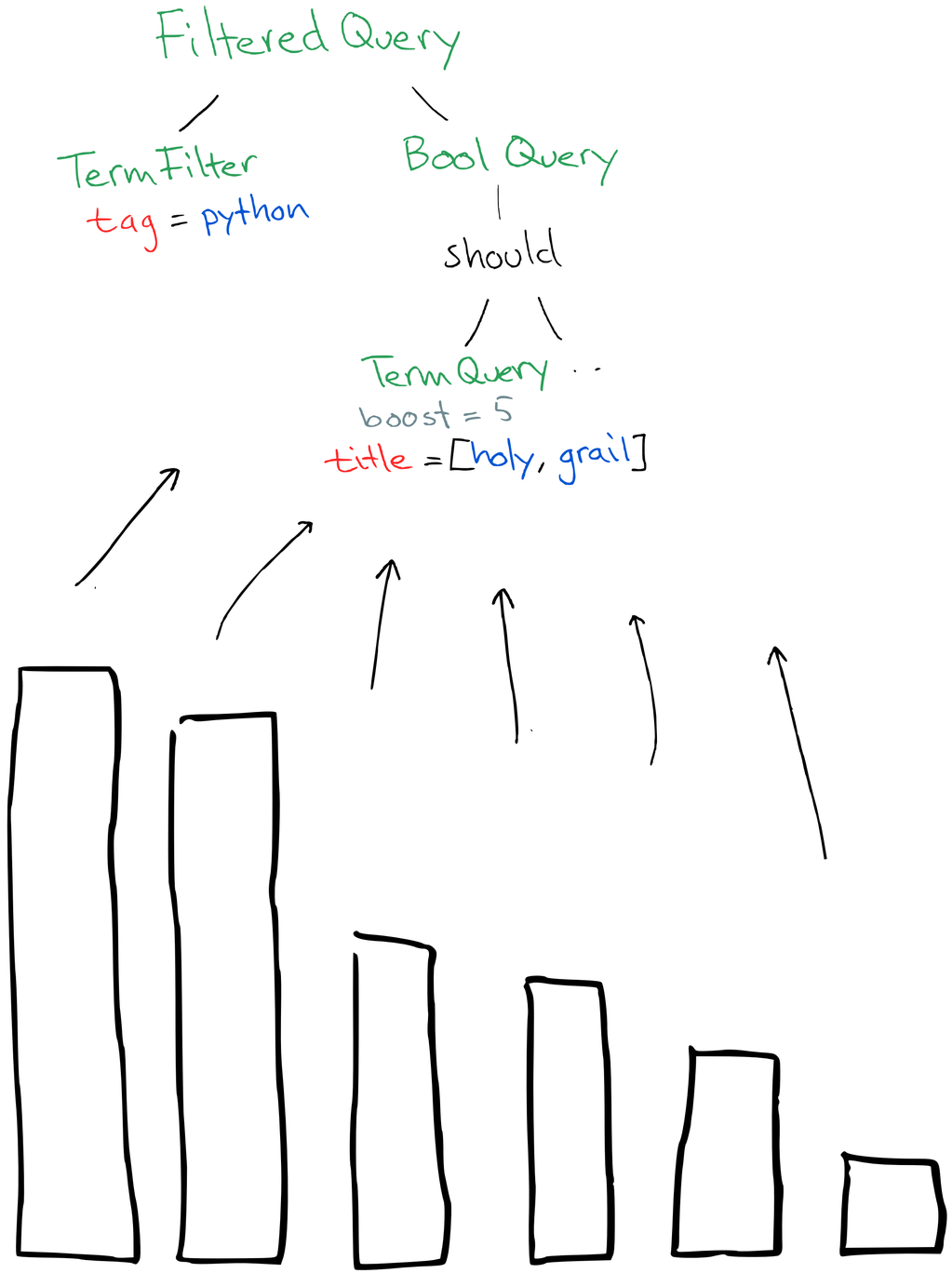

Filtered Query
TermFilter
tag = python
Bool Query
should
TermQuery
boost = 5
title = [holy, grail]

