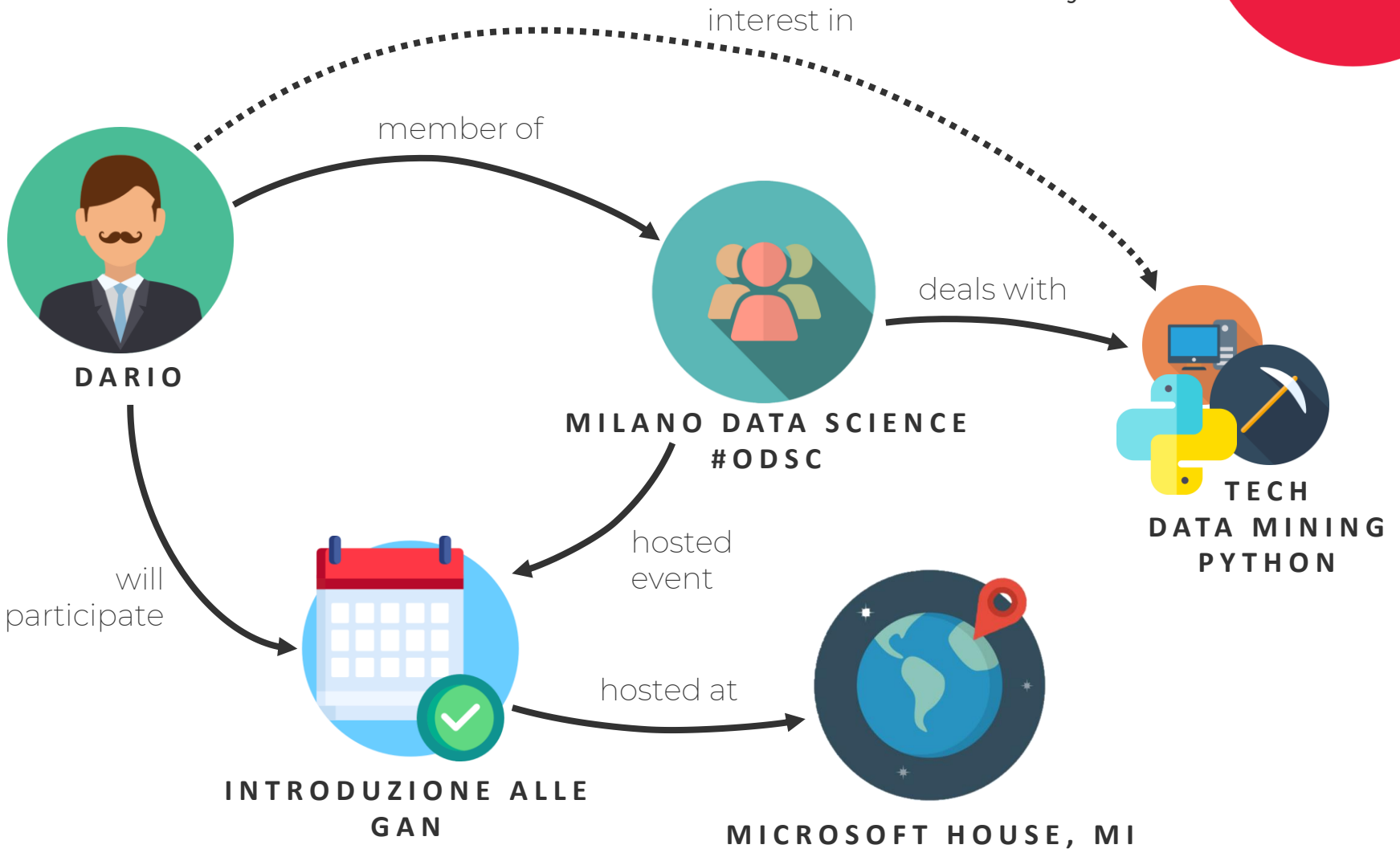
UNIVERSITÀ DEGLI STUDI DI MILANO BICOCCA

social network analysis

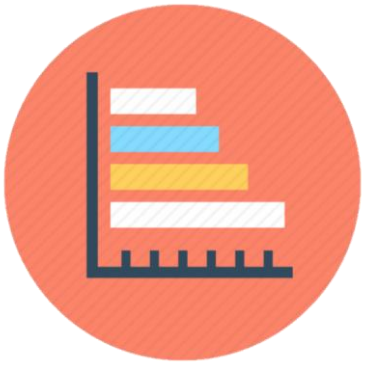
Dario Bertazioli
Fabrizio D'Intinosante
Massimiliano Perletti

introduction

available in **186** countries
40 millions users
320k active groups
12k daily events



goals



quantitative measures

temporal distribution of the events

event area of influence

recommender system efficiency analysis

covered points

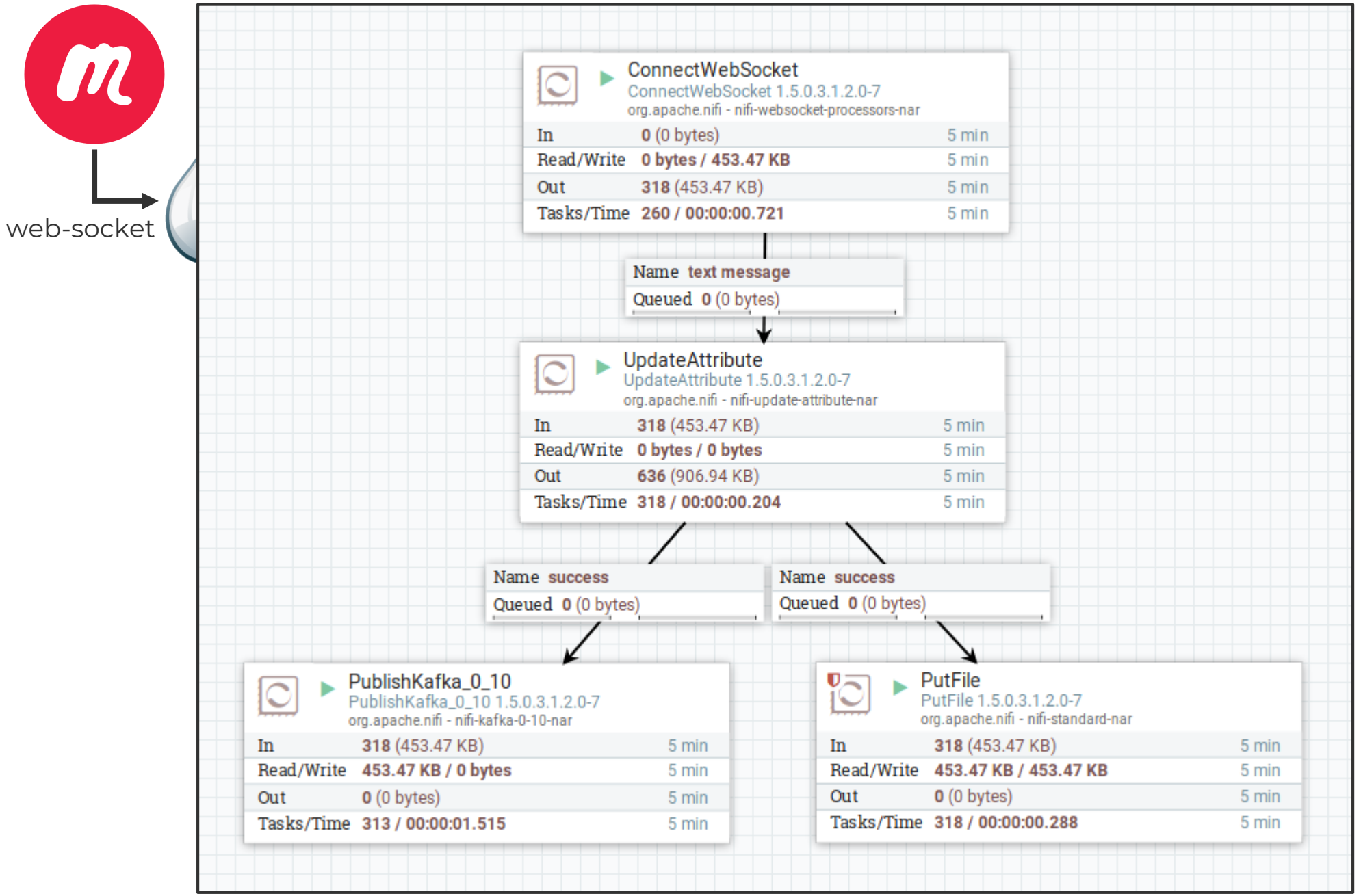
from Big Data V(s)

