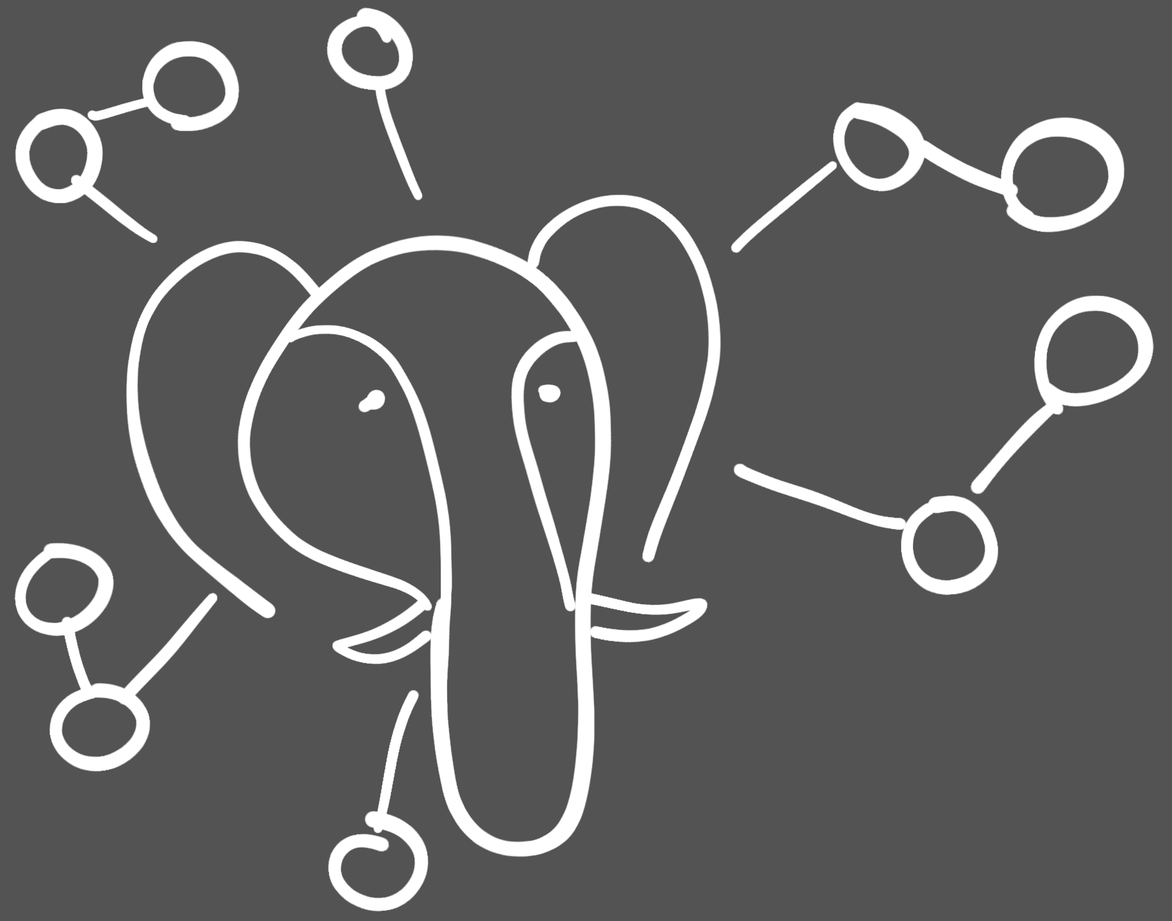
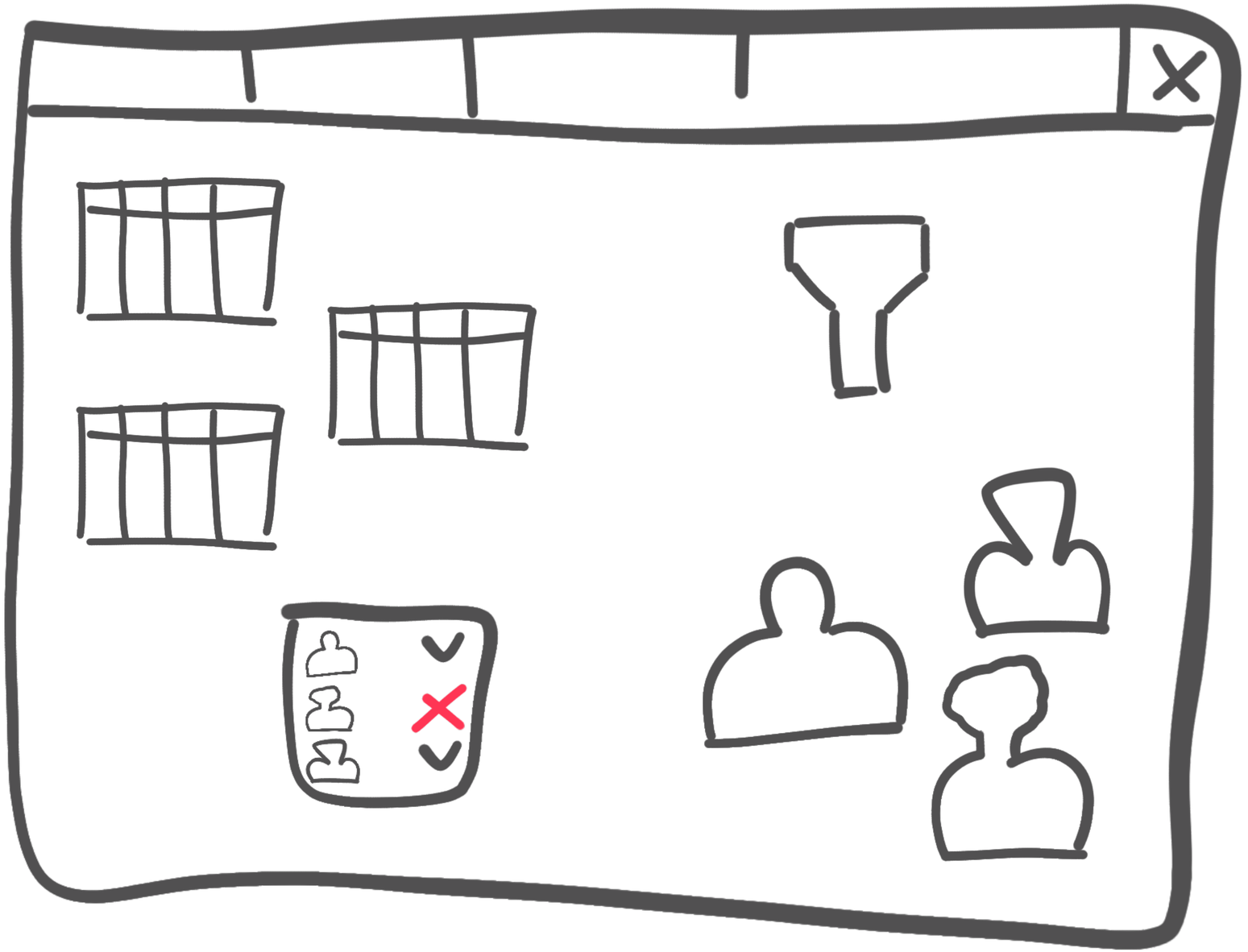
aiven

Put PostgreSQL at the center of your (meta)data universe!