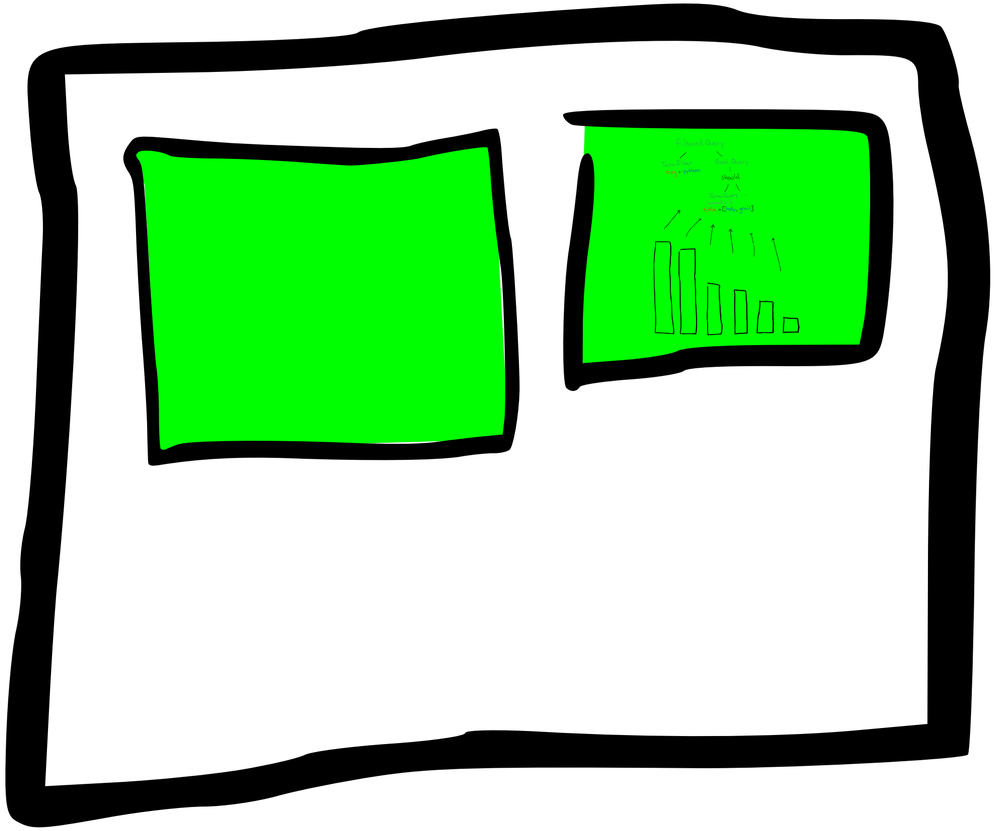

Filtered Query
TermFilter
tag = python
Bool Query
should
TermQuery
boost = 5
title = [holy, grail]