volume
more than 2
GB database

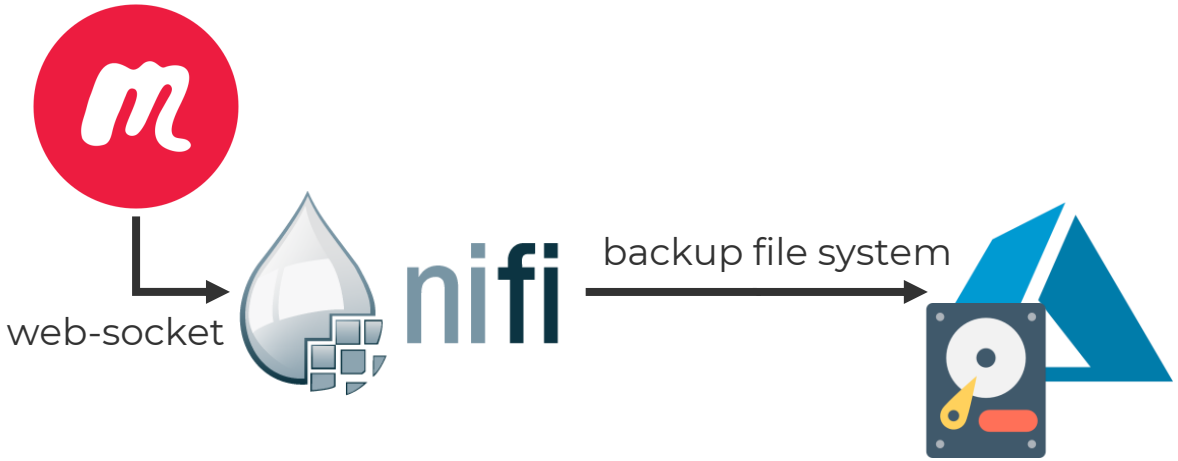
velocity
streaming data
acquisition

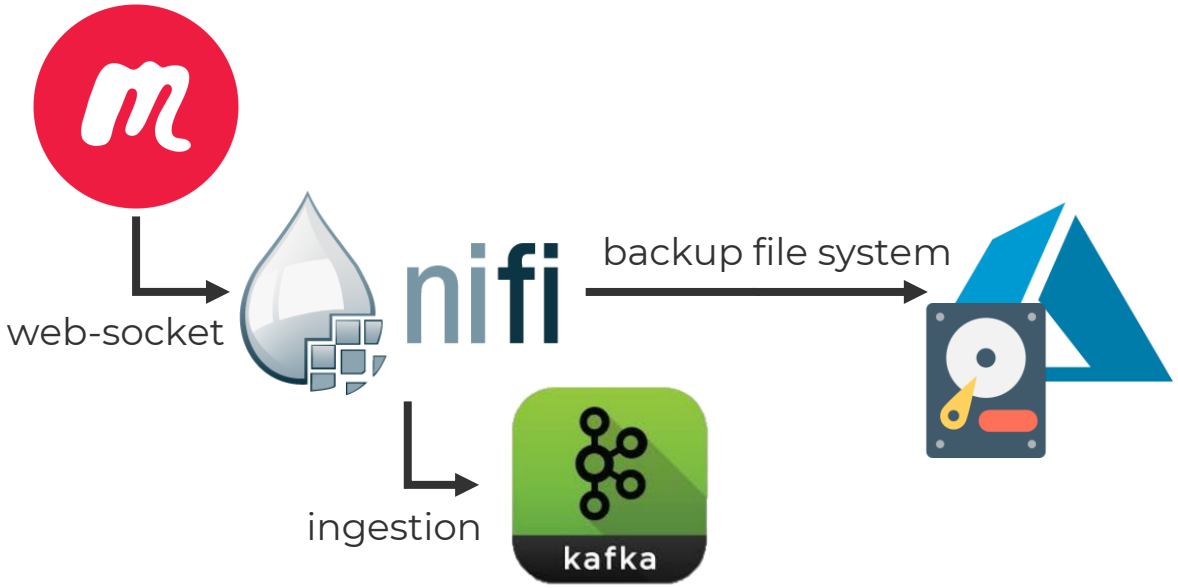
architecture

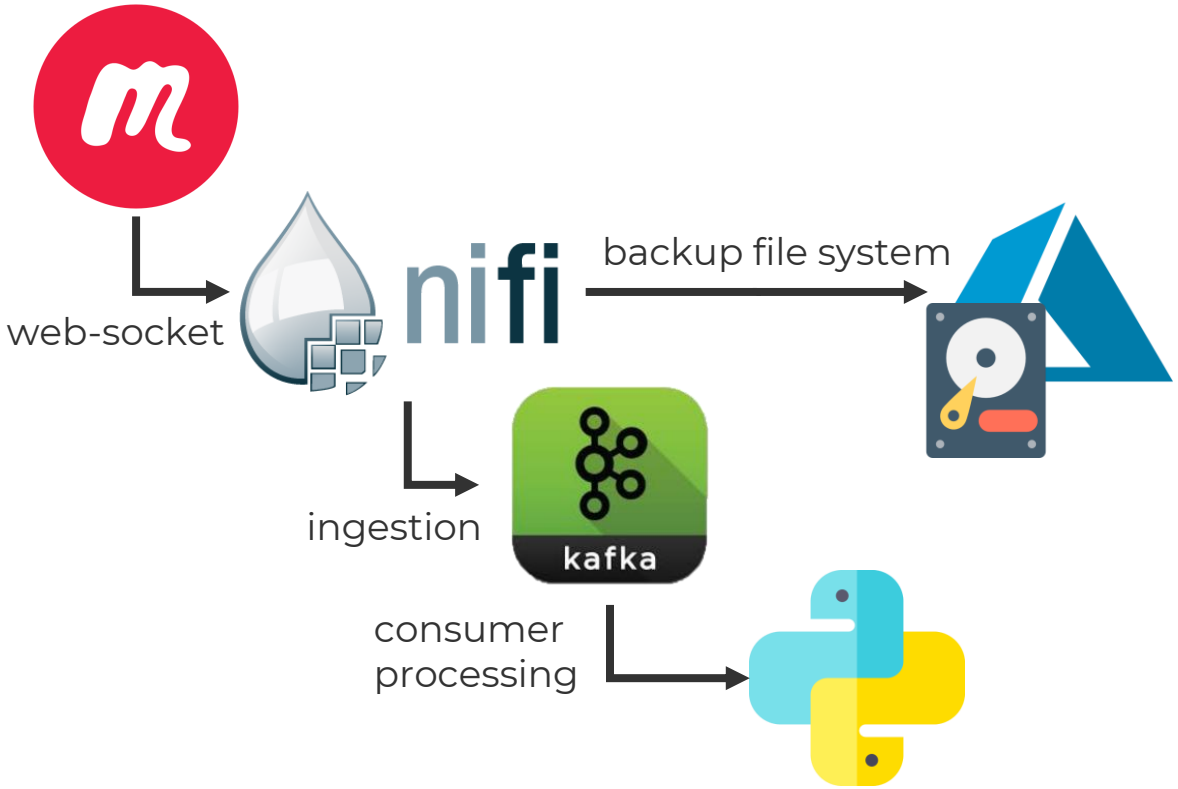

web-socket
nifi

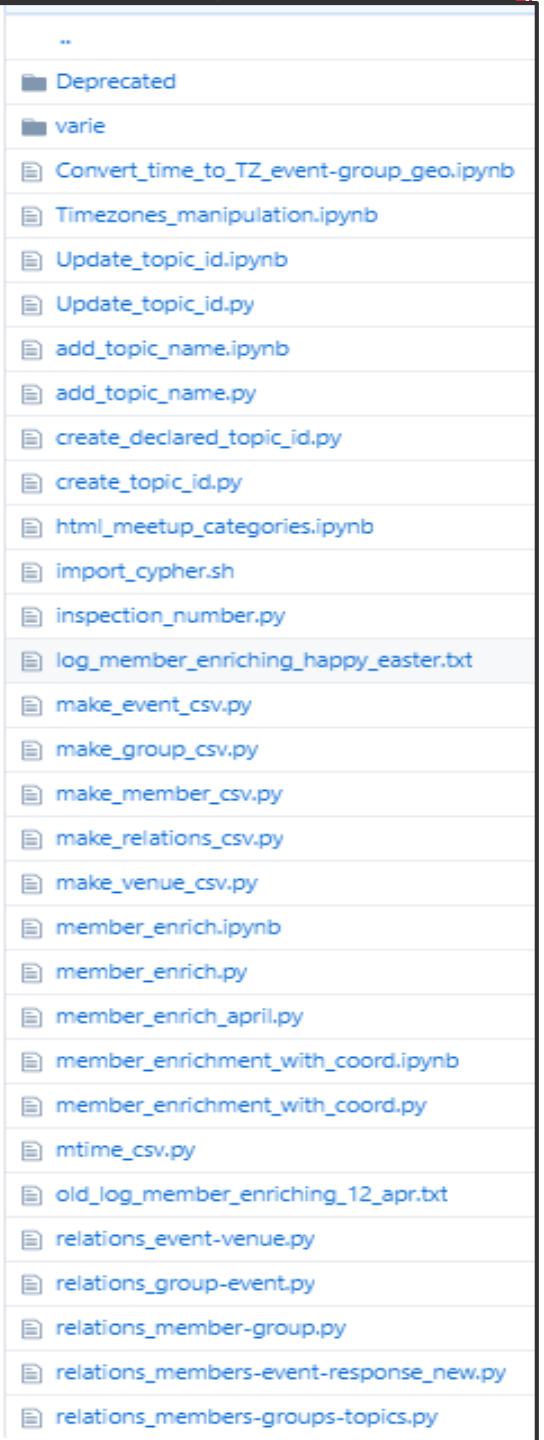

web-socket
ConnectWebSocket
ConnectWebSocket 1.5.0.3.1.2.0-7
org.apache.nifi - nifi-websocket-processors-nar
In 0 (0 bytes) 5 min
Read/Write 0 bytes / 453.47 KB 5 min
Out 318 (453.47 KB) 5 min
Tasks/Time 260 / 00:00:00.721 5 min
Name text message
Queued 0 (0 bytes)
UpdateAttribute
UpdateAttribute 1.5.0.3.1.2.0-7
org.apache.nifi - nifi-update-attribute-nar
In 318 (453.47 KB) 5 min
Read/Write 0 bytes / 0 bytes 5 min
Out 636 (906.94 KB) 5 min
Tasks/Time 318 / 00:00:00.204 5 min
Name success
Queued 0 (0 bytes)
Name success
Queued 0 (0 bytes)
PublishKafka_0_10
PublishKafka_0_10 1.5.0.3.1.2.0-7
org.apache.nifi - nifi-kafka-0-10-nar
In 318 (453.47 KB) 5 min
Read/Write 453.47 KB / 0 bytes 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 313 / 00:00:01.515 5 min
PutFile
PutFile 1.5.0.3.1.2.0-7
org.apache.nifi - nifi-standard-nar
In 318 (453.47 KB) 5 min
Read/Write 453.47 KB / 453.47 KB 5 min
Out 0 (0 bytes) 5 min
Tasks/Time 318 / 00:00:00.288 5 min