Francesco Tisiot - Senior Developer Advocate

@ftisiot



https://www.postgresql.org/docs/current/catalogs.html

pg_fdw

pg_indexes

pg_users

pg_tables

pg_triggers

pg_rules

pg_extension

pg_views

pg_user_mappings

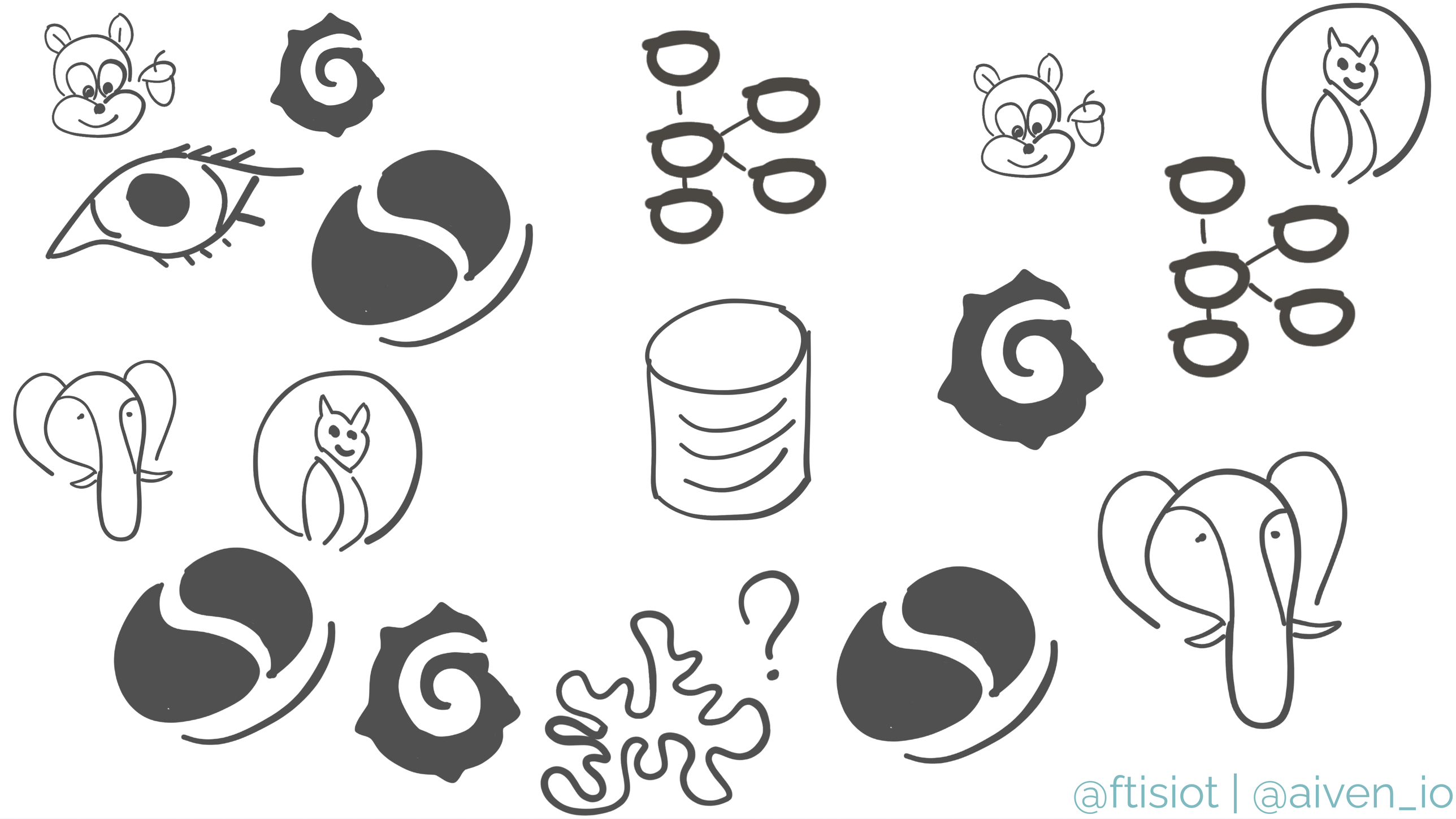
pg_replication_slots

pg_available_indexes

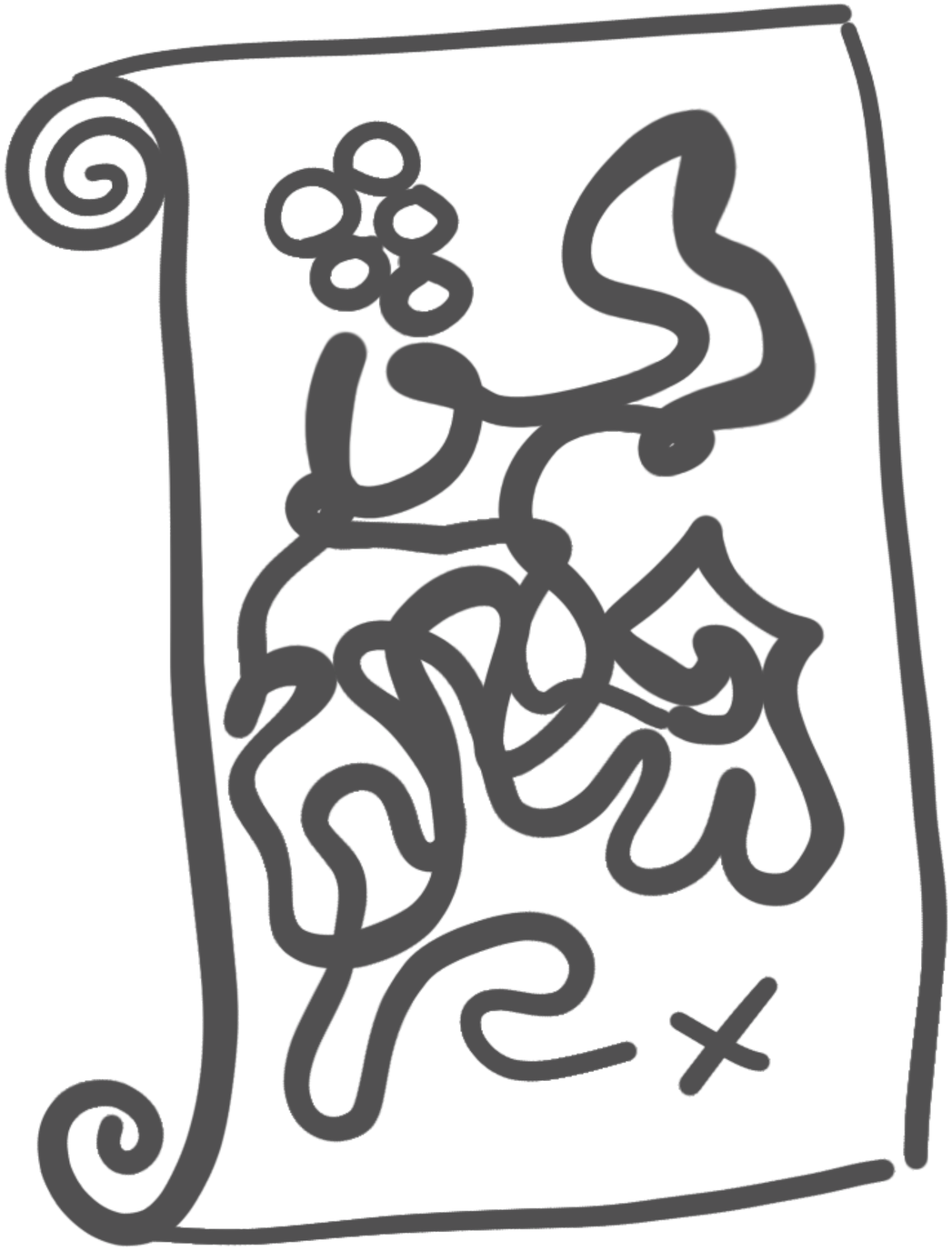
pg_roles

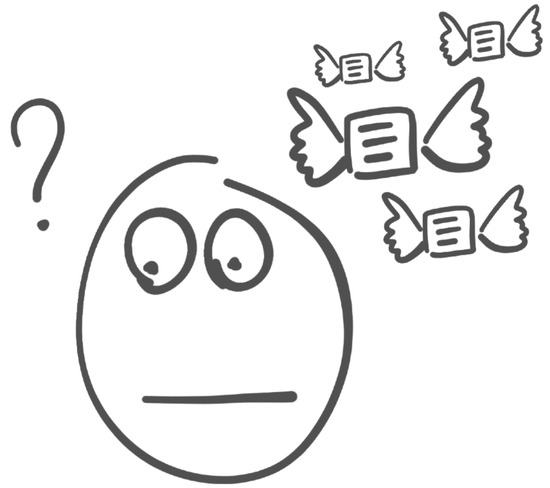
96

tables in PG 14

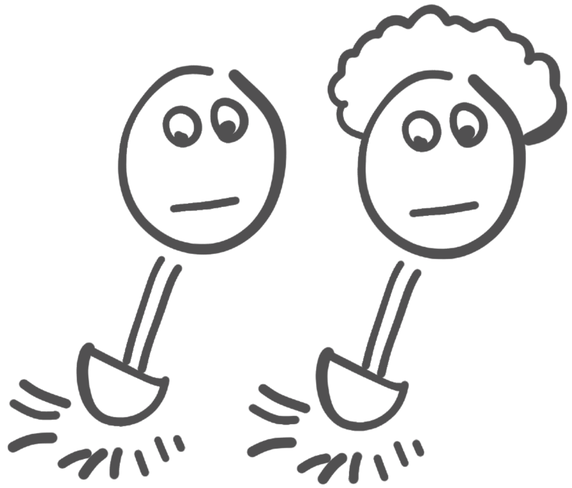


@ftisiot | @aiven_io

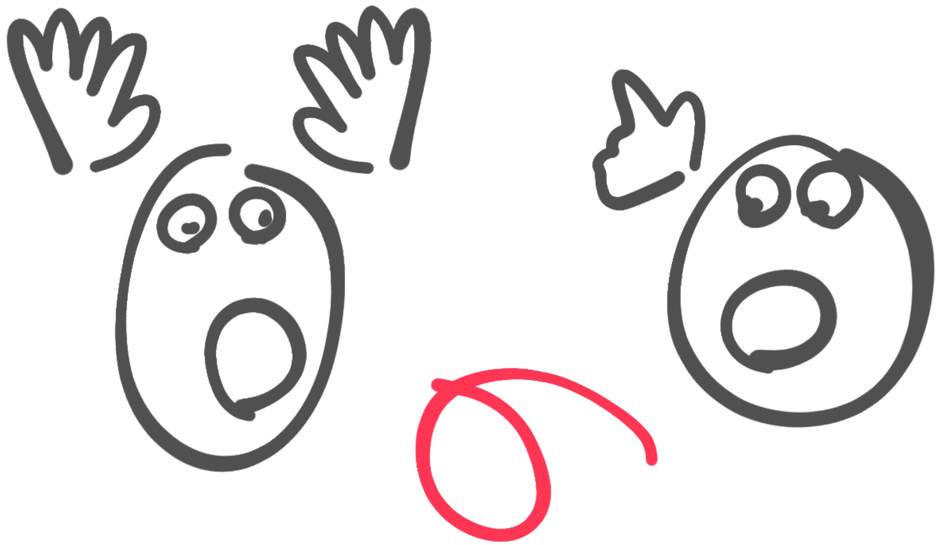




Lost Information



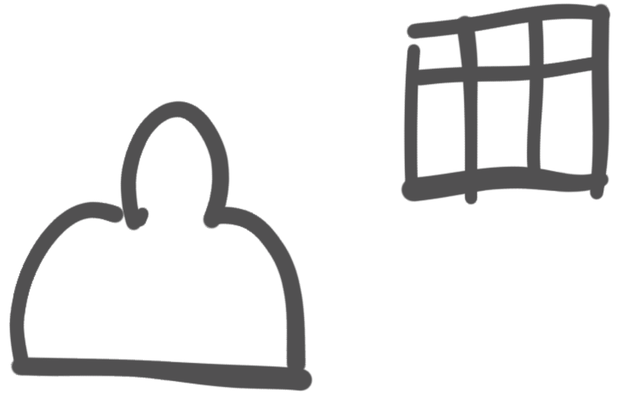
Duplicated Efforts



Conflicting Data

Data Loss

Data Lineage

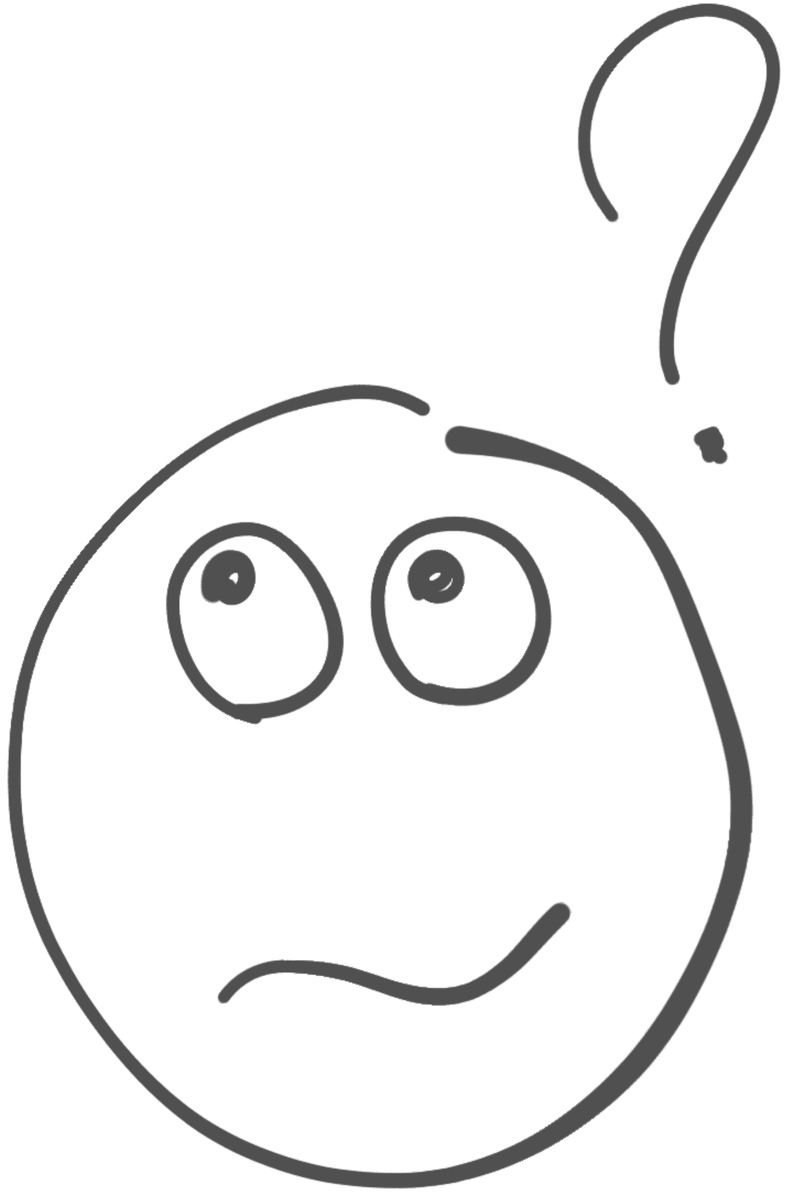


Security Assessment

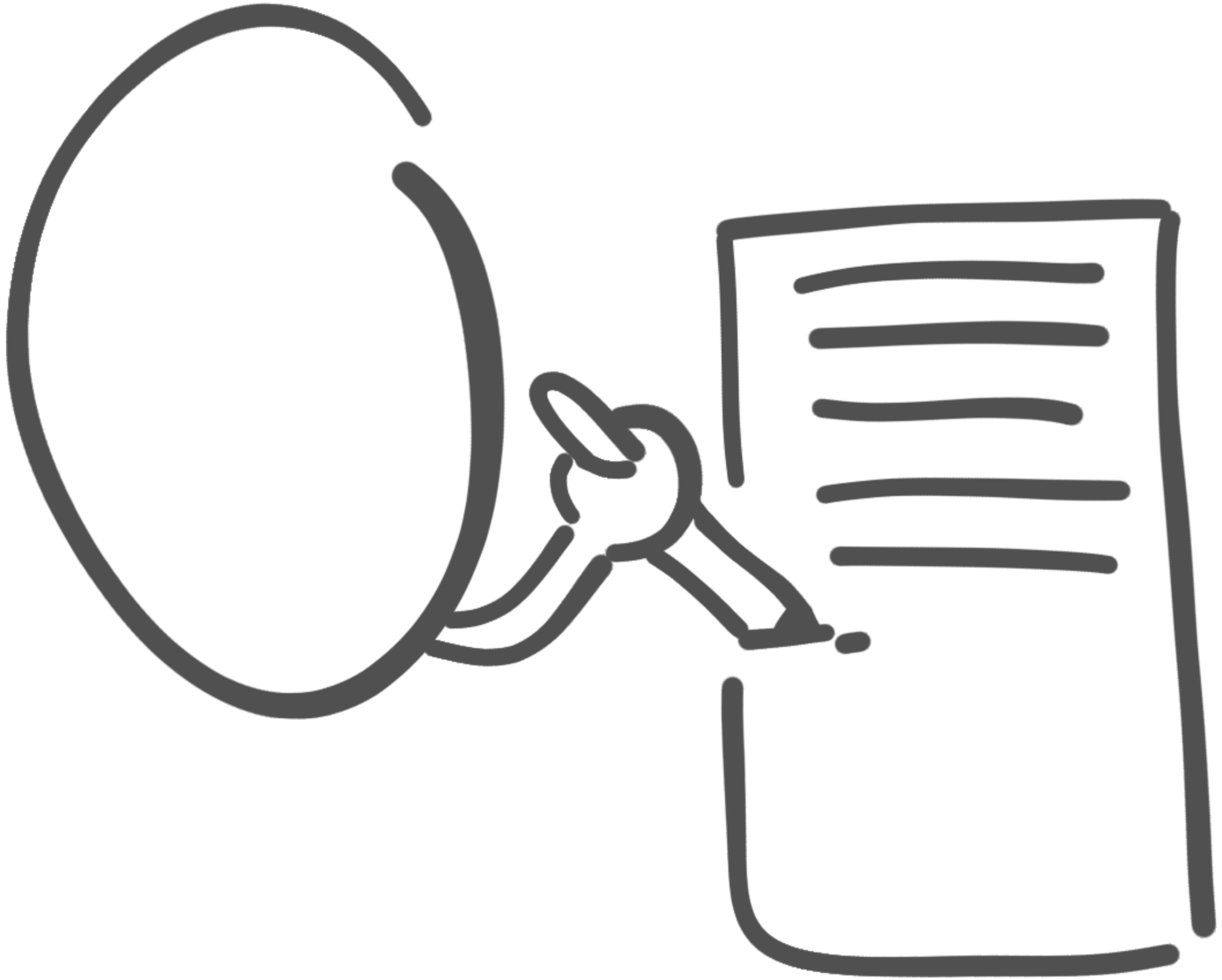
Impact Assessments

GDPR Assessments

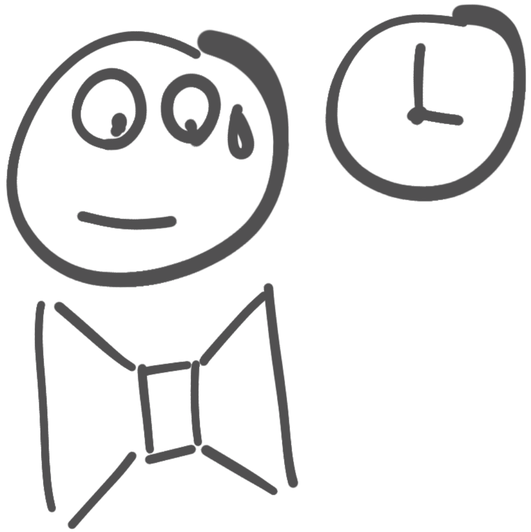
How to fix?



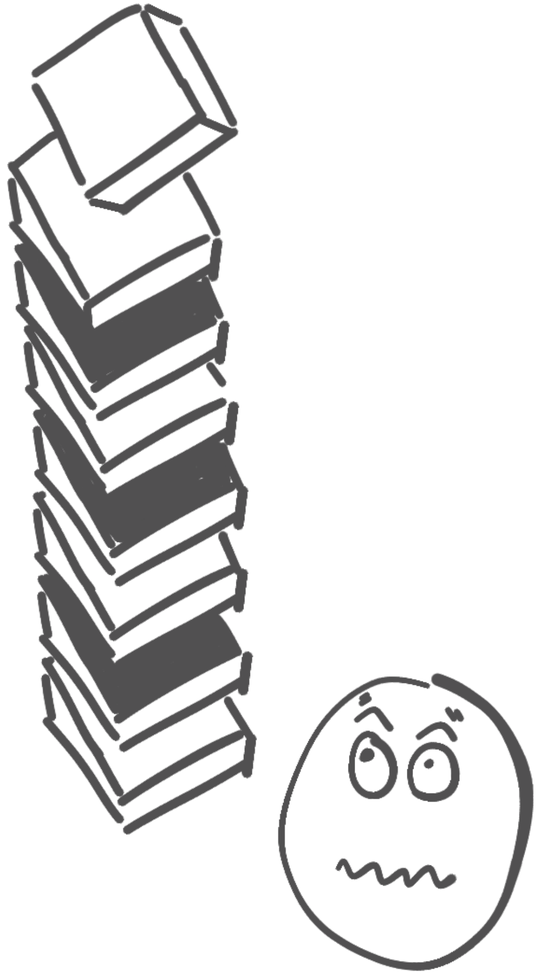
Docs FTW!



Complex



Time consuming



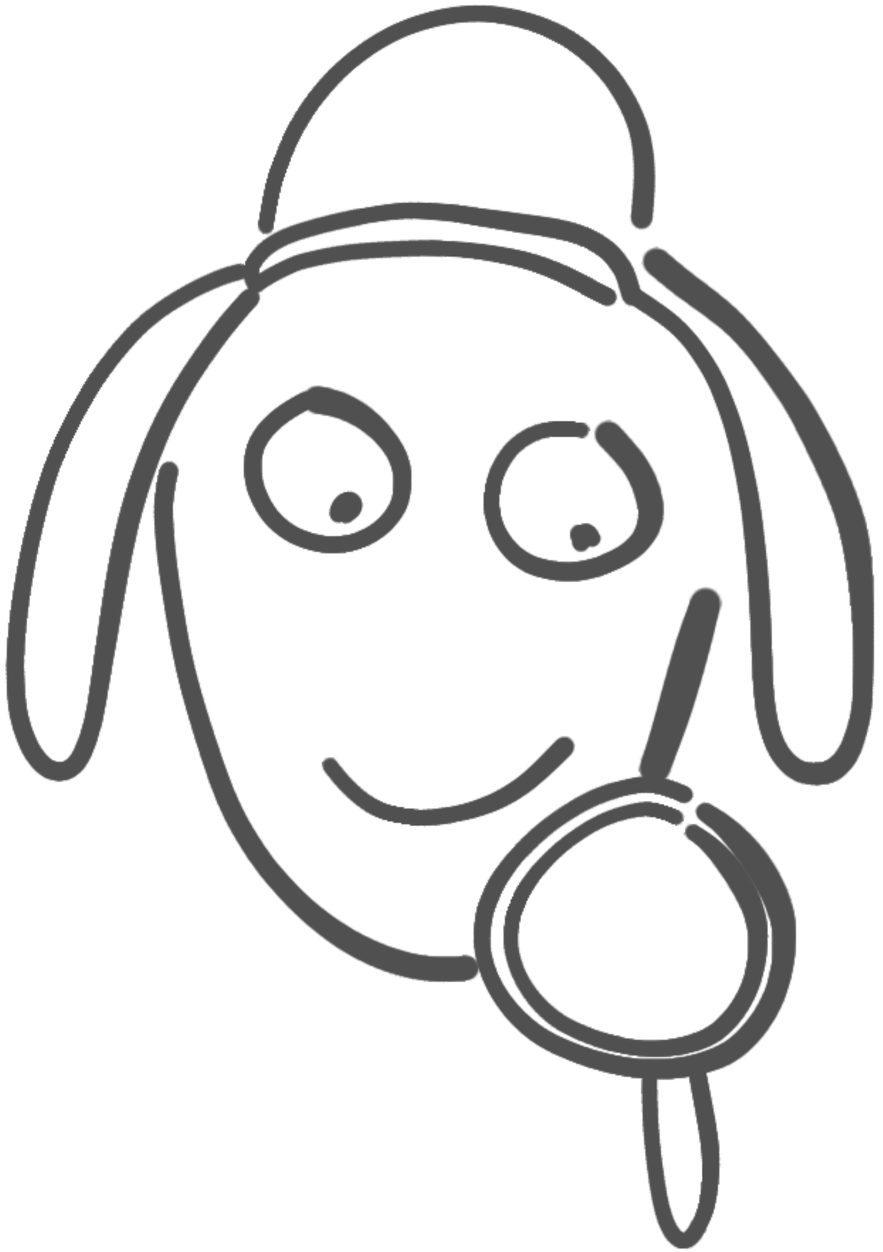
Piling Up

Absent

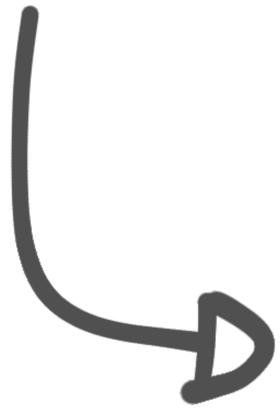
Leaving

How to fix, please?

Look around!