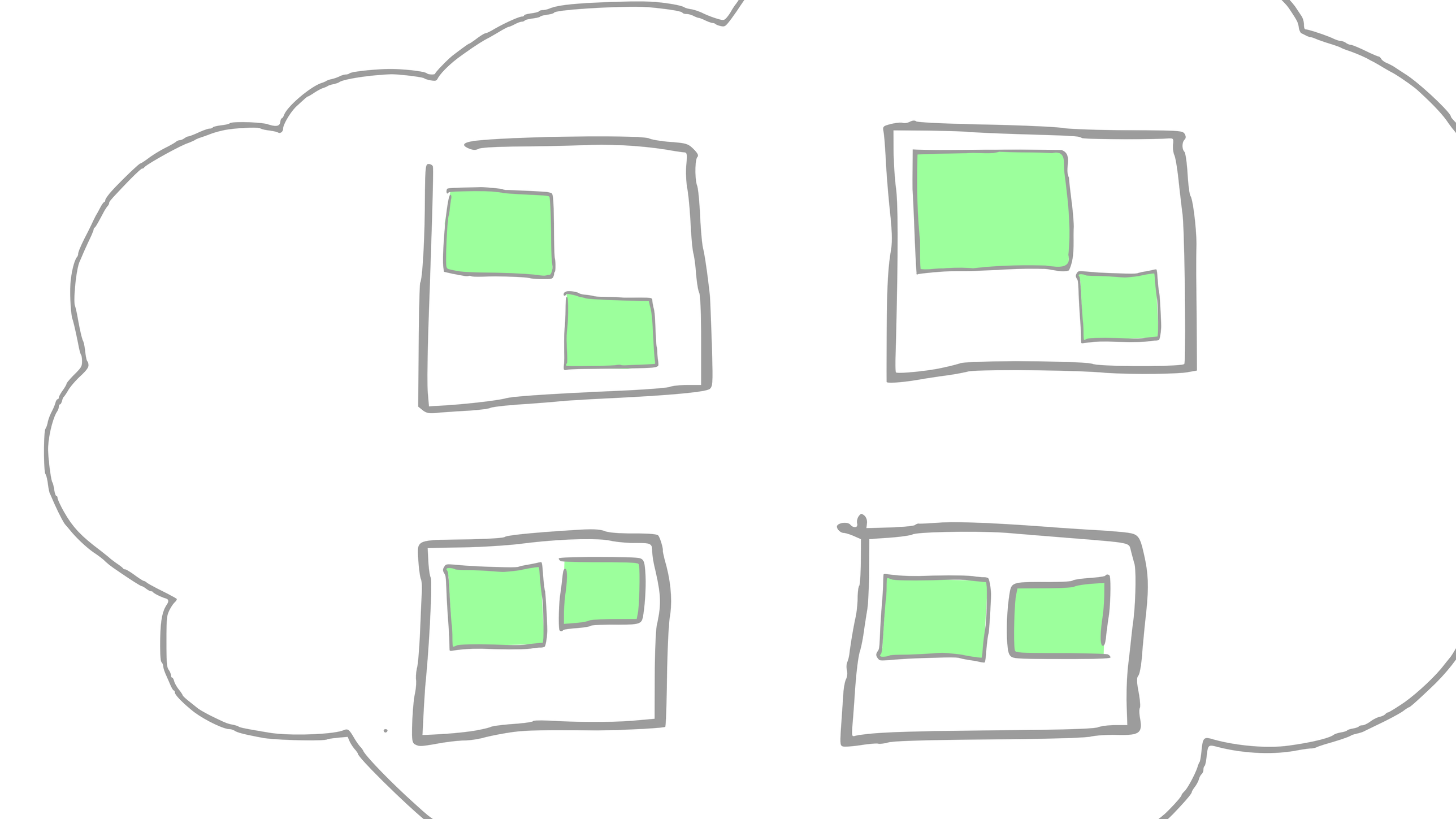

Filtered Query
TermFilter
tag = python
Bool Query
should
TermQuery
boost = 5
title = [holy, grail]
cold
caches