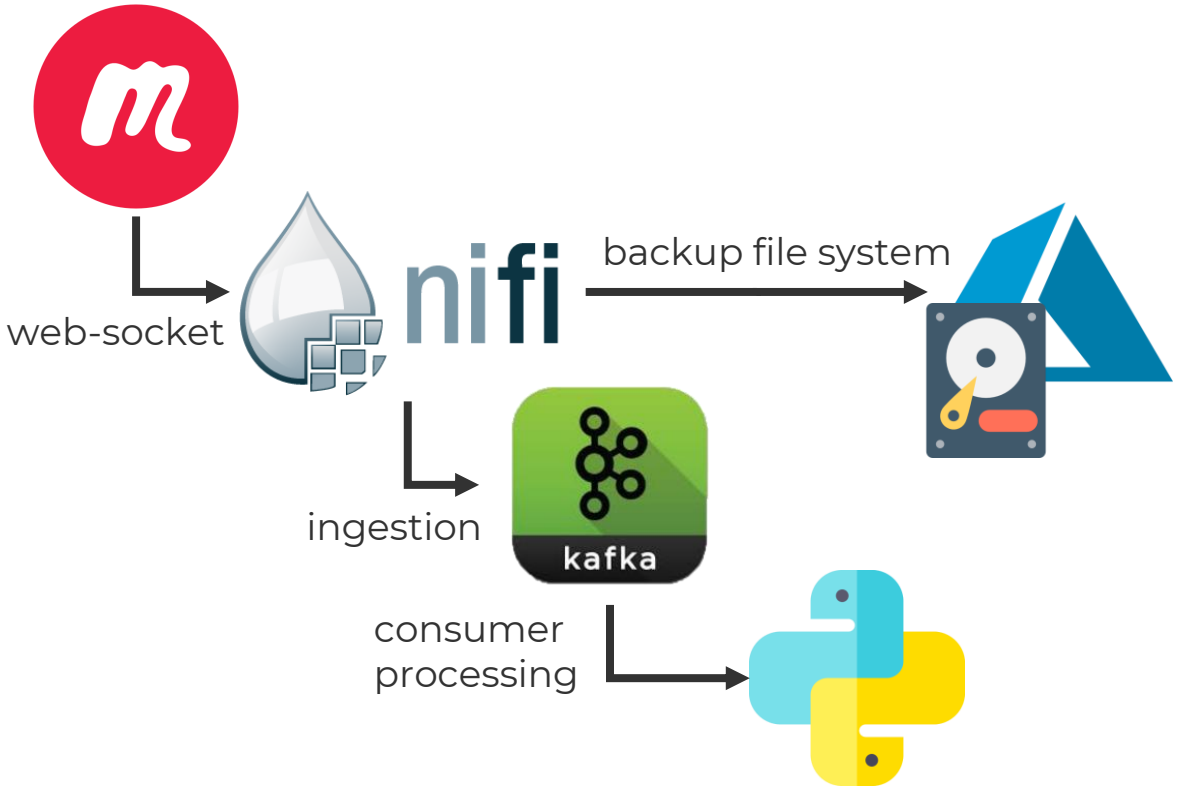

web-socket
nifi

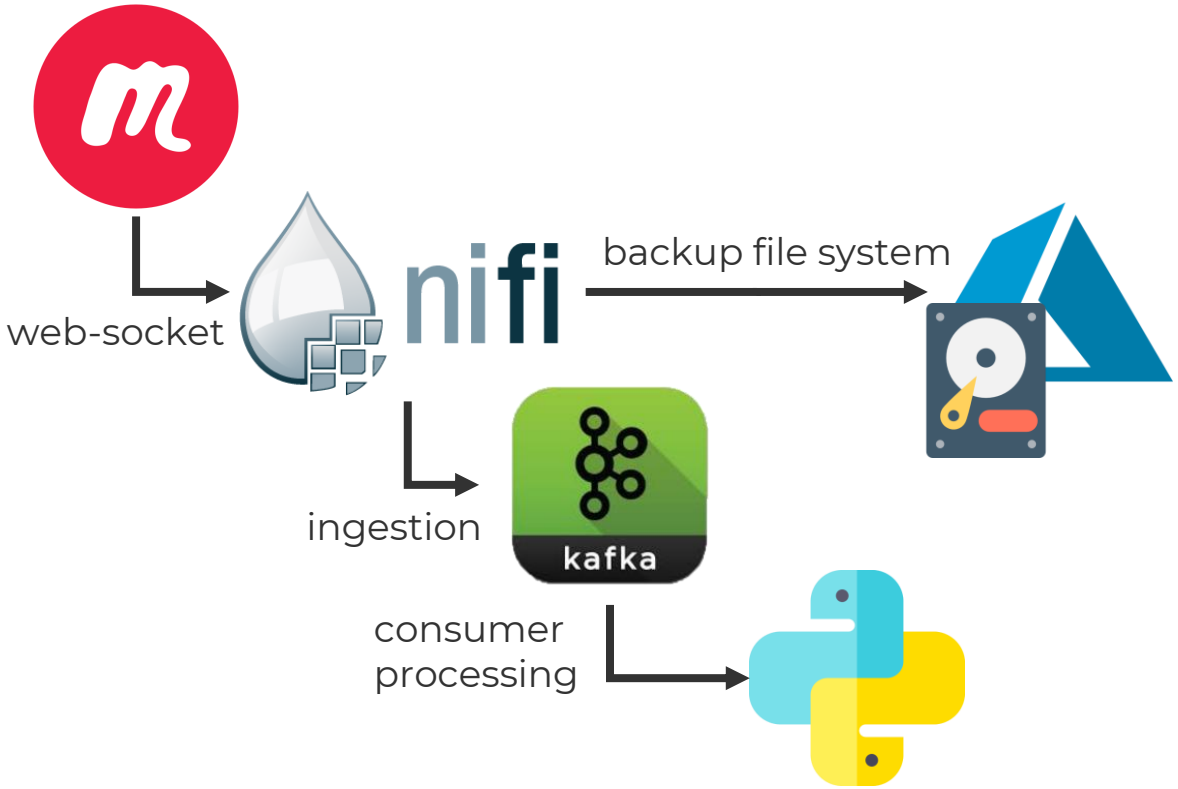

web-socket
nifi
backup file system

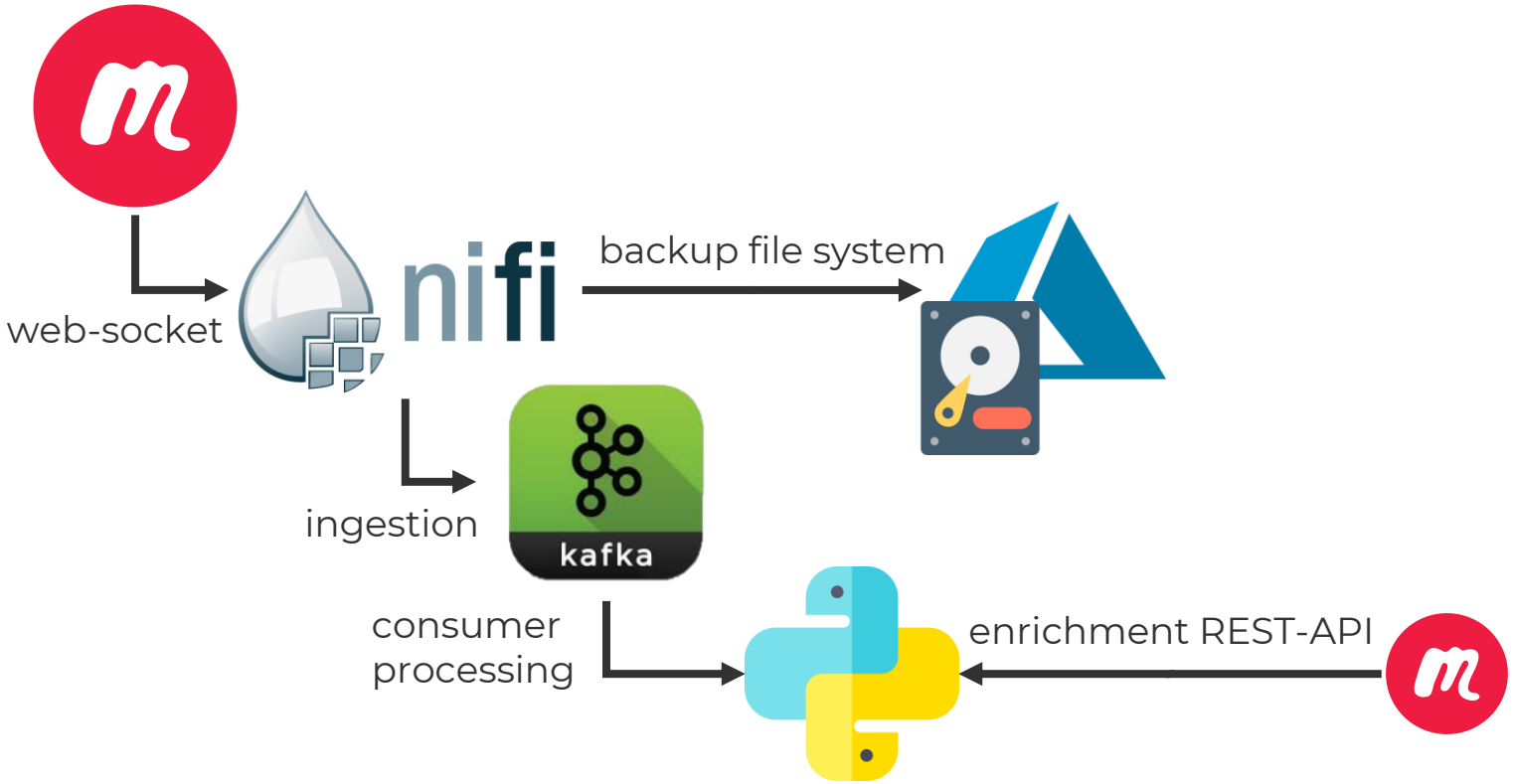

web-socket
nifi
backup file system
ingestion
kafka


web-socket
nifi
backup file system
ingestion
kafka
consumer
processing

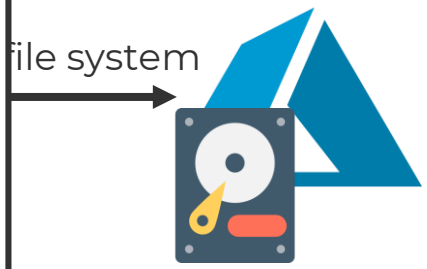

..
Deprecated
varie
Convert_time_to_TZ_event-group_geo.ipynb
Timezones_manipulation.ipynb
Update_topic_id.ipynb
Update_topic_id.py
add_topic_name.ipynb
add_topic_name.py
create_declared_topic_id.py
create_topic_id.py
html_meetup_categories.ipynb
import_cypher.sh
inspection_number.py
log_member_enriching_happy_easter.txt
make_event_csv.py
make_group_csv.py
make_member_csv.py
make_relations_csv.py
make_venue_csv.py
member_enrich.ipynb
member_enrich.py
member_enrich_april.py
member_enrichment_with_coord.ipynb
member_enrichment_with_coord.py
mtime_csv.py
old_log_member_enriching_12_apr.txt
relations_event-venue.py
relations_group-event.py
relations_member-group.py
relations_members-event-response_new.py
relations_members-groups-topics.py