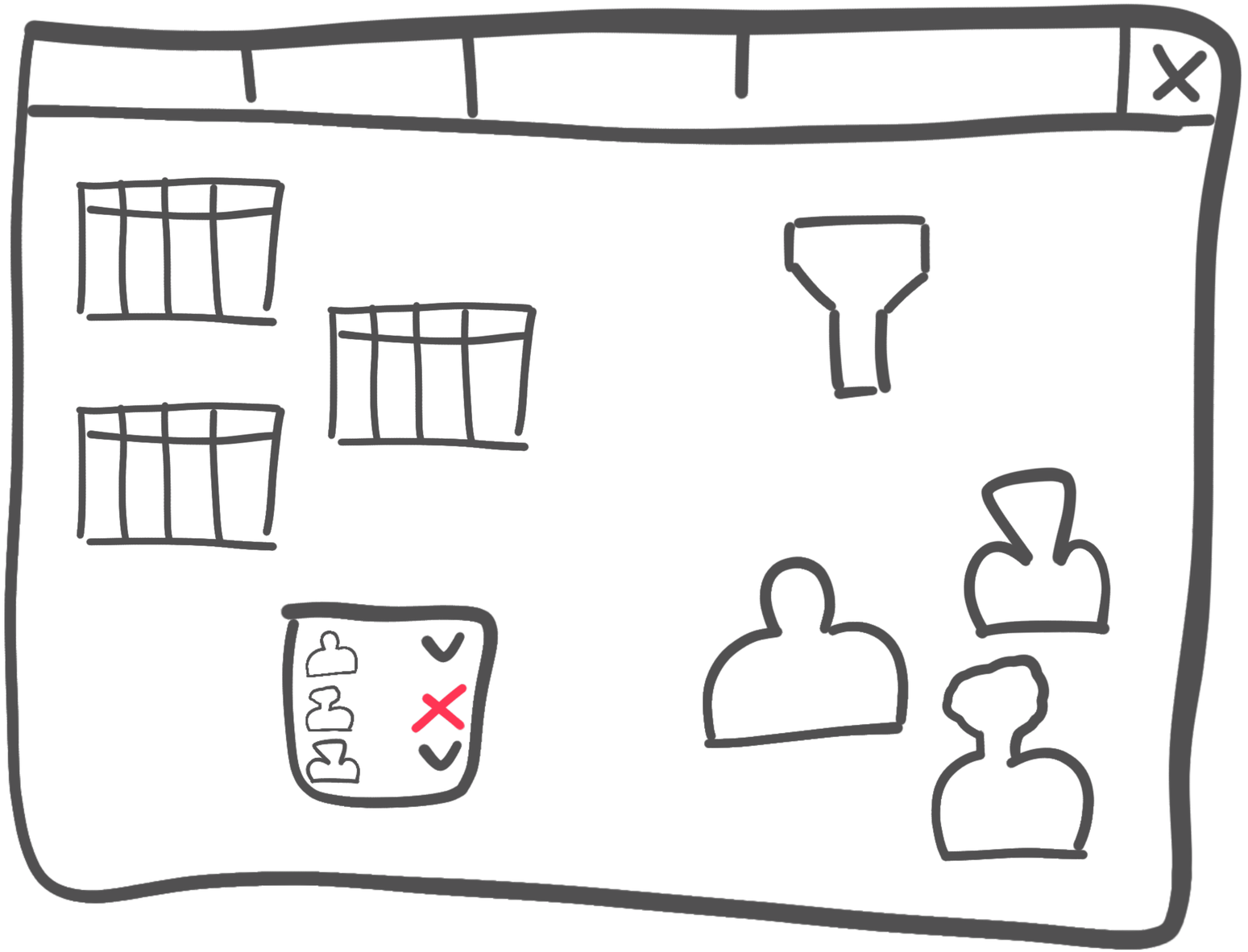
MetaData

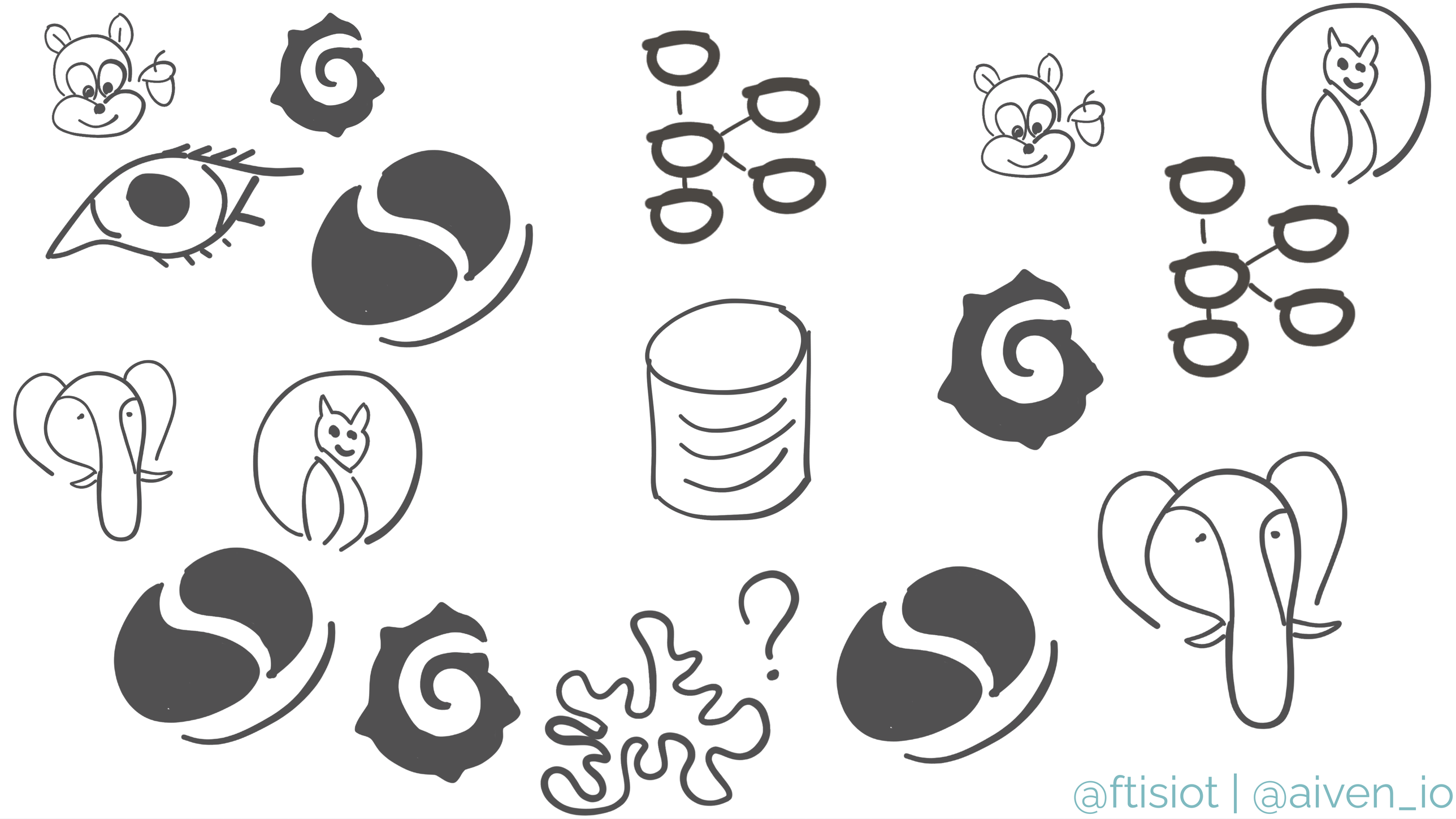


Beyond

https://www.kaggle.com/datasets/unsdsn/world-happiness

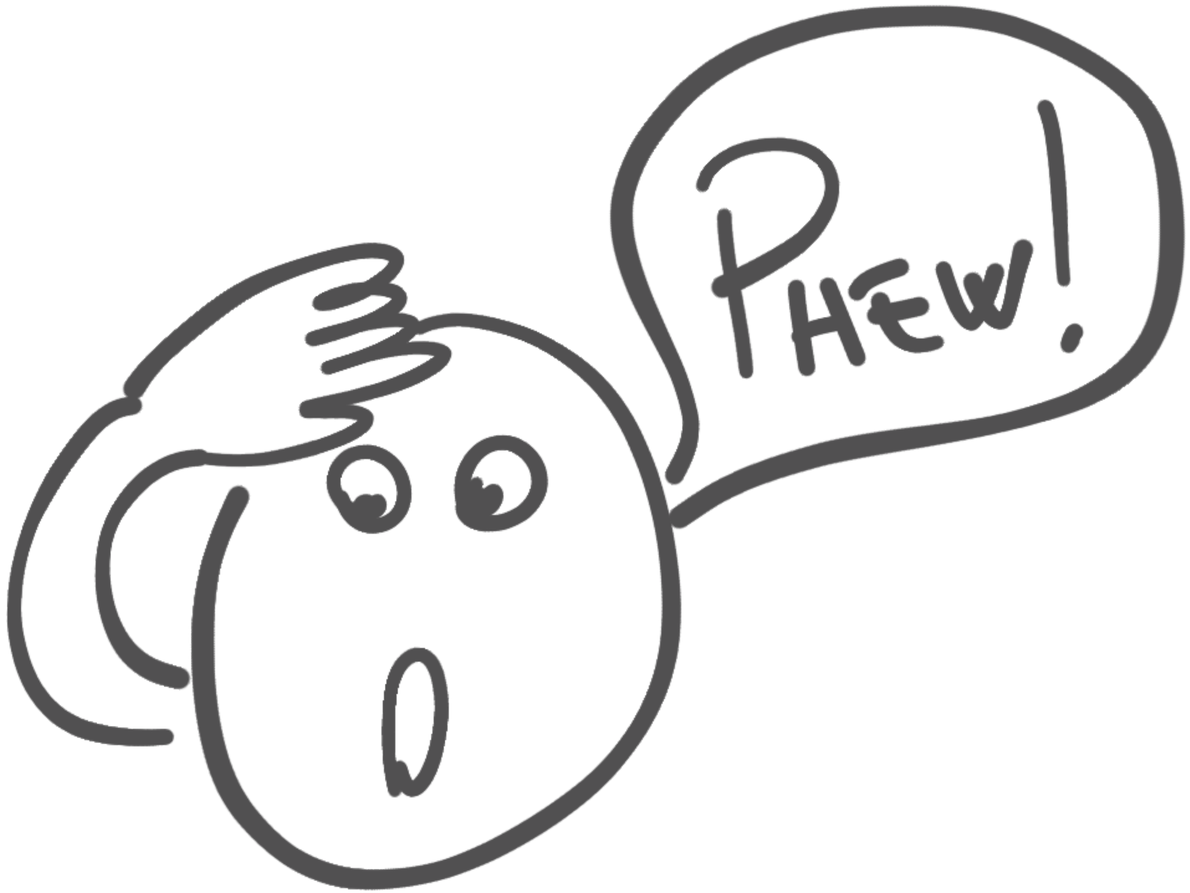
Rank	Country	Score	GDP per capita	Social support
1	Finland	7.769	1.340	1.587
2	Denmark	7.600	1.383	1.573
3	Norway	7.554	1.488	1.582
4	Iceland	7.494	1.380	1.624





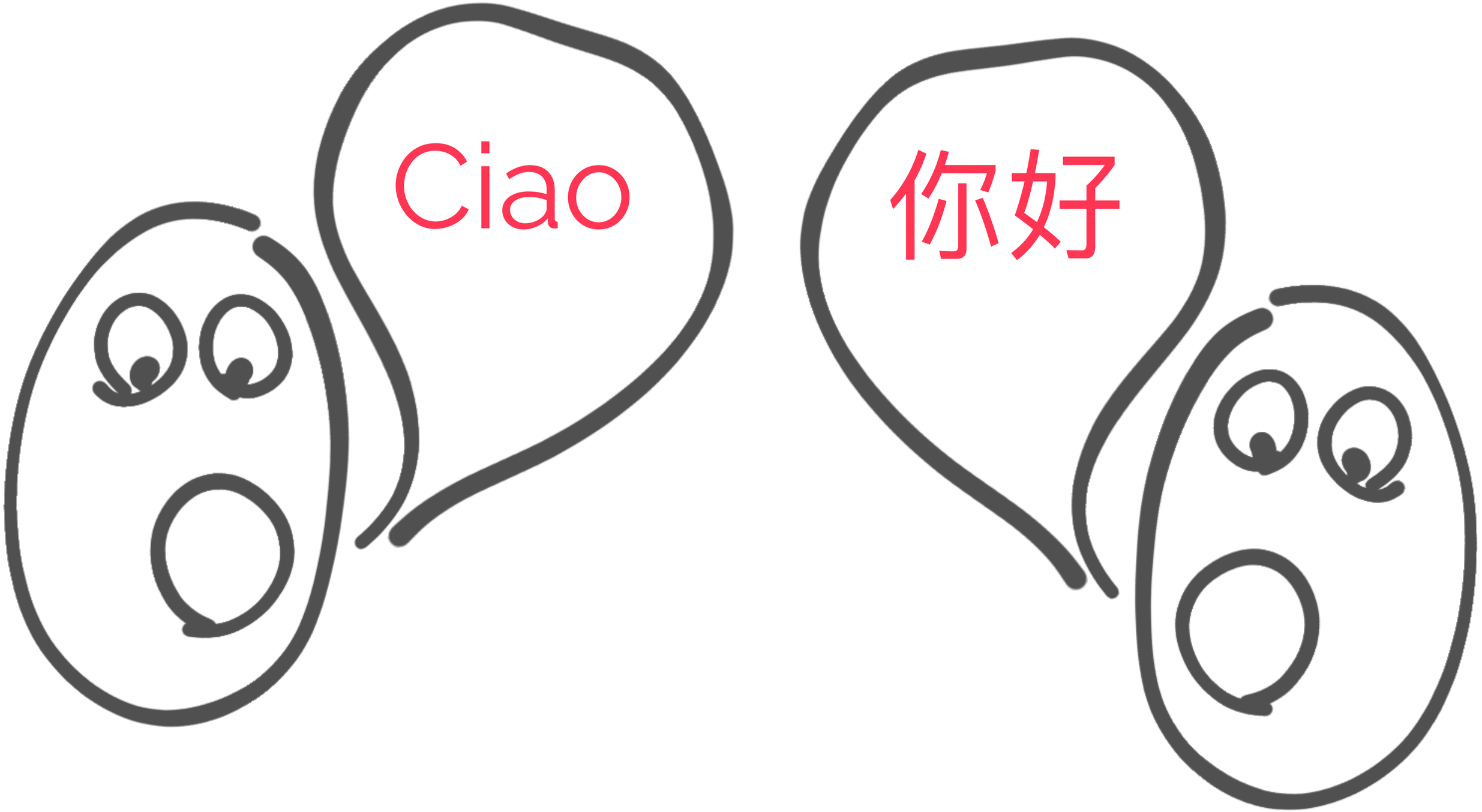
@ftisiot | @aiven_io

Open Source



```
 table_schema | table_name | position |   column_name    |  data_type   | max_length |
--------------+------------+----------+------------------+--------------+------------+
 public       | animal     |        1 | id               | integer      |         32 |
 public       | animal     |        2 | name             | char varying |        100 |
 public       | animal     |        3 | age              | integer      |         32 |
(3 rows)
```

```
[{
    "cleanup_policy": "delete",
    "min_insync_replicas": 1,
    "partitions": 10,
    "replication": 2,
    "retention_bytes": -1,
    "retention_hours": 168,
    "state": "ACTIVE",
    "topic_name": "website_clicks"
  }]
```