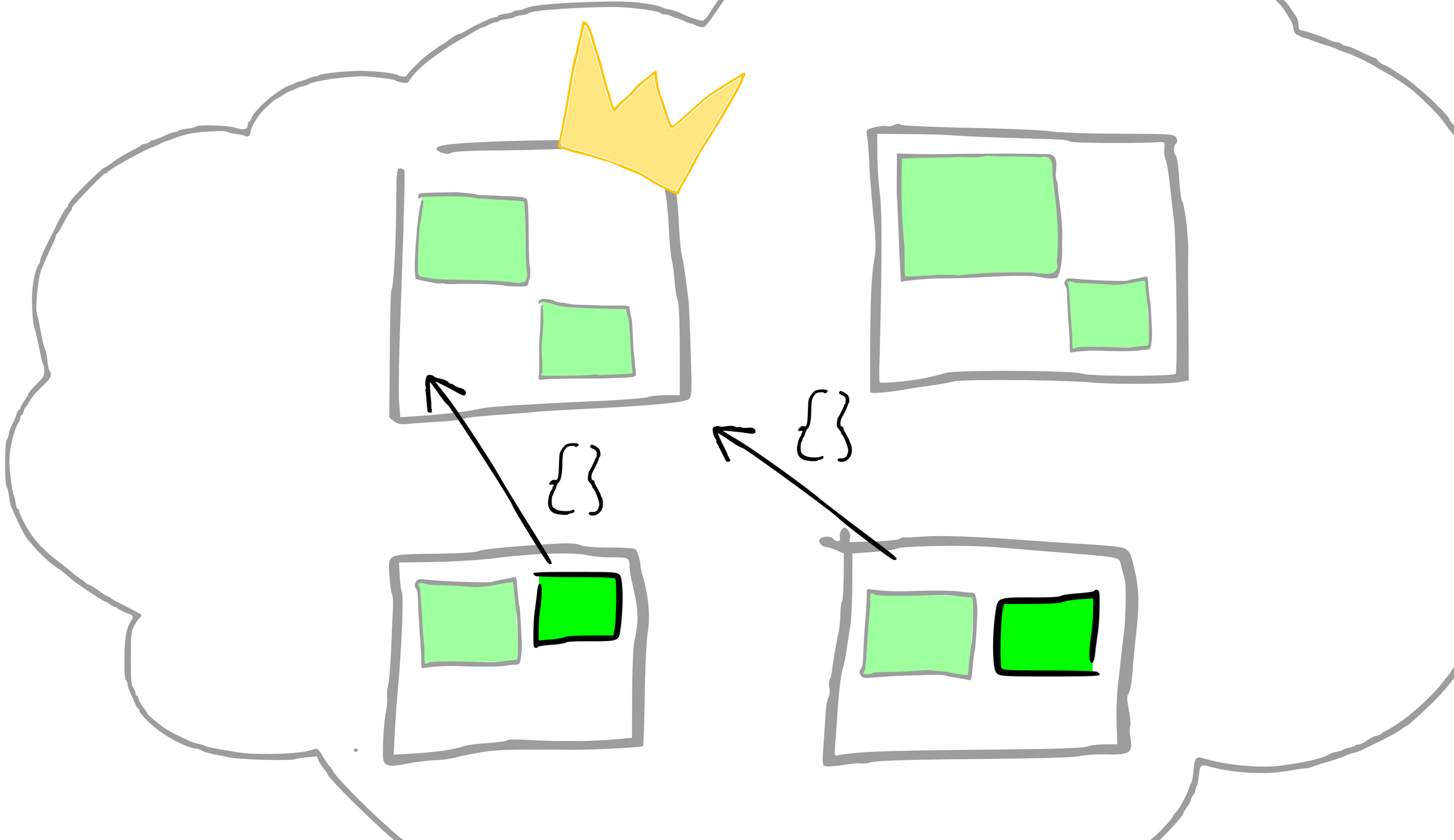

01001
Filtered Query
TermFilter
tag = python
Bool Query
should
TermQuery
boost = 5
title = [holy, grail]
[(1, 0.90), (5, 0.7), ...]