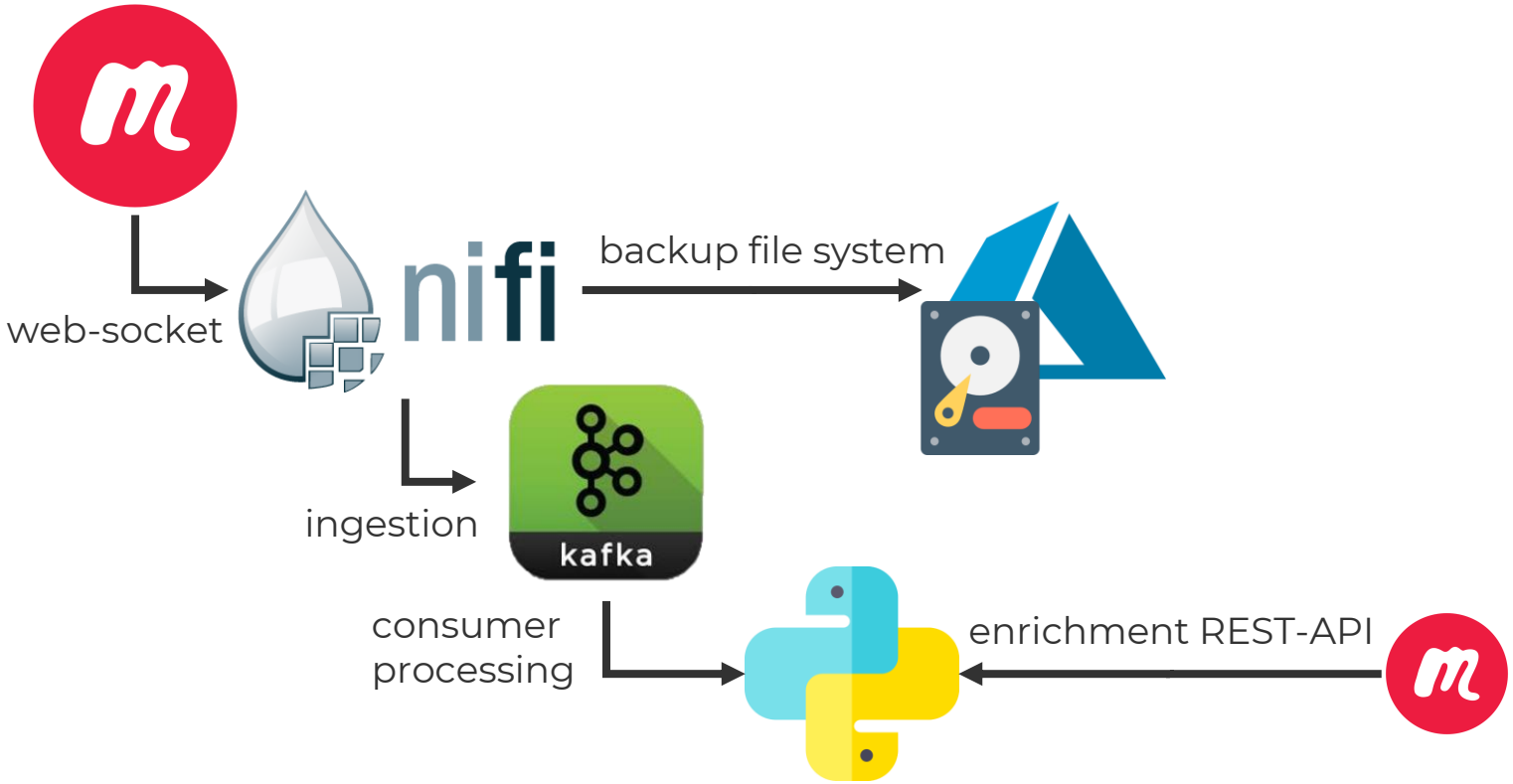

web-socket
nifi
backup file system
ingestion
kafka
consumer
processing

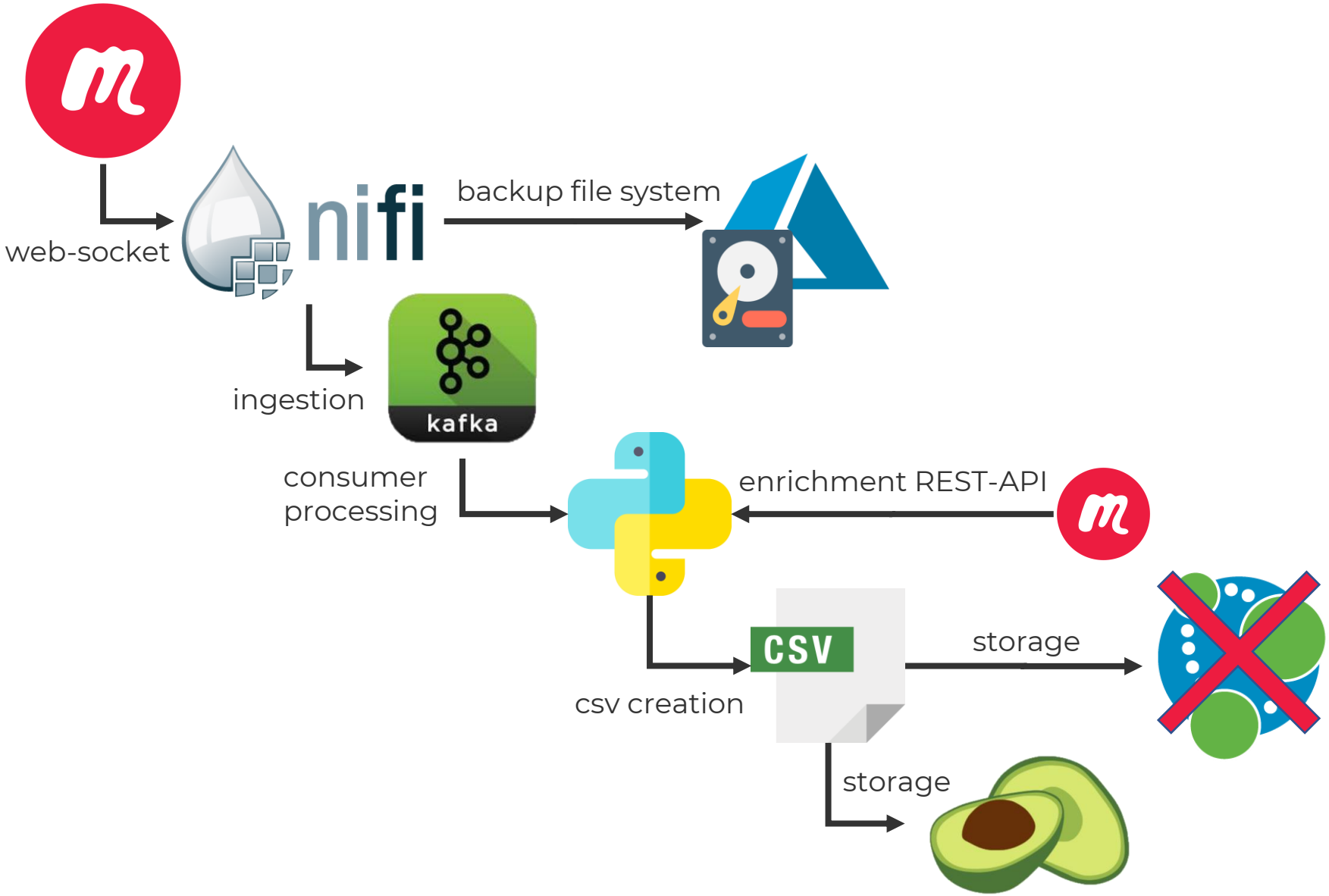

web-socket
nifi
backup file system
ingestion
kafka
consumer
processing

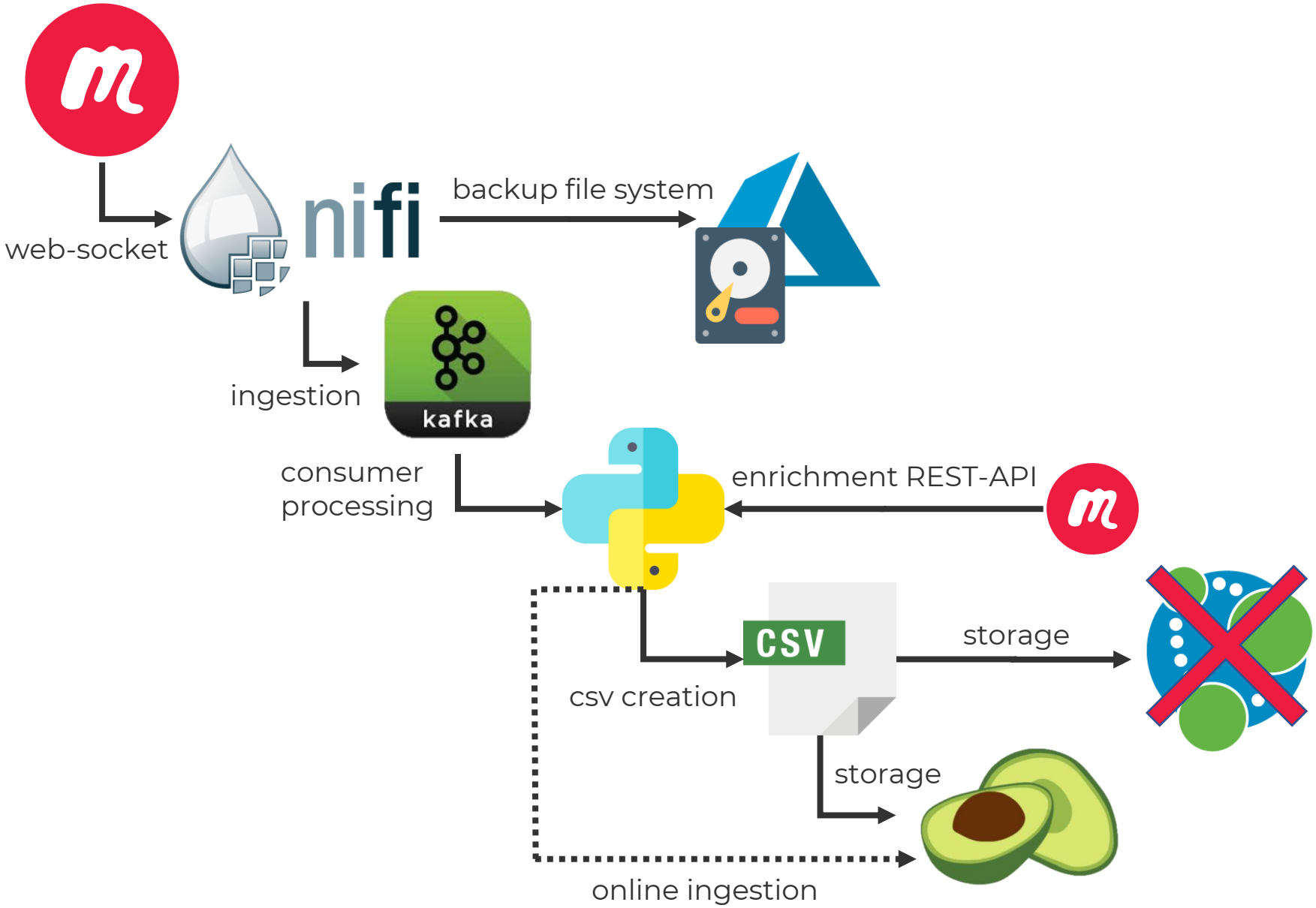

web-socket
nifi
backup file system
ingestion
kafka
consumer
processing
enrichment REST-API

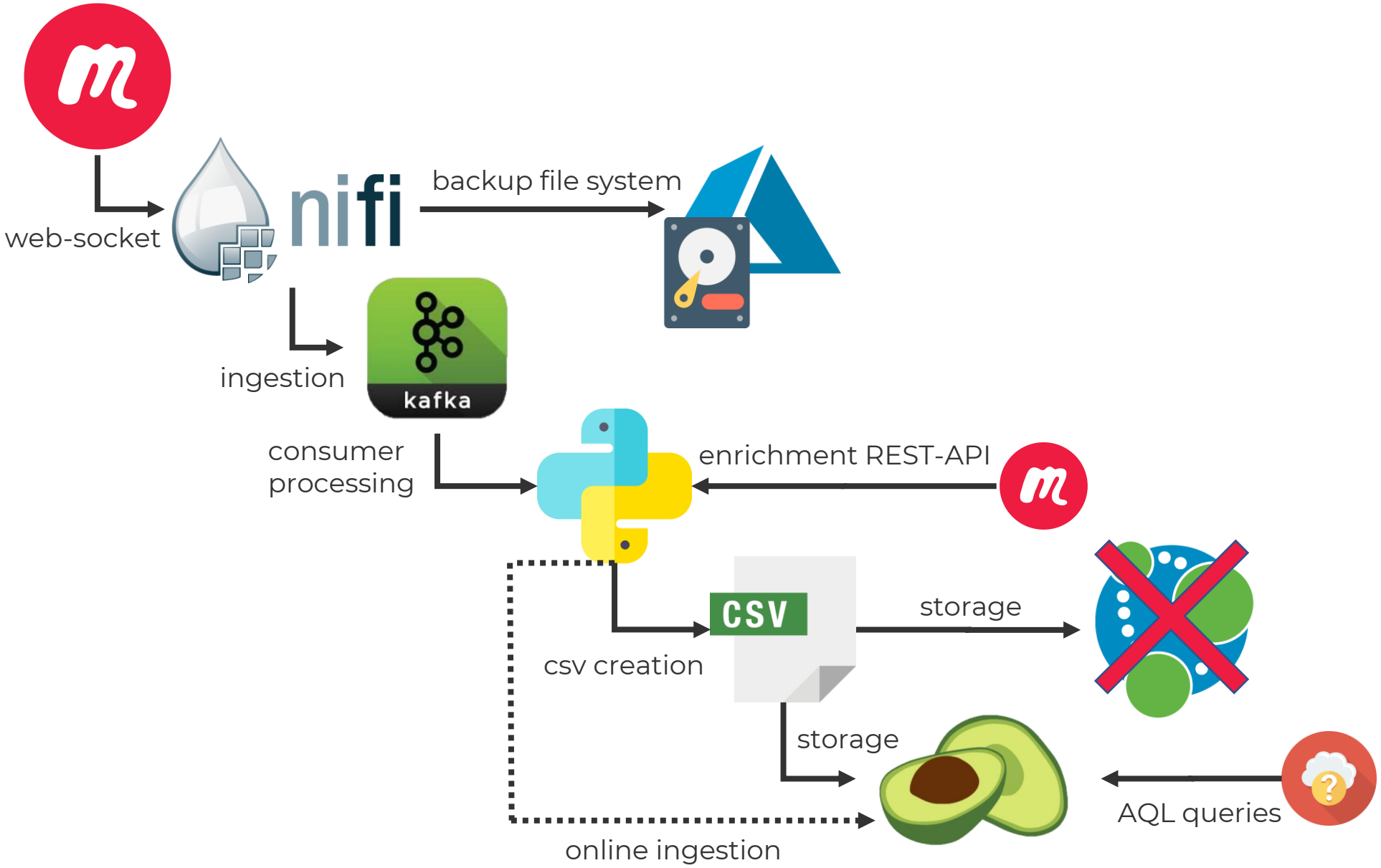
```python
def main():
        count=0
        tot_count=0
        print "requesting at http://api.meetup.com/2/members"

        #for line in member_df.itertuples():           #basic iter
        #for x, y in pairwise(member_df.itertuples()): #pairs iter
        for z in grouper(member_df.itertuples(),3): #groups

            #request parameters
            per_page = 1
            #results_we_got = per_page #more pages output
            offset = 0

            #get id (one should really vectorize it)
            id_0=member_df.iloc[z[0].Index]['member_id']
            id_1=member_df.iloc[z[1].Index]['member_id']
            id_2=member_df.iloc[z[2].Index]['member_id']

            # Meetup.com documentation here: http://www.meetup.com/meetup_api/docs/2/groups/
            try:
                response0=get_results({"member_id":id_0, "key":max_key, "page":per_page, "offset":offset}, tot_count,1)
                time.sleep(0.05) #PLS U NO BAN me

                response1=get_results({"member_id":id_1, "key":my_api_key, "page":per_page, "offset":offset}, tot_count,2)
                time.sleep(0.05) #PLS U NO BAN me

                response2=get_results({"member_id":id_2, "key":fabri_key, "page":per_page, "offset":offset}, tot_count,3)
            except Exception as e:
                print "exception encountered at requesting: "
                print e
            time.sleep(0.15) #PLS U NO BAN me
            offset += 1
            #results_we_got = response['meta']['count']
            #time.sleep(1)
            count+=1
            tot_count+=1
            if debug:
                print "resp 0"
                print response0
                print "resp 1"
                print response1
                print "resp 2"
                print response2
```




web-socket
nifi
backup file system
ingestion
kafka
consumer
processing
enrichment REST-API


web-socket
nifi
backup file system
ingestion
kafka
consumer
processing
enrichment REST-API
CSV
storage
csv creation
storage


web-socket
nifi
backup file system
ingestion
kafka
consumer
processing
enrichment REST-API
CSV
csv creation
storage
storage
online ingestion