Ciao
你好

Database

Column

Schema

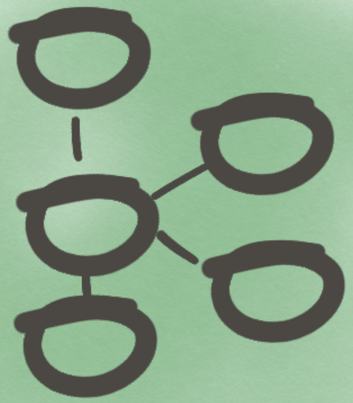
Table

User

Permission

User

Partition



Topic

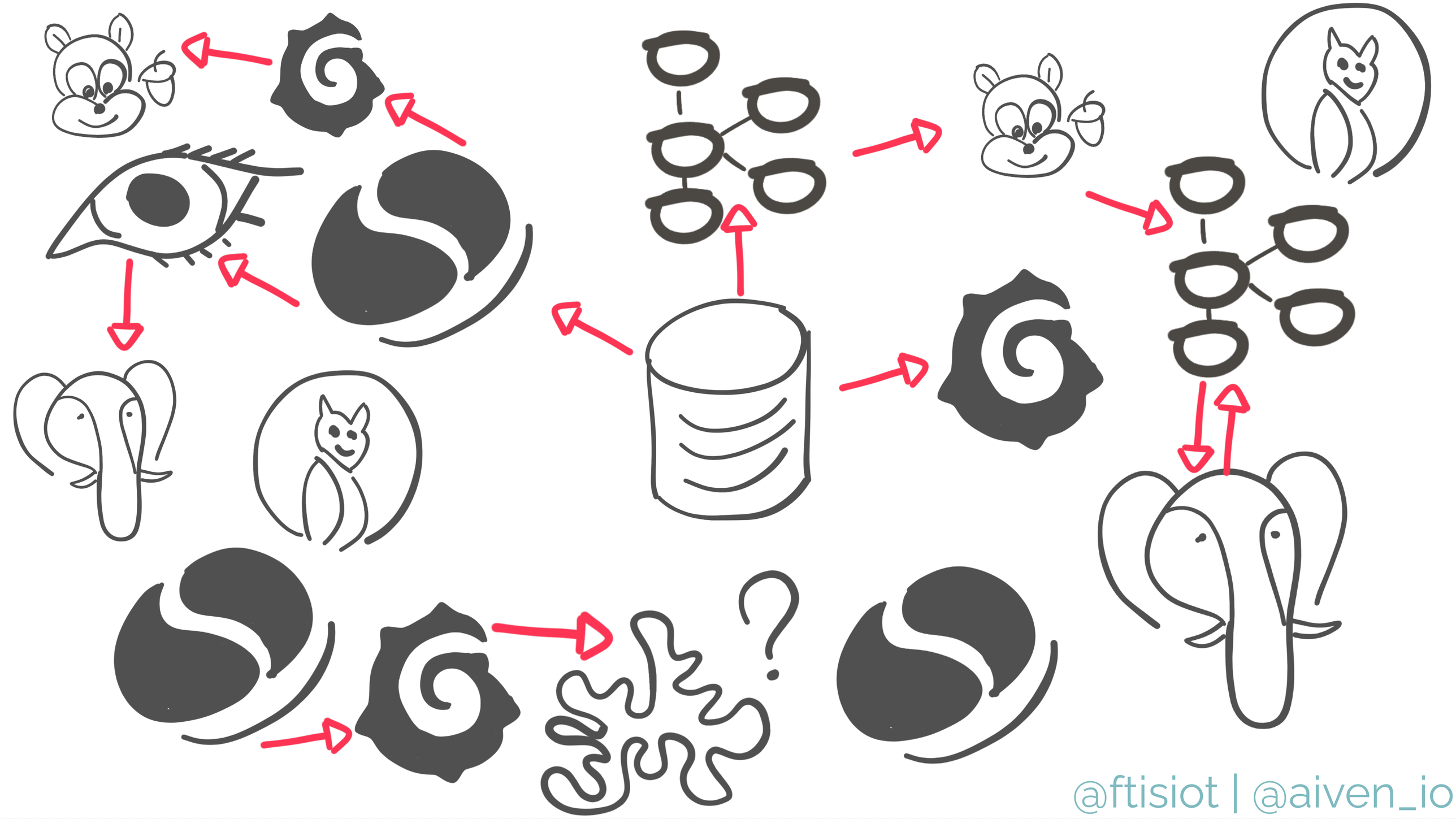
ACL

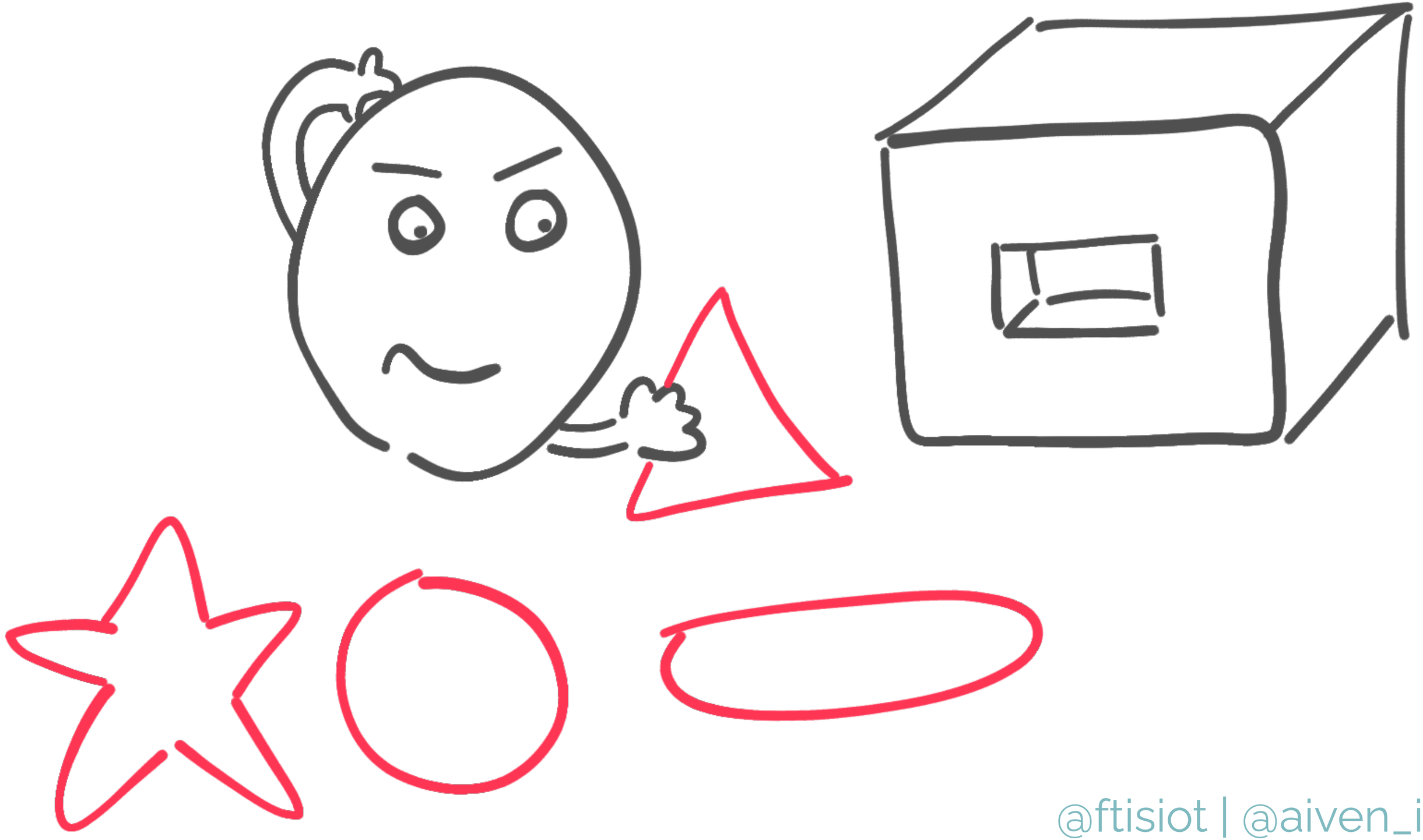
Broker

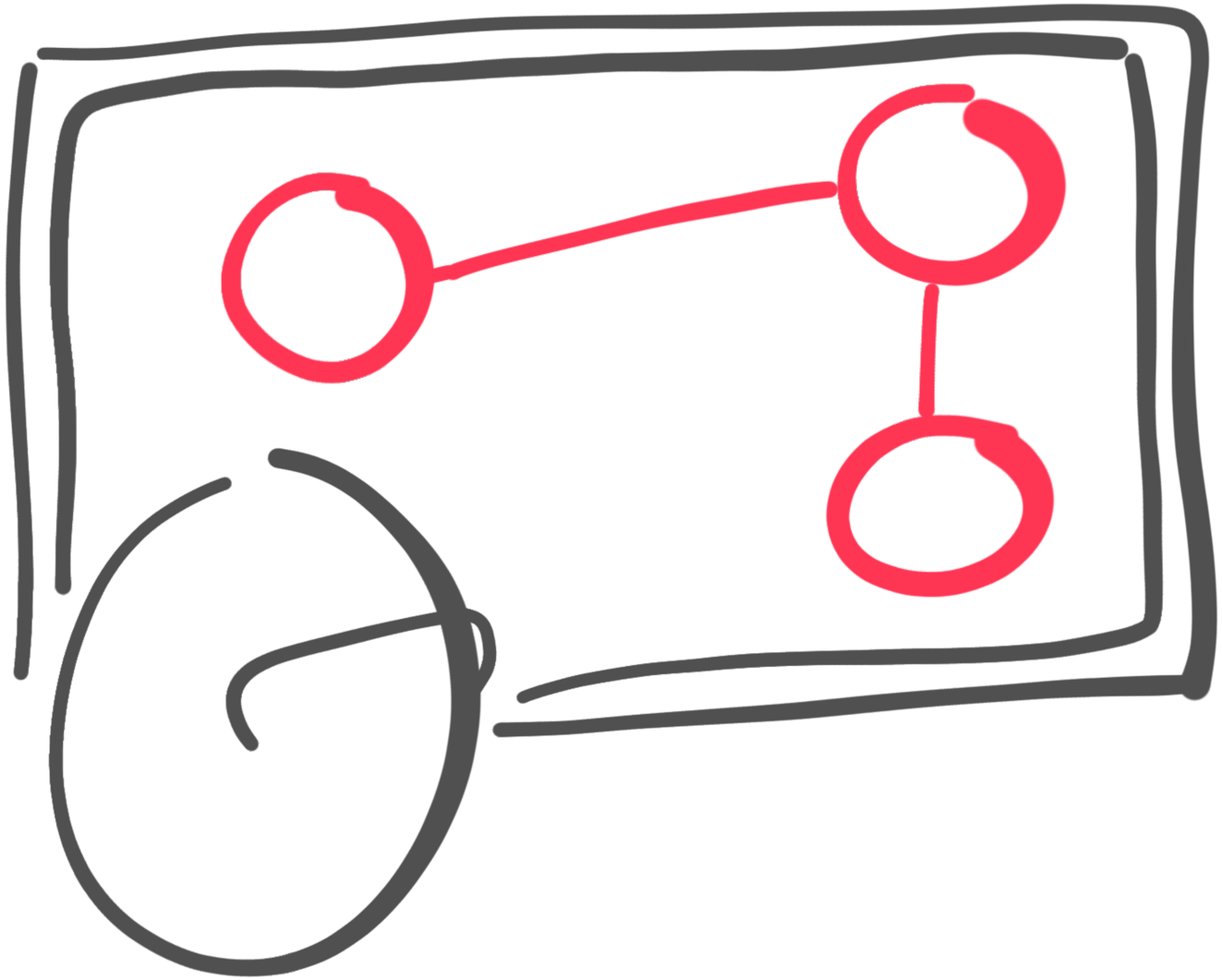
DataSource

Dashboard

Visualisation



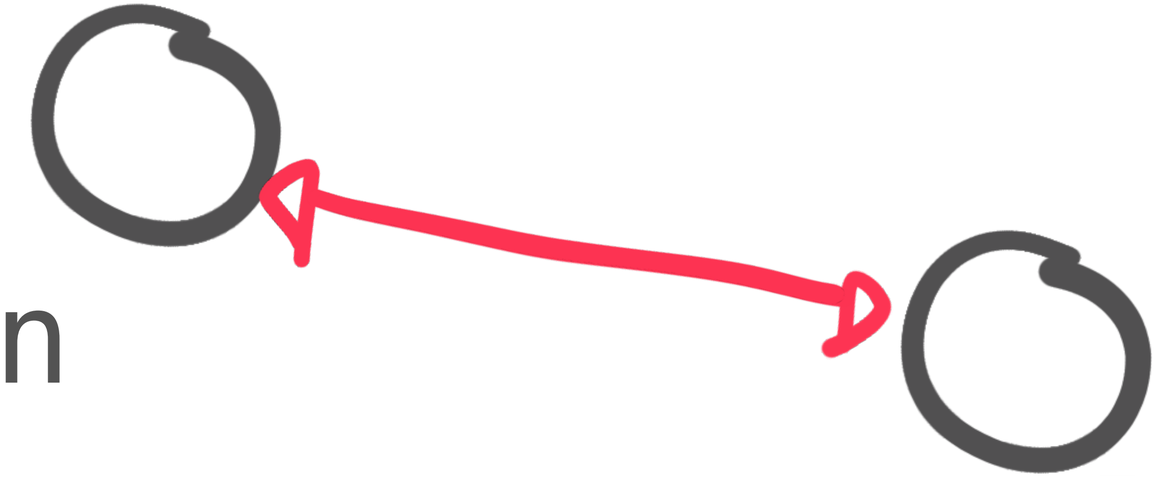




Graph Theory

Node

Edge



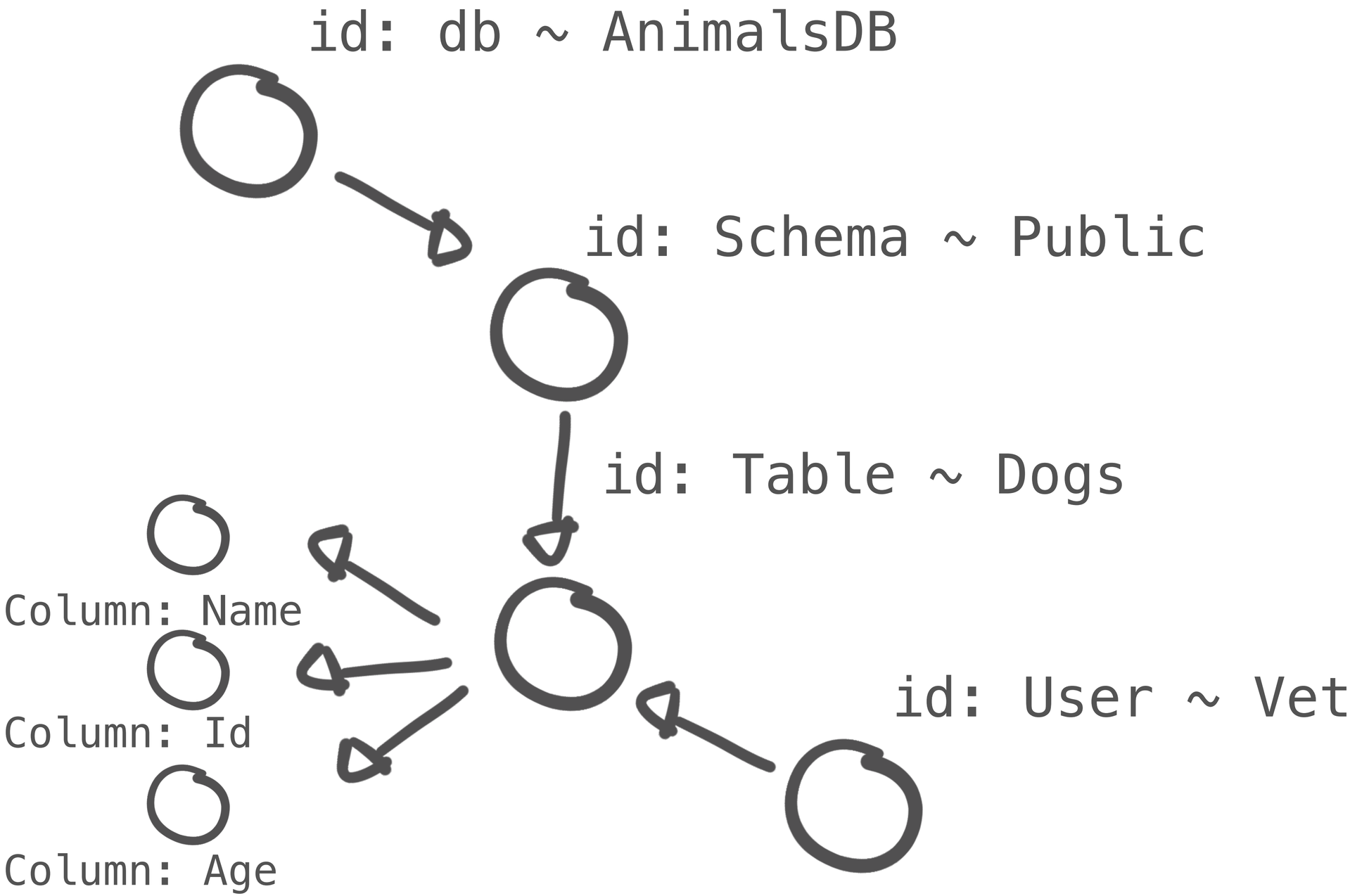
id:S-Bahn

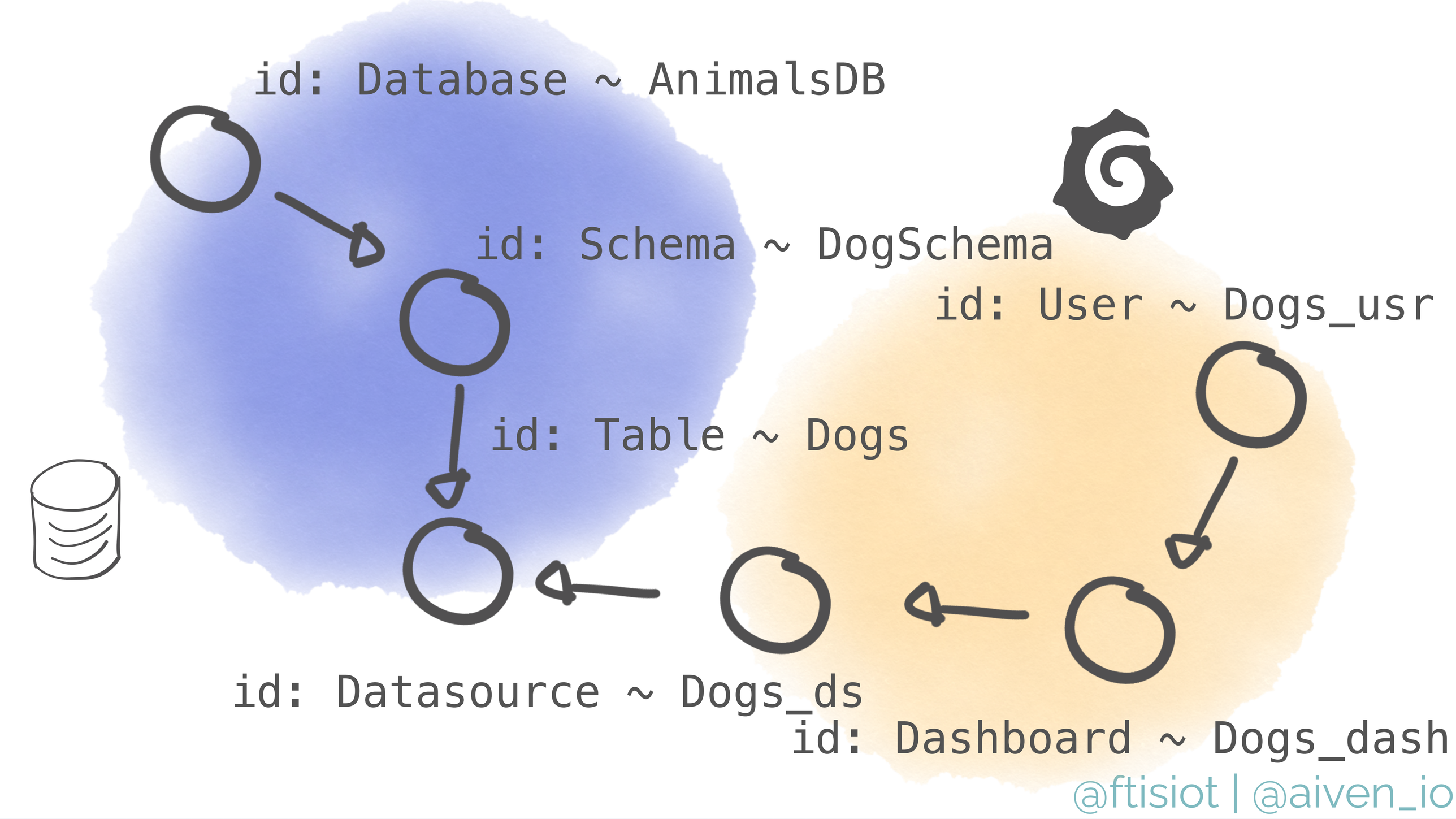
Type: Train line
Sub Type: Light
Age: 98
Interests:
[Transport people,
Polizeieinsatz]

From:S-Bahn To:Fritz
Type: Uses
Start Year: 2015