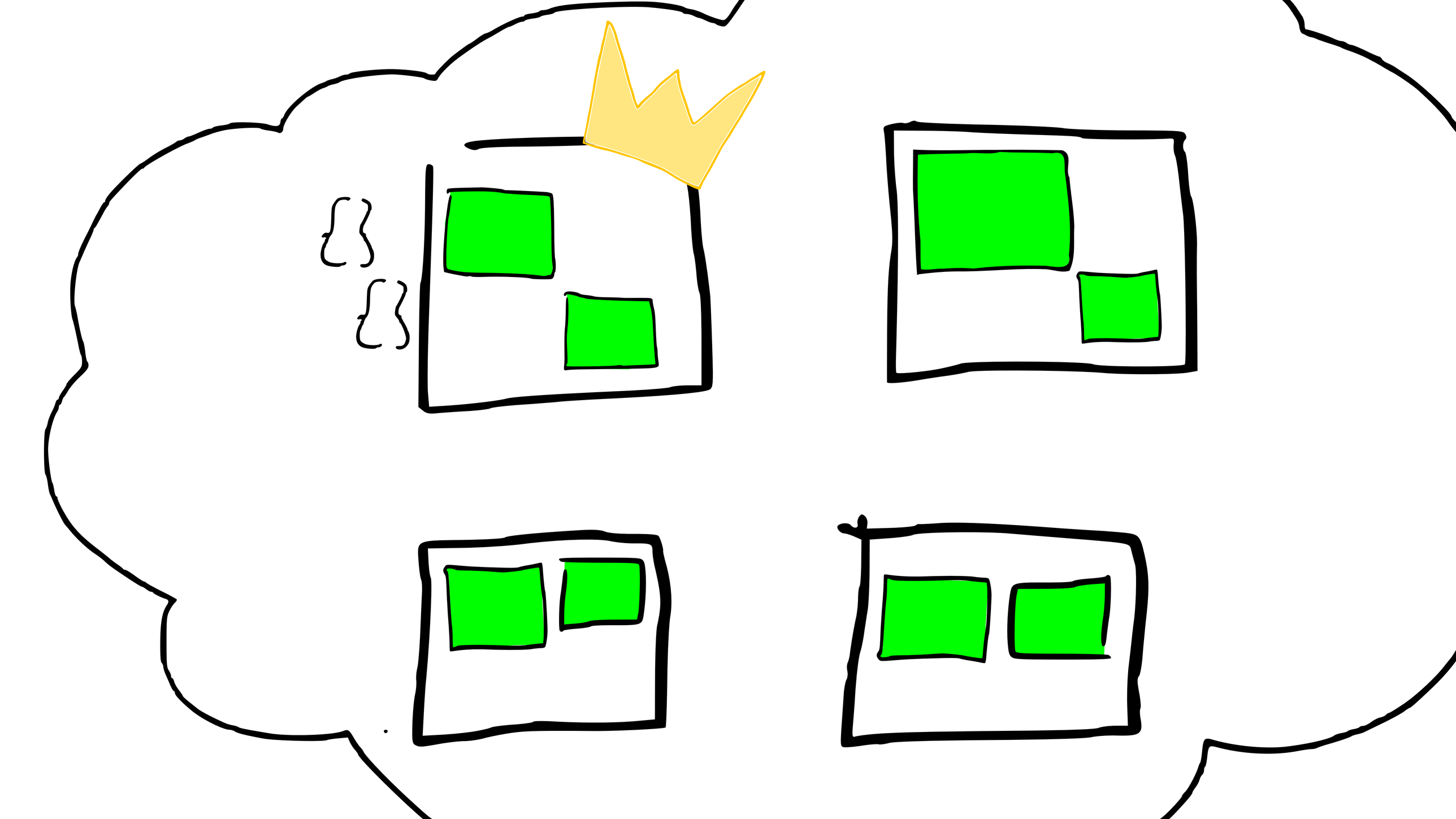

Filtered Query
TermFilter
tag = python
Bool Query
should
TermQuery
boost = 5
title = [holy, grail]