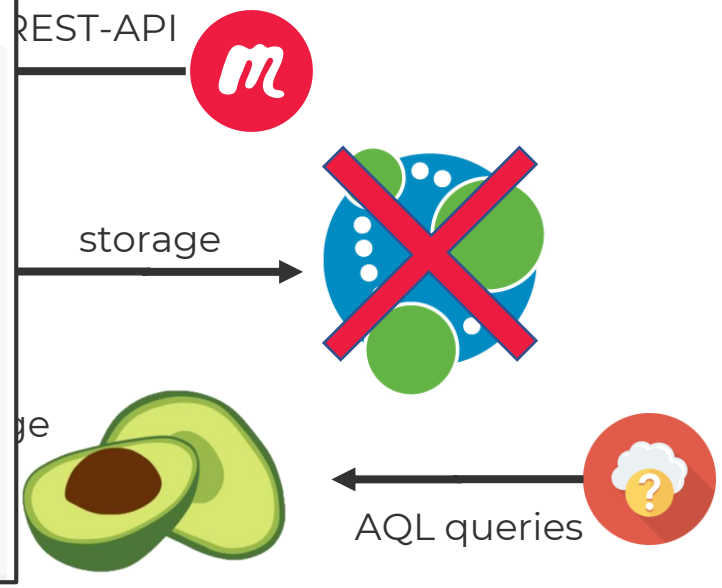

web-socket
nifi
backup file system
ingestion
kafka
consumer
processing
enrichment REST-API
CSV
csv creation
storage
storage
AQL queries
online ingestion

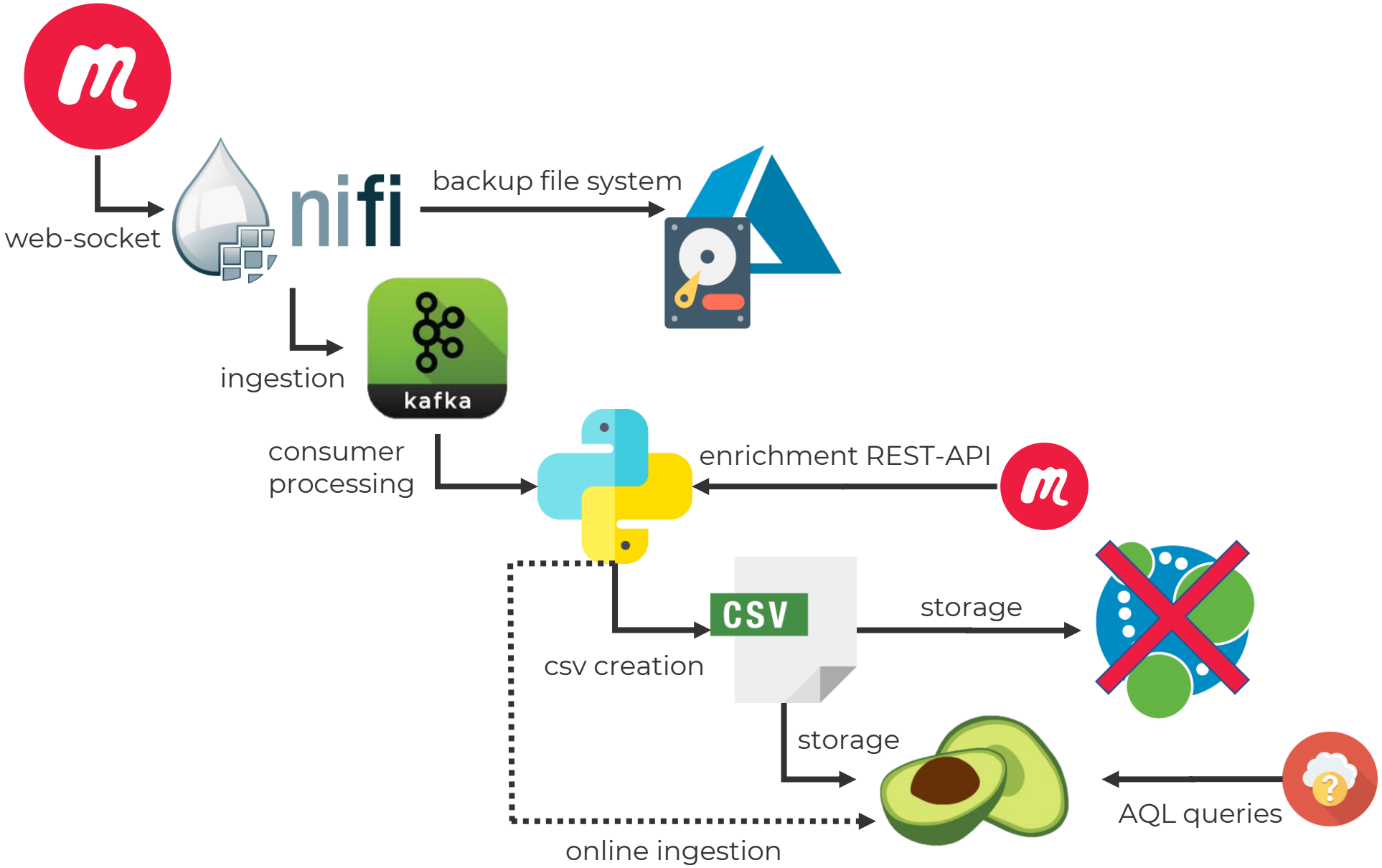
- Massimo numero di partecipanti ad un evento per Paese

```
let eventsWithPart = (
    for m in member
        for e in 1..1 outbound m will_partecipates
            collect event_name = e.event_name, event = e._id with count into participants
            sort participants desc
            return {"participants": participants, "event_id": event, "name": event_name}
)
for event in eventsWithPart
    for g in 1..1 inbound event.event_id hosted_event
    collect country = g.group_country
    aggregate max_part = max(event.participants)
    sort max_part desc
    return {country, max_part}
```

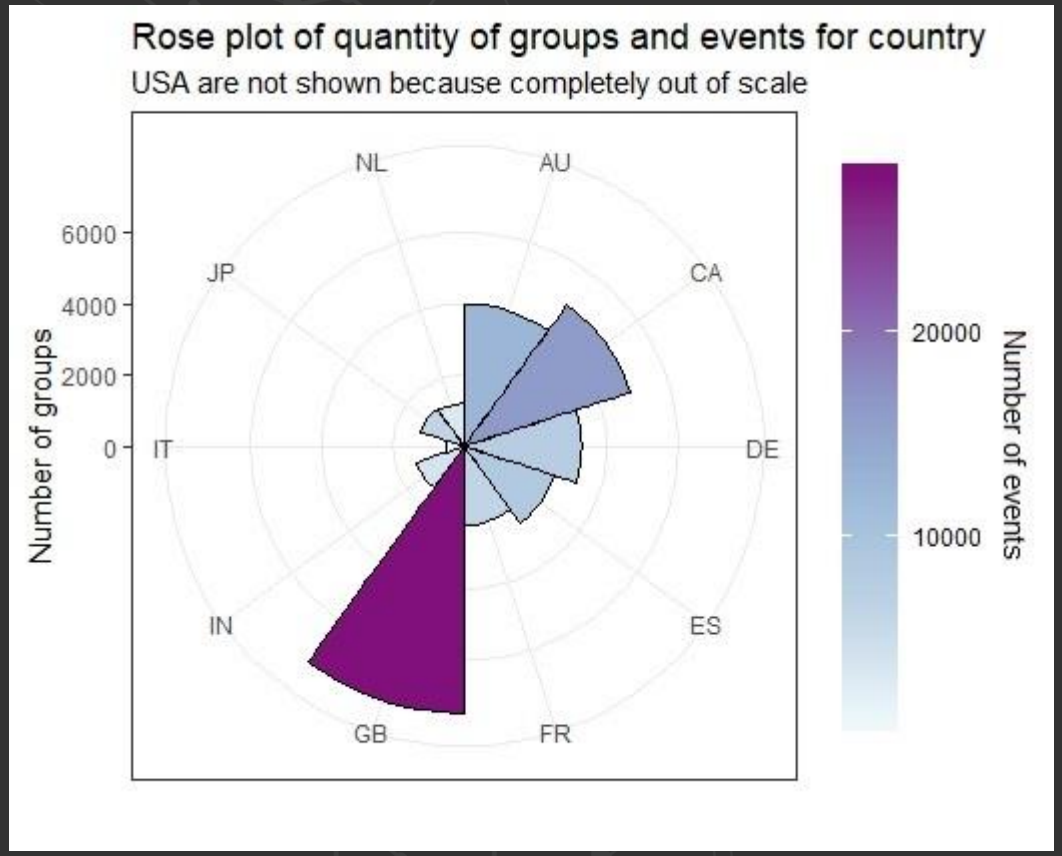
- Jaccard similarity measure between declared and group topics for every member

```
match (m:Member)-[r:IS_INTERESTED_IN]->(t:Topic)<-[r1:DECLARED_INTEREST_IN]-(m)
with count(t) as corrispondenze, m
match (m)-[r:DECLARED_INTEREST_IN]->(t:Topic)
with count(r) as dichiarati, corrispondenze, m
match (m)-[r1:IS_INTERESTED_IN]-(t:Topic)
with m, corrispondenze, dichiarati, count(r1) as interessato
return m.id as ID,
corrispondenze,
dichiarati,
interessato,
(toFloat(corrispondenze))/((toFloat(dichiarati)+toFloat(interessato))-toFloat(corrispondenze)) as jac_similarity
order by jac_similarity desc
```