Frequency: daily
(when it works)

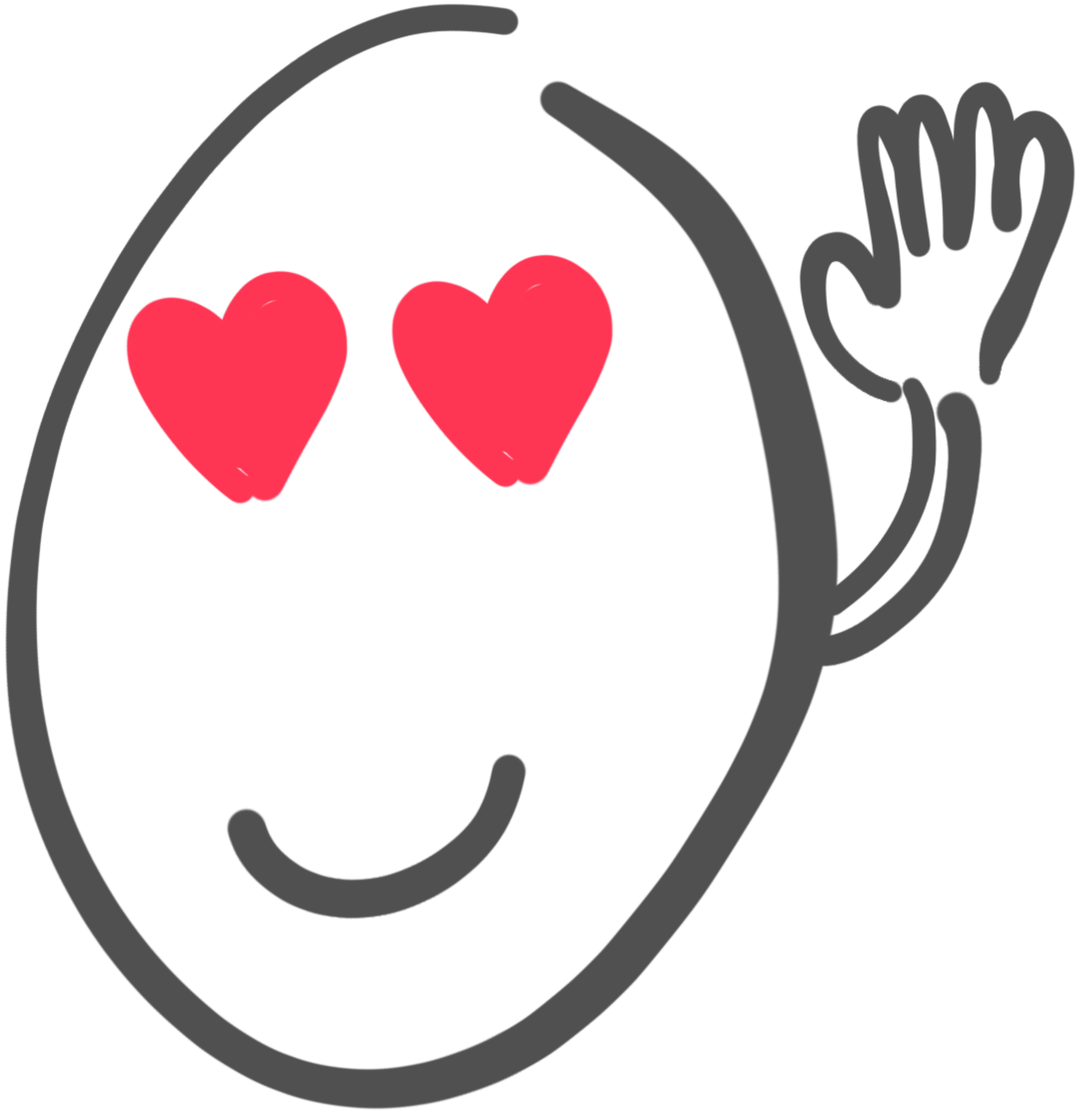
id:Fritz

Type: Human
Job: Informatiker
Age: 40
Interests:
[Curry Wurst,
public transport,
lay down in parks]

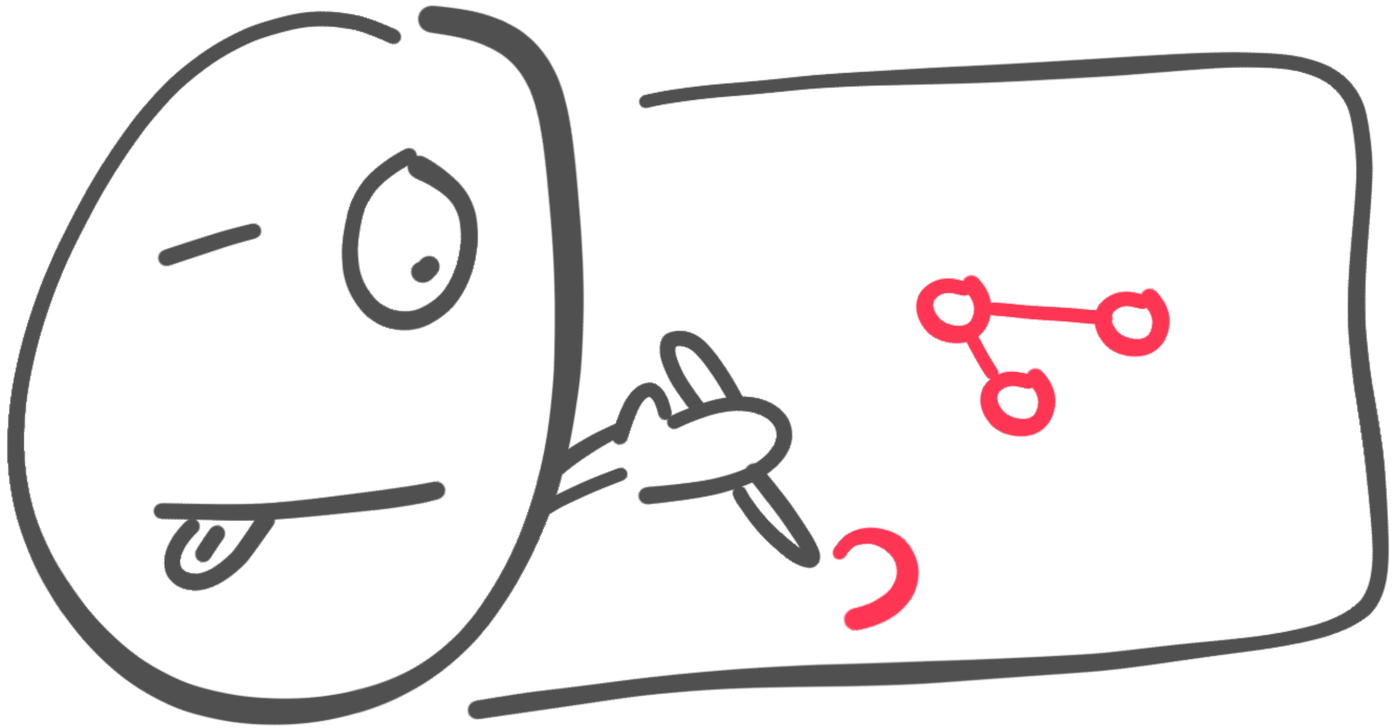

id: db ~ AnimalsDB
id: Schema ~ Public
id: Table ~ Dogs
Column: Name
Column: Id
Column: Age
id: User ~ Vet


id: Database ~ AnimalsDB
id: Schema ~ DogSchema
id: User ~ Dogs_usr
id: Table ~ Dogs
id: Datasource ~ Dogs_ds
id: Dashboard ~ Dogs_dash

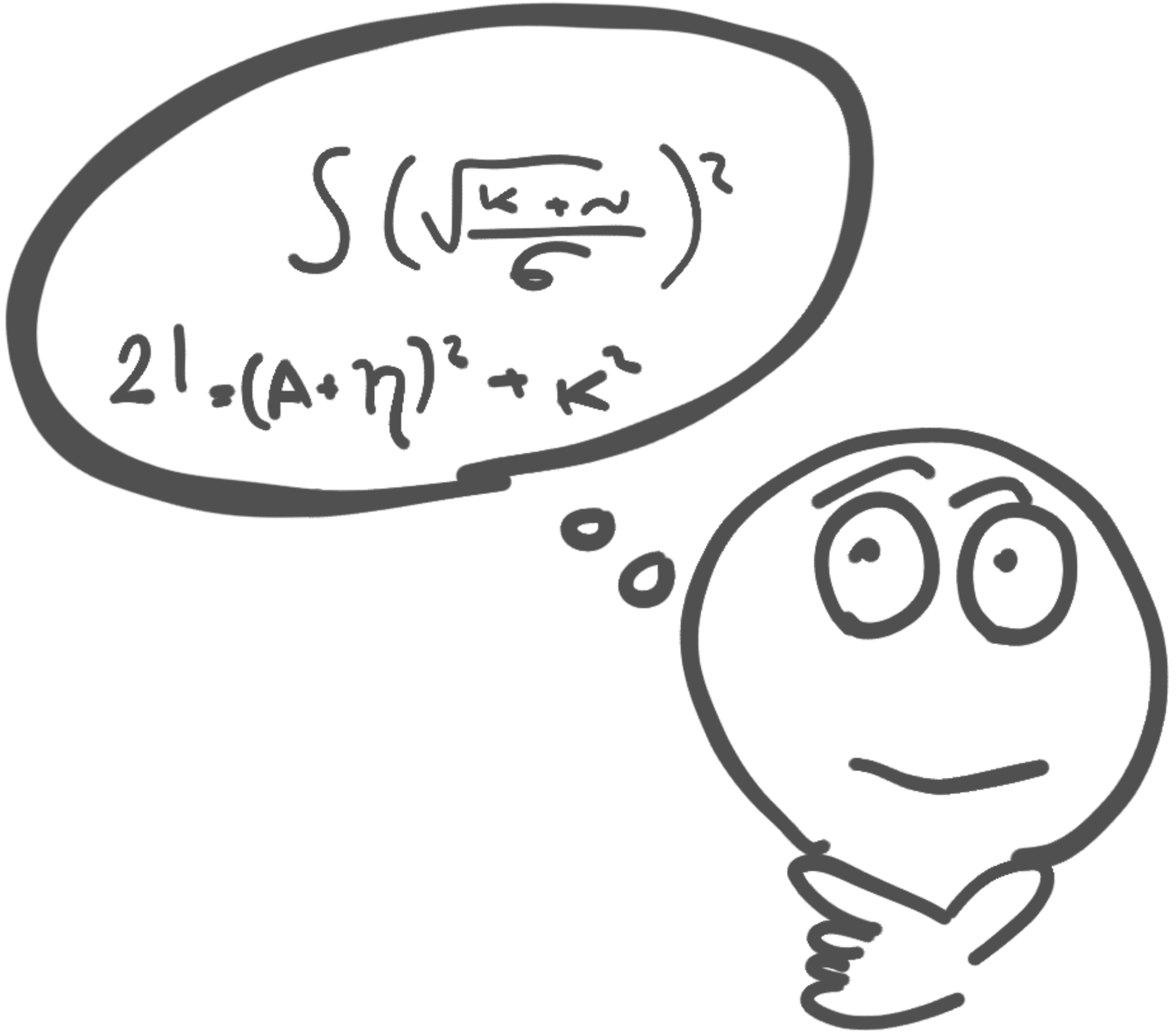
Love the idea!



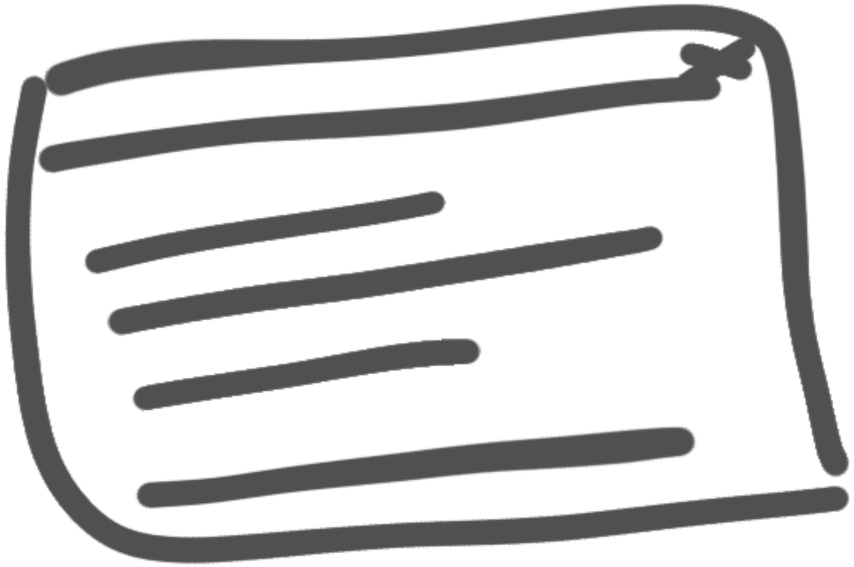
Let's Do It!