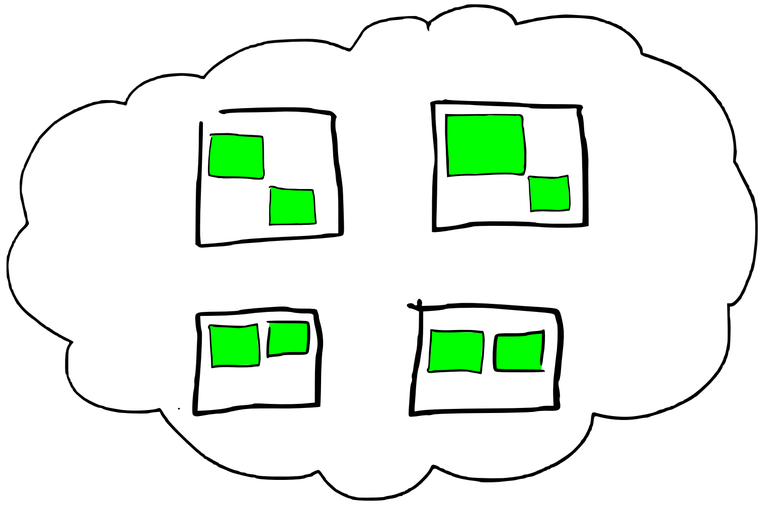

Filtered Query
TermFilter
tag = python
Bool Query
should
TermQuery
boost = 5
title = [holy, grail]

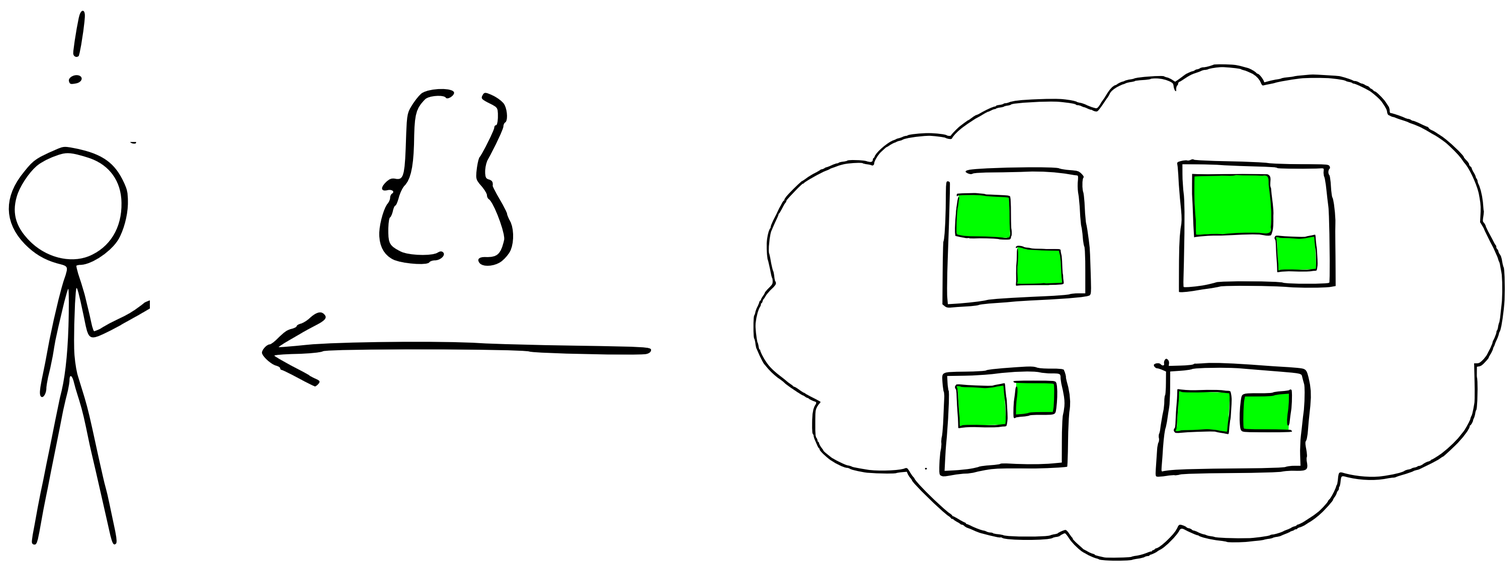












Search by index terms

Text analysis gives us terms

Search by segment

Uses several data structures

# Immutable segments

# Shard == Lucene Index

# Elasticsearch Index abstracts Lucene Indexes

# … across nodes in a cluster

?!?

# Learn More!

found.no/foundation

Follow

@alexbrasetvik @foundsays

FOUND