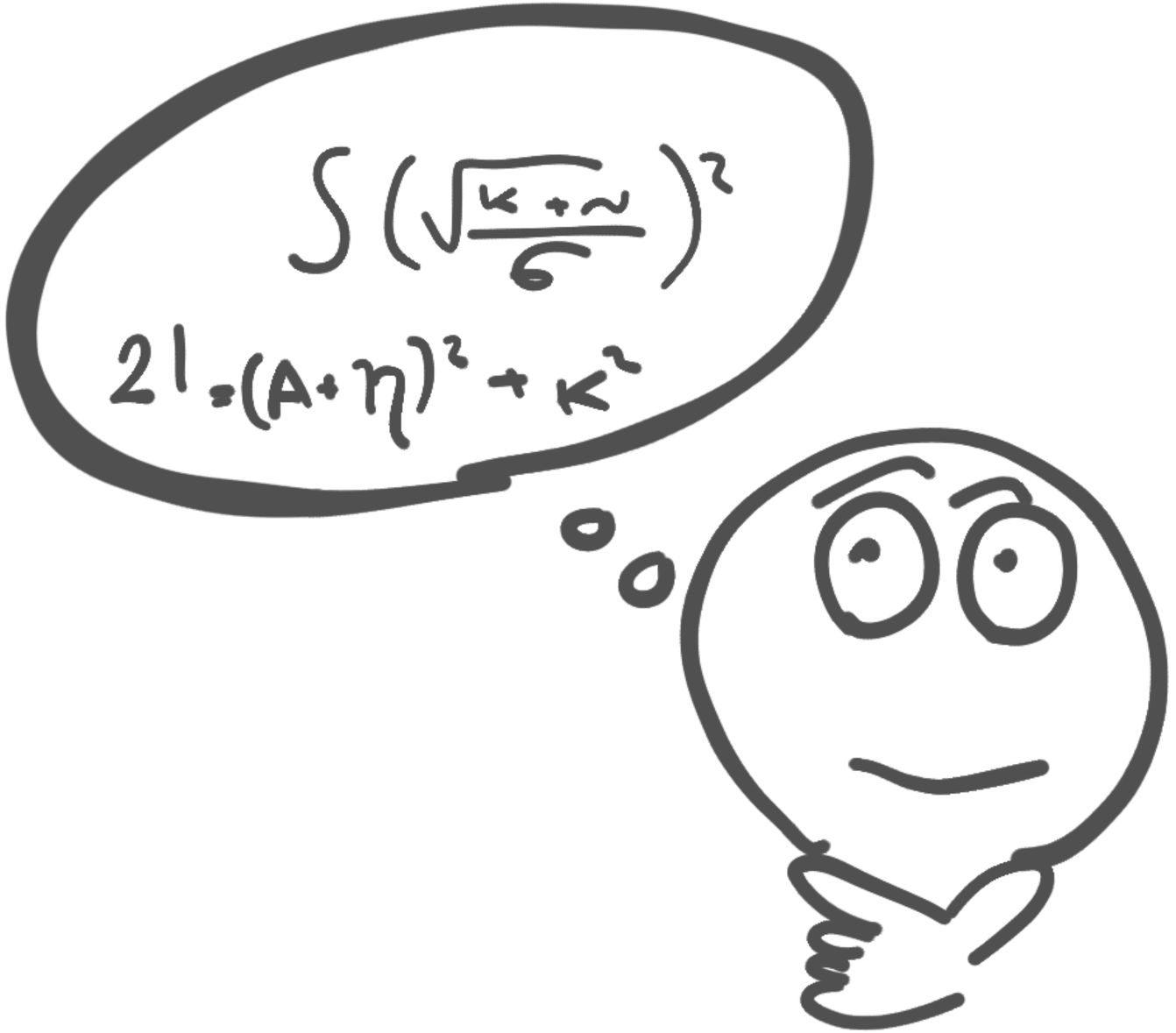
Get The Data?



{REST}



Join The Data?



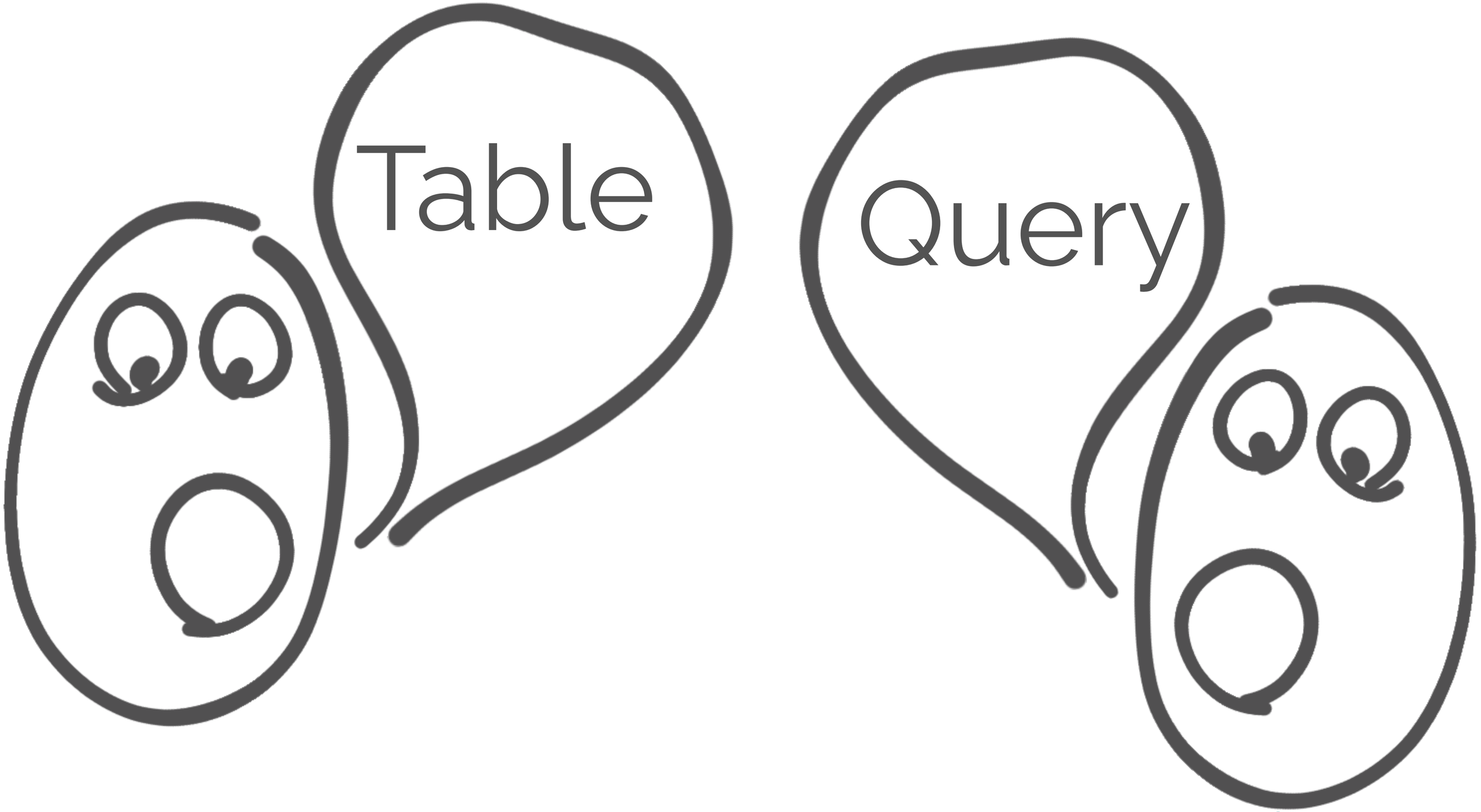


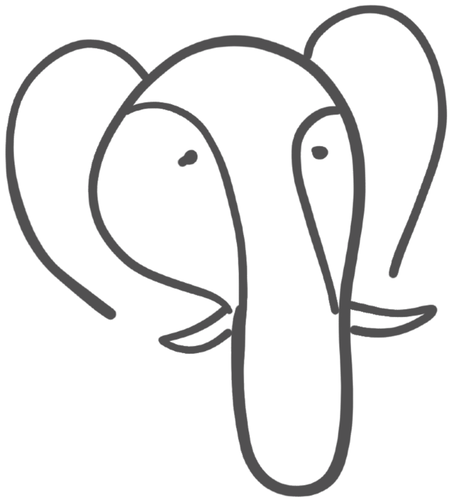
Table
Query


animal.DB.net
192.168.1.153

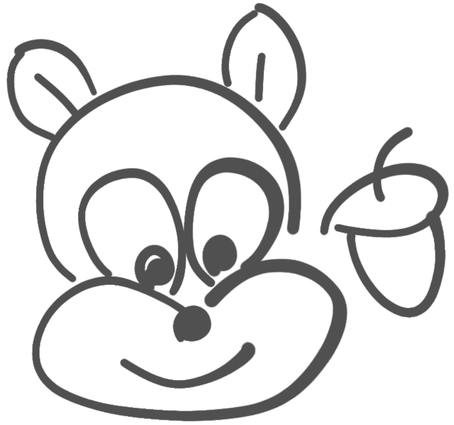
Don't despair

https://github.com/aiven/metadata-parser

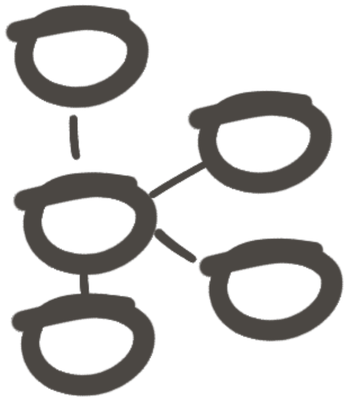
Aiven's Metadata Parser



PostgreSQL



Apache Flink

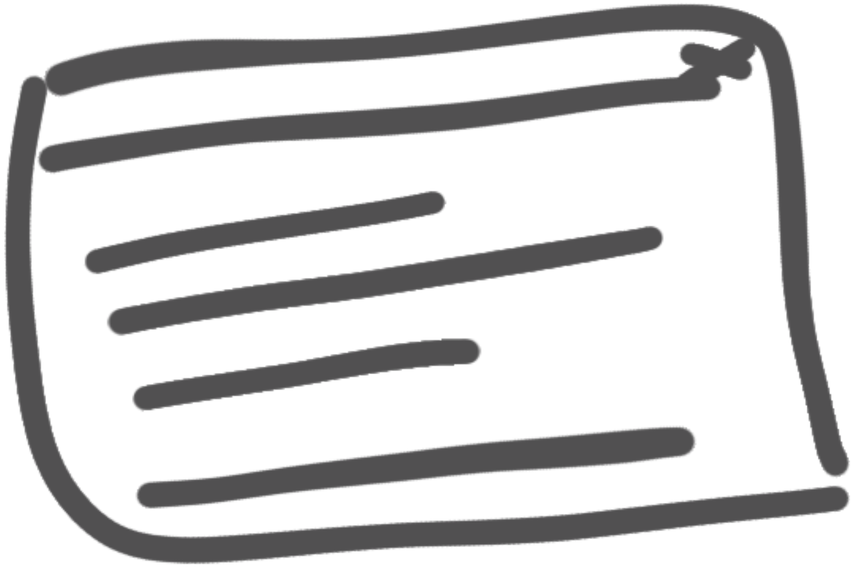


Apache Kafka

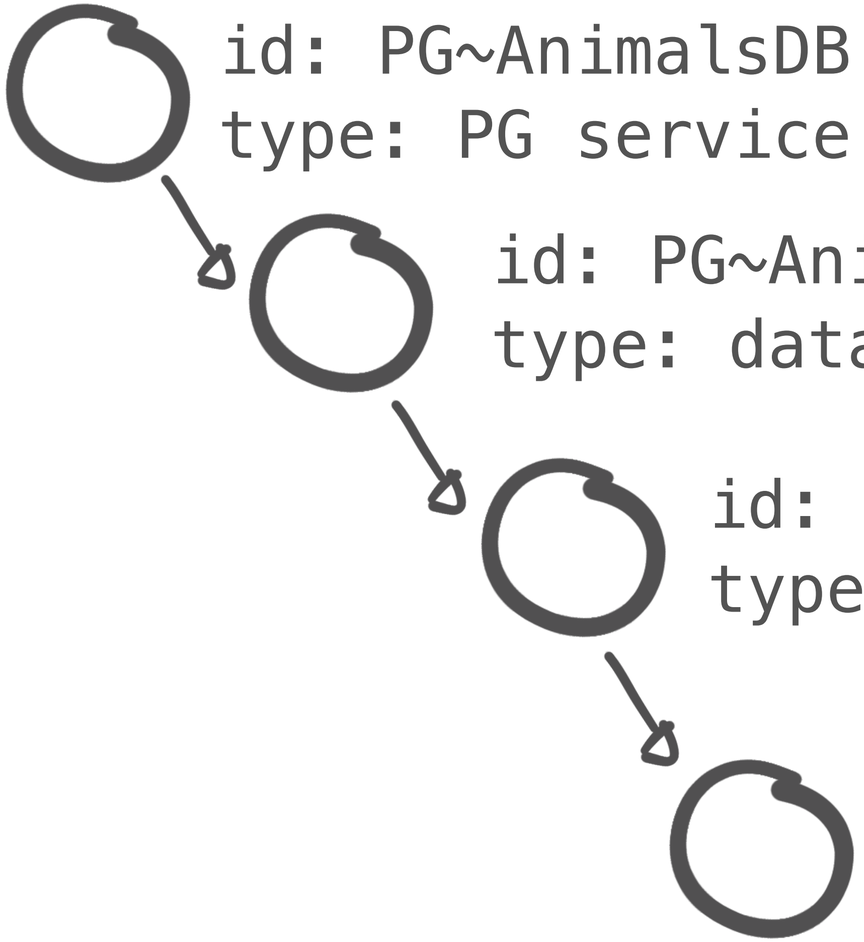
Grafana

OpenSearch

{REST}



```
 table_schema | table_name   | pos | column_name |     data_type      | max_length | is_nullable
--------------+--------------+-----+-------------+--------------------+------------+-------------
 public       | animals      |   1 | id          | integer            |         32 | NO
```



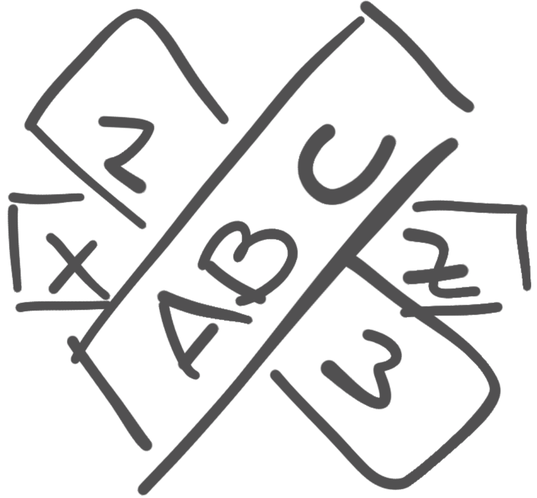
id: PG~AnimalsDB
type: PG service

id: PG~AnimalsDB~schema~public
type: database schema

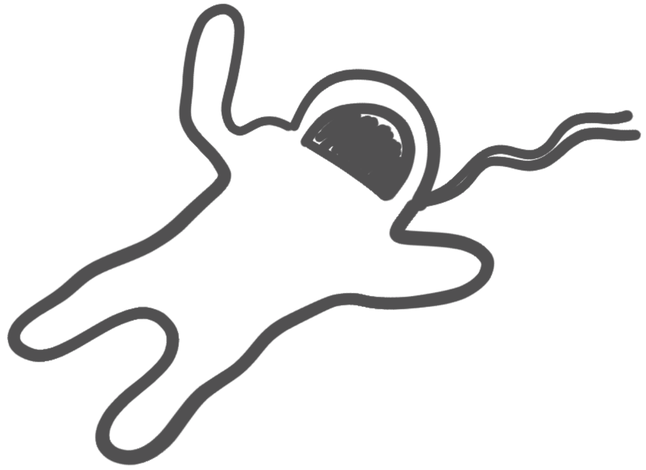
id: PG~AnimalsDB~schema~public~table~animals
type: database table

id: PG~AnimalsDB~schema~public~table~animals~column~id
type: database column
data_type: integer
max_length: 32
is_nullable: NO

Smart Parsing



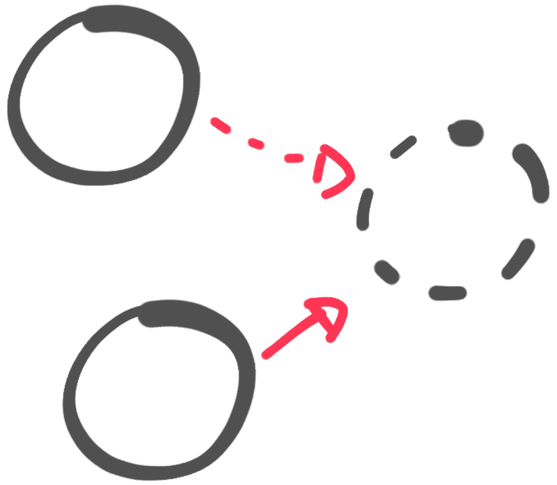
IPs, Aliases



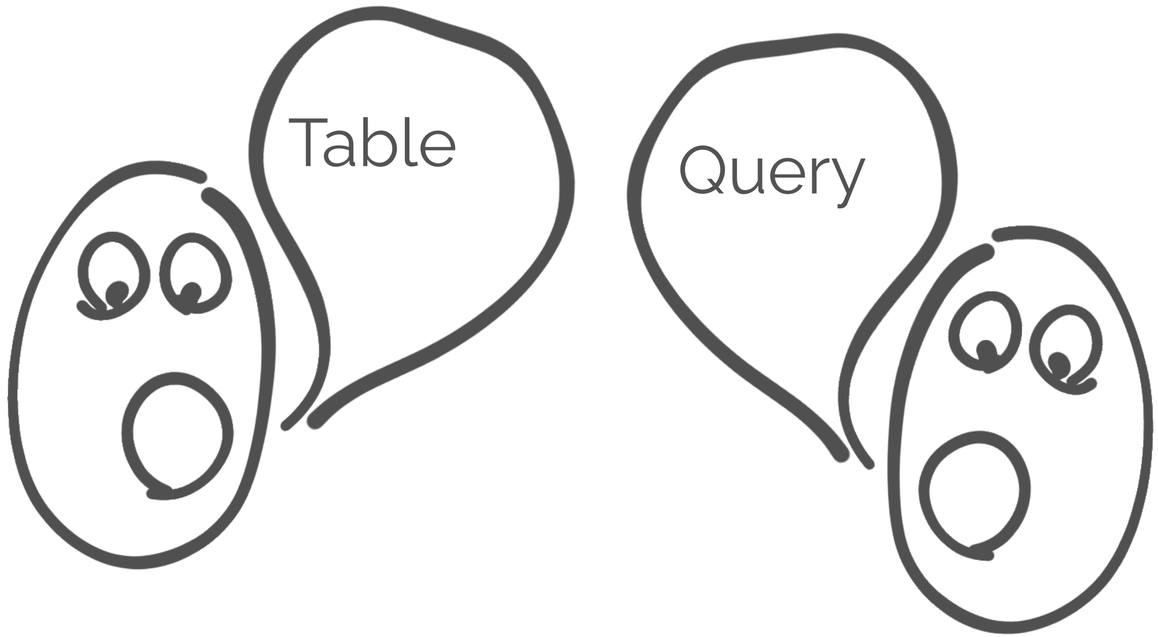
External endpoints

A ~ B ~ C ~ D

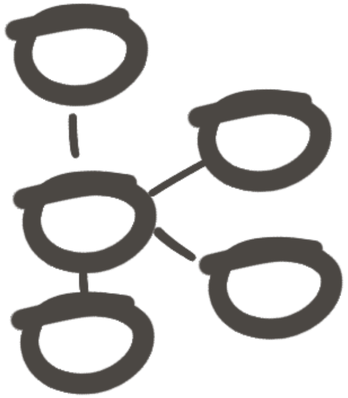
Standard Semantics



Node checks

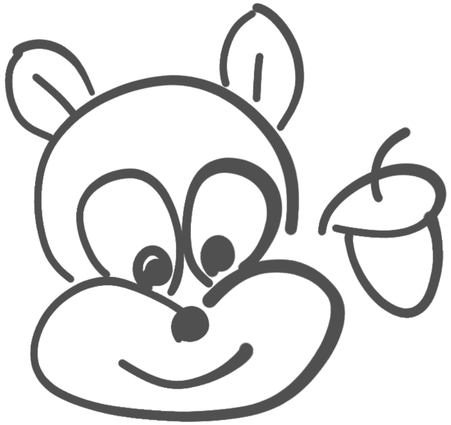


Links Discovery



Kafka Connect

Grafana Datasource



Flink Jobs

```
{
   "name": "cdc-source-pg",
   "connector.class": "io.debezium.connector.postgresql.PostgresConnector",
   "database.hostname": "mydatabasehost.aiven.io",
   "database.port": 13909,
   "database.user": "avnadmin",
   "database.password": "XXXXYYYYYYY",
   "database.dbname": "defaultdb",
   "database.sslmode": "require",
   "plugin.name": "wal2json",
   "slot.name": "test_slot",
   "publication.name": "test_pub",
   "database.server.name": "my_animals",
   "table.include.list": "public.animals"
}
```

id: AnimalsDatabase
type: PostgreSQL service
host: mydatabasehost.aiven.io

id: AnimalsDatabase ~ Public ~ animals
type: database table

id: AnimalsKafka ~ my_animals.public.animals
type: Kafka topic

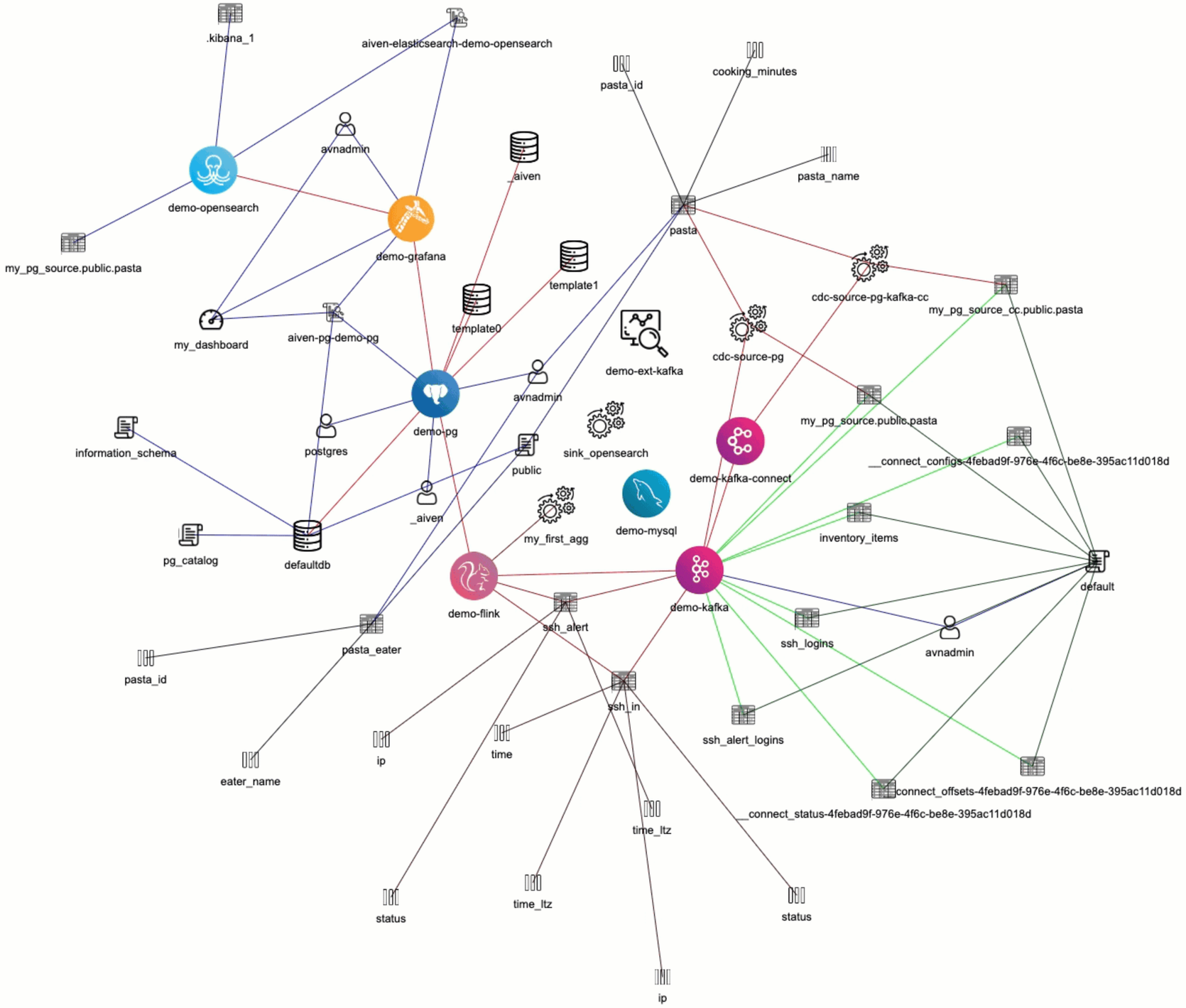
id: AnimalsKafka
type: Apache Kafka service

.DOT

https://graphviz.org/doc/info/lang.html

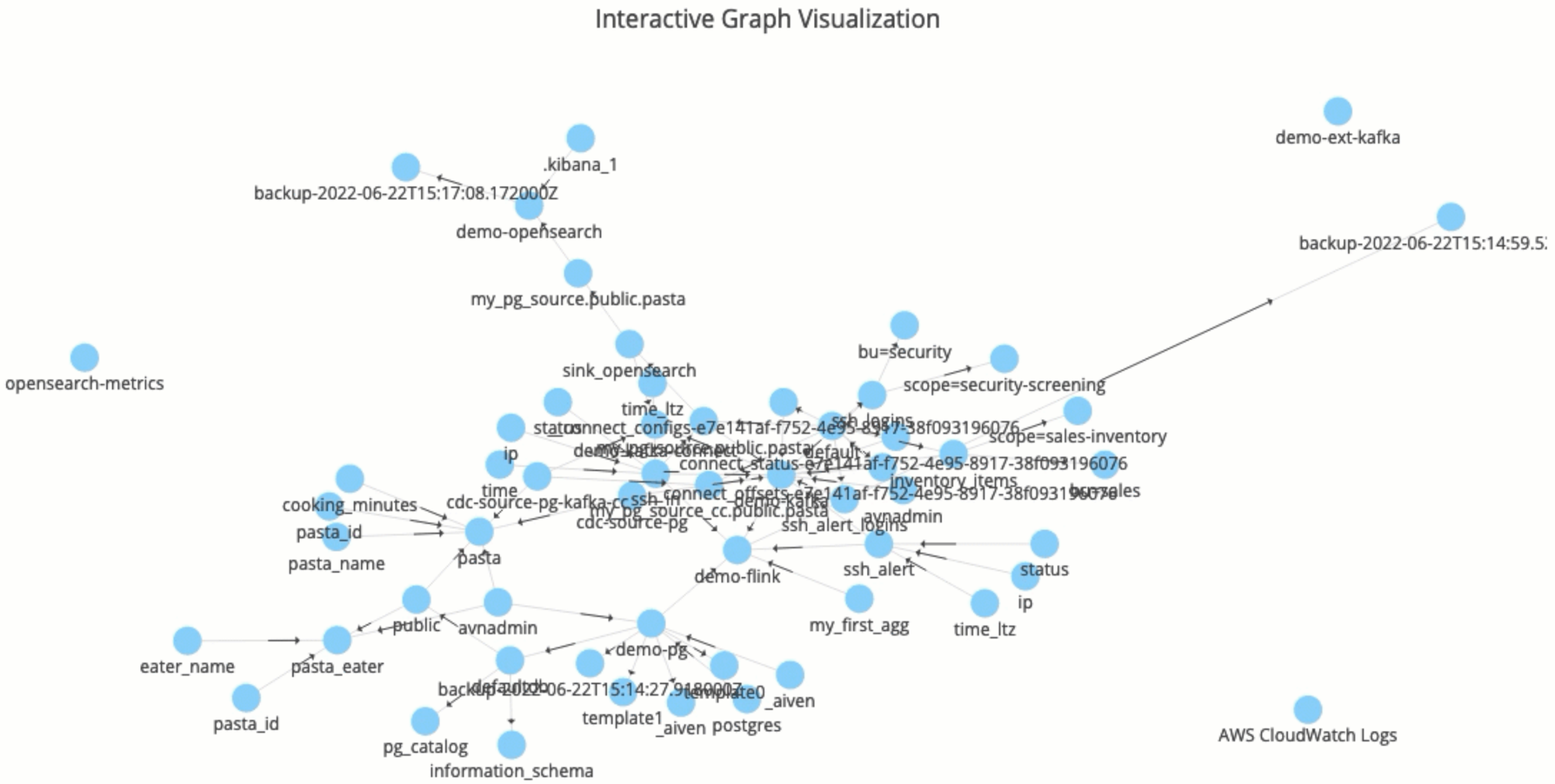
.GML

https://en.wikipedia.org/wiki/Geography_Markup_Language


.kibana_1
aiven-elasticsearch-demo-opensearch
avnadmin
demo-opensearch
demo-grafana
my_pg_source.public.pasta
my_dashboard
aiven-pg-demo-pg
template0
template1
_aiven
demo-pg
avnadmin
information_schema
postgres
public
sink_opensearch
_aiven
pg_catalog
defaultdb
my_first_agg
demo-mysql
demo-flink
demo-kafka
pasta_eater
pasta_id
eater_name
ssh_alert
ip
time
ssh_in
time_ltz
status
time_ltz
status
ip
pasta_id
cooking_minutes
pasta_name
pasta
demo-ext-kafka
cdc-source-pg-kafka-cc
my_pg_source_cc.public.pasta
cdc-source-pg
my_pg_source.public.pasta
demo-kafka-connect
__connect_configs-4febad9f-976e-4f6c-be8e-395ac11d018d
inventory_items
default
ssh_logins
avnadmin
ssh_alert_logins
connect_offsets-4febad9f-976e-4f6c-be8e-395ac11d018d
__connect_status-4febad9f-976e-4f6c-be8e-395ac11d018d

Node To Search Input the Node Id to visualize.

Select a node



Select Node Types

To filter particular services/node types.

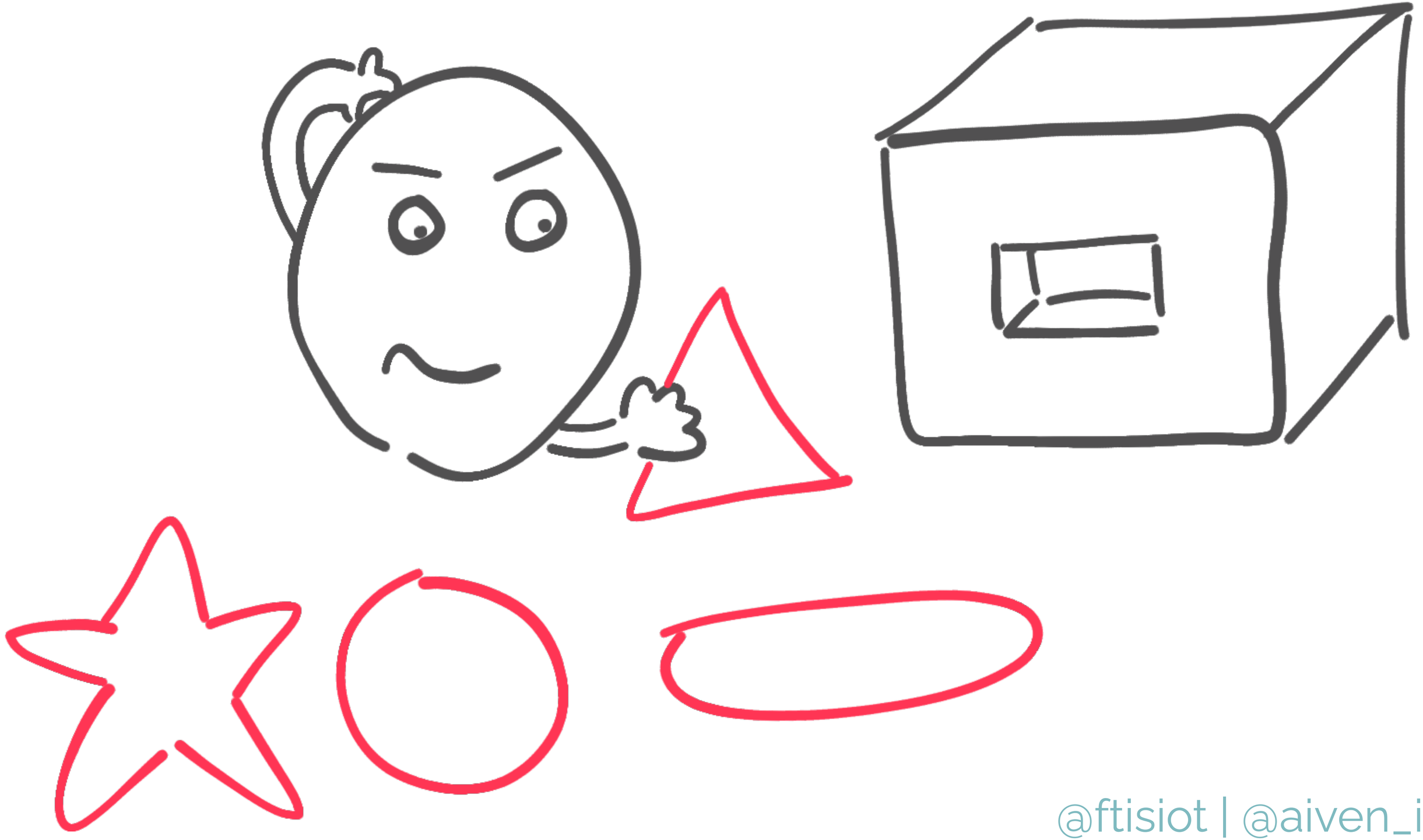
- [ ] backup
- [ ] database
- [ ] external_endpoint
- [ ] flink job
- [ ] flink table
- [ ] flink table column
- [ ] index
- [ ] kafka-connect
- [ ] schema
- [ ] service
- [ ] table
- [ ] table column
- [ ] tag
- [ ] topic
- [ ] topic-acl
- [ ] user

Hover Data

Mouse over values in the graph.


```
{
"id": "kafka~demo-
kafka~topic~inventory_items",
"service_type": "kafka",
"type": "topic",
"cleanup_policy": "delete",
"label": "inventory_items"
```

How about PostgreSQL ?



Tables
+
JSONB
+
Recursive Queries

Nodes + Edges

Nodes

Id

pg~demo-pg~schema~public~table~pasta~column~pasta_name

JSON payload

```
{
"type": "table column",
"label": "pasta_name",
"data_type": "character varying",
"is_nullable": "YES",
"service_type": "pg"
}
```

Edges

From Id	To Id	JSON payload
`demo-grafana`	`demo-pg`	`{ "main_type": "integration", "int_type": "datasource", "label": "datasource", "integration_id": "XYZ" }`

Enhancements

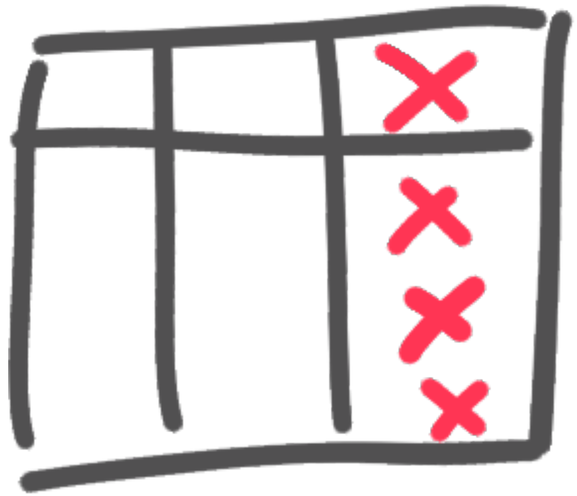
GIN Index

```
CREATE INDEX metadata_parser_nodes_idx ON
metadata_parser_nodes USING GIN (json_content);
```

GDPR

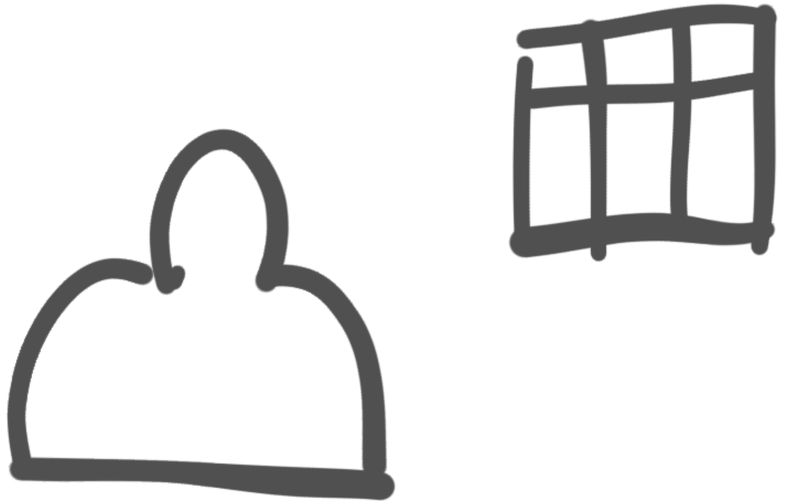
```
with recursive paths (id, last_label, last_type, last_service_type, list_of_edges, nr_items) as (
    select
        id,
        json_content ->> 'label',
        json_content ->> 'type',
        json_content ->> 'service_type',
        ARRAY[((n.json_content ->> 'type') || ':' || (n.json_content ->> 'label'))],
        1
    from metadata_parser_nodes n
    where json_content ->> 'label' = 'pasta'
    UNION ALL
    select
        n.id,
        n.json_content ->> 'label',
        n.json_content ->> 'type',
        n.json_content ->> 'service_type',
        list_of_edges || ((n.json_content ->> 'type') || ':' || (n.json_content ->> 'label')),
        nr_items + 1
    from paths p
    join metadata_parser_edges e on p.id = e.source_id
    join metadata_parser_nodes n on e.destination_id = n.id
    ) CYCLE id SET is_cycle USING items_ids
select last_label, last_type, last_service_type, list_of_edges
from paths
where is_cycle = False;
```

```
      last_label       |   last_type   | last_service_type |                                  list_of_edges
-----------------------+---------------+-------------------+-------------------------------------------------------------------------
 pasta                 | table         | pg                | {table:pasta}
 public                | schema        | pg                | {table:pasta,schema:public}
 avnadmin              | user          | pg                | {table:pasta,user:avnadmin}
 pasta_id              | table column  | pg                | {table:pasta,"table column:pasta_id"}
 cooking_minutes       | table column  | pg                | {table:pasta,"table column:cooking_minutes"}
 pasta_name            | table column  | pg                | {table:pasta,"table column:pasta_name"}
 cdc-source-pg         | kafka-connect | kafka             | {table:pasta,kafka-connect:cdc-source-pg}
 cdc-source-pg-kafka-cc | kafka-connect | kafka_connect     | {table:pasta,kafka-connect:cdc-source-pg-kafka-cc}
 defaultdb             | database      | pg                | {table:pasta,schema:public,database:defaultdb}
 pasta_eater           | table         | pg                | {table:pasta,schema:public,table:pasta_eater}
(10 rows)
```



```
where json_content ->> 'label' not in ('pasta_id','pasta_eater')
```

```
where id <> 'pg~demo-pg~schema~public~table~pasta~column~pasta_id'
```

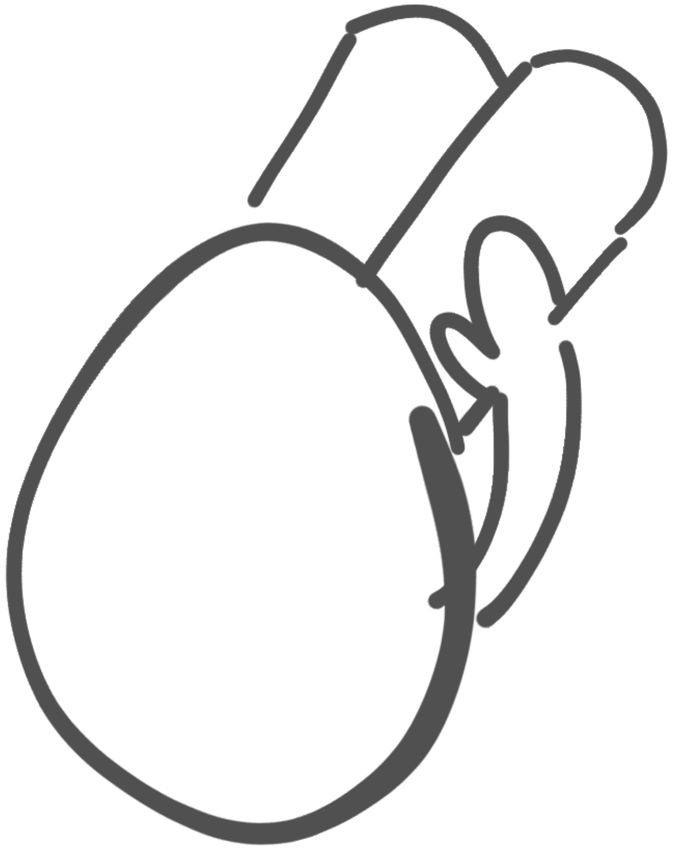


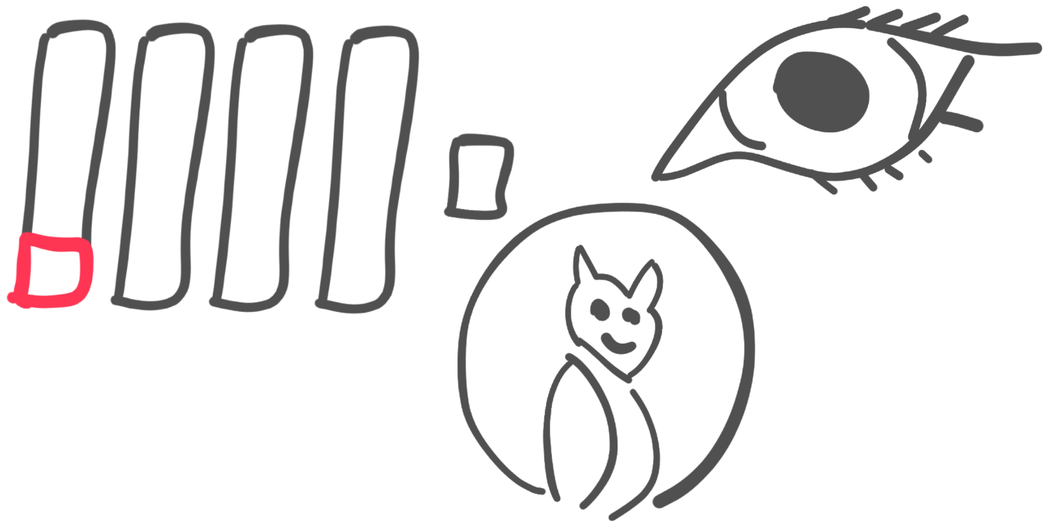
```
where json_content ->> 'type' = 'user'

where json_content @> '{"type":"user"}'
```

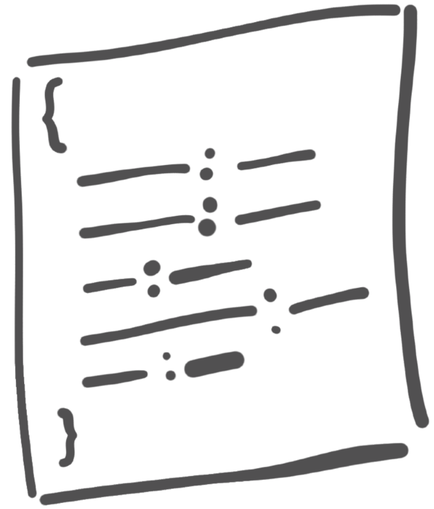
```
 last_label | last_type | last_service_type |
------------+-----------+-------------------+-----------------------------------
 avnadmin   | user      | pg                | {table:pasta,user:avnadmin}
 avnadmin   | user      | pg                | {table:pasta,schema:public,table:pasta_eater,user:avn
 avnadmin   | user      | kafka             | {table:pasta,kafka-connect:cdc-source-pg-kafka-cc,top
 avnadmin   | user      | kafka             | {table:pasta,kafka-connect:cdc-source-pg,topic:my_pg_
 avnadmin   | user      | opensearch        | {table:pasta,kafka-connect:cdc-source-pg,topic:my_pg_
 avnadmin   | user      | grafana           | {table:pasta,schema:public,database:defaultdb,datasou
 avnadmin   | user      | opensearch        | {table:pasta,kafka-connect:cdc-source-pg-kafka-cc,top
(8 rows)
```

What's next?

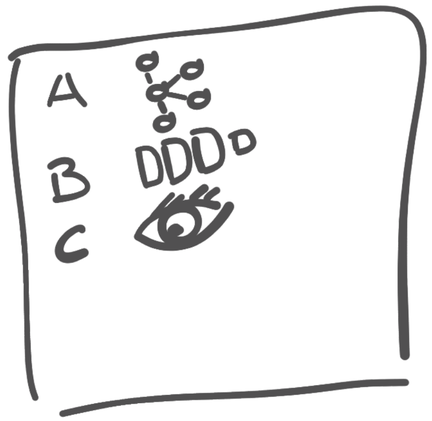




Tools Coverage

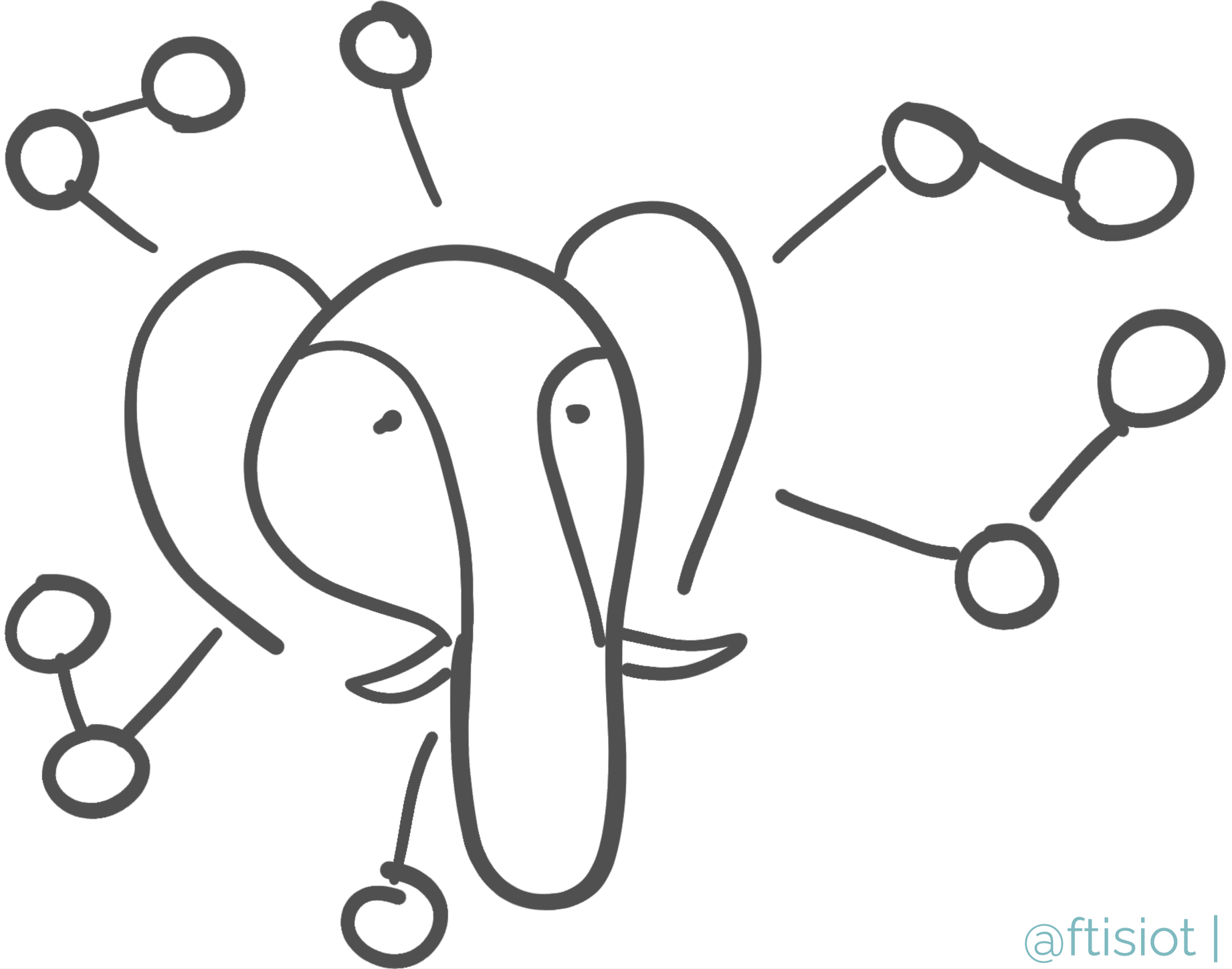


Better Parsing



List of endpoints

Apache AGE



Wanna Know More?

https://ftisiot.net/talks/pg-metadata/