web-socket
nifi
backup file system
ingestion
kafka
consumer
processing
enrichment REST-API
CSV
csv creation
storage
storage
AQL queries
online ingestion

results



quantitative measures

quantity of events by Country

quantity of groups by Country

maximum number of participants by Country

average number of guests for each participant

trend topic among users

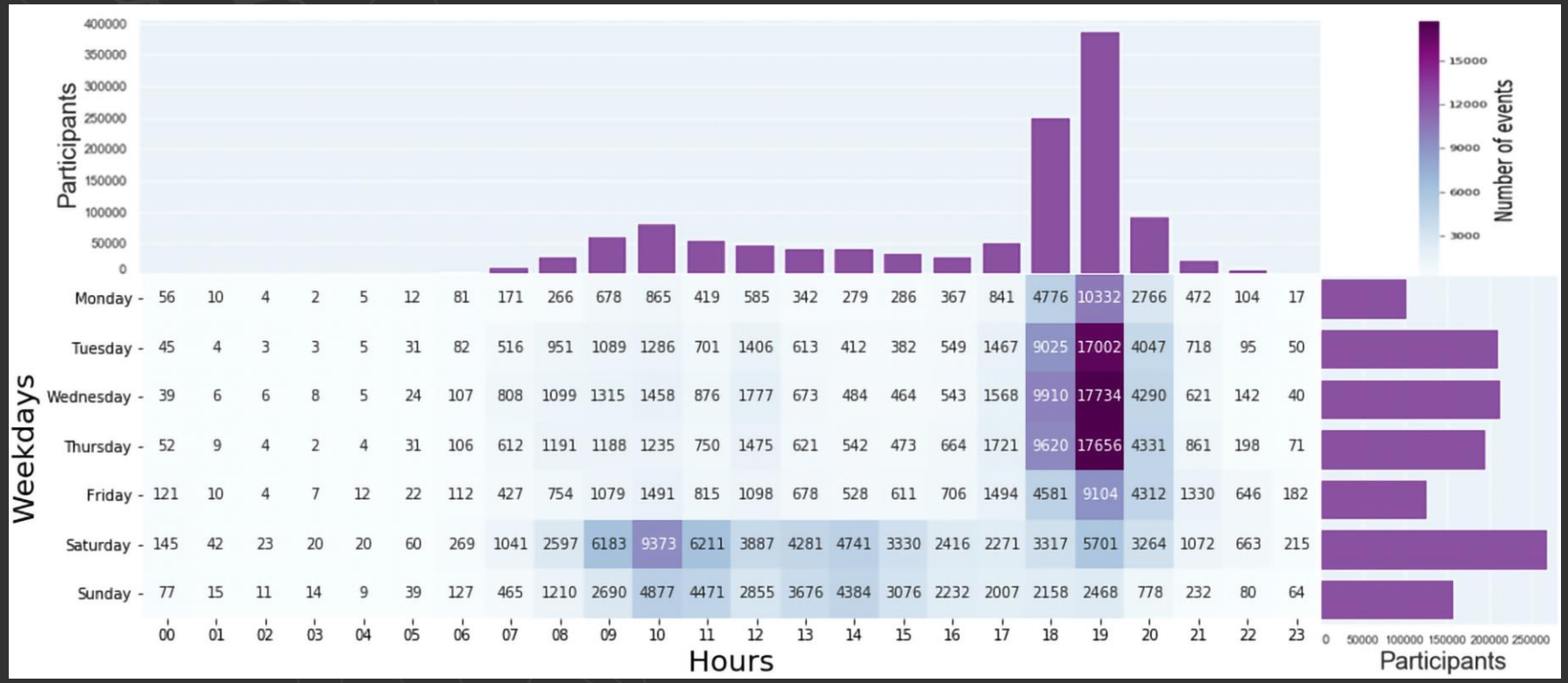
trend topic for groups

temporal distribution of the events

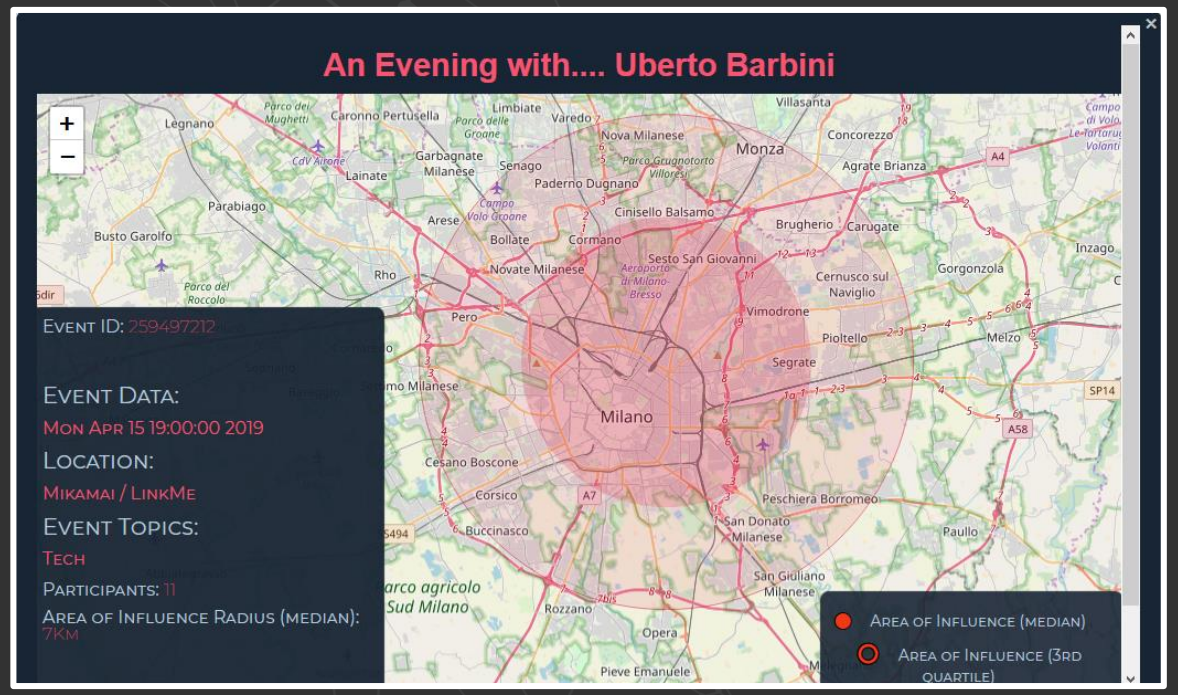
best moment for a meetup

- in a day
- during the week

worldwide vs locally



area of influence (event)



an interactive map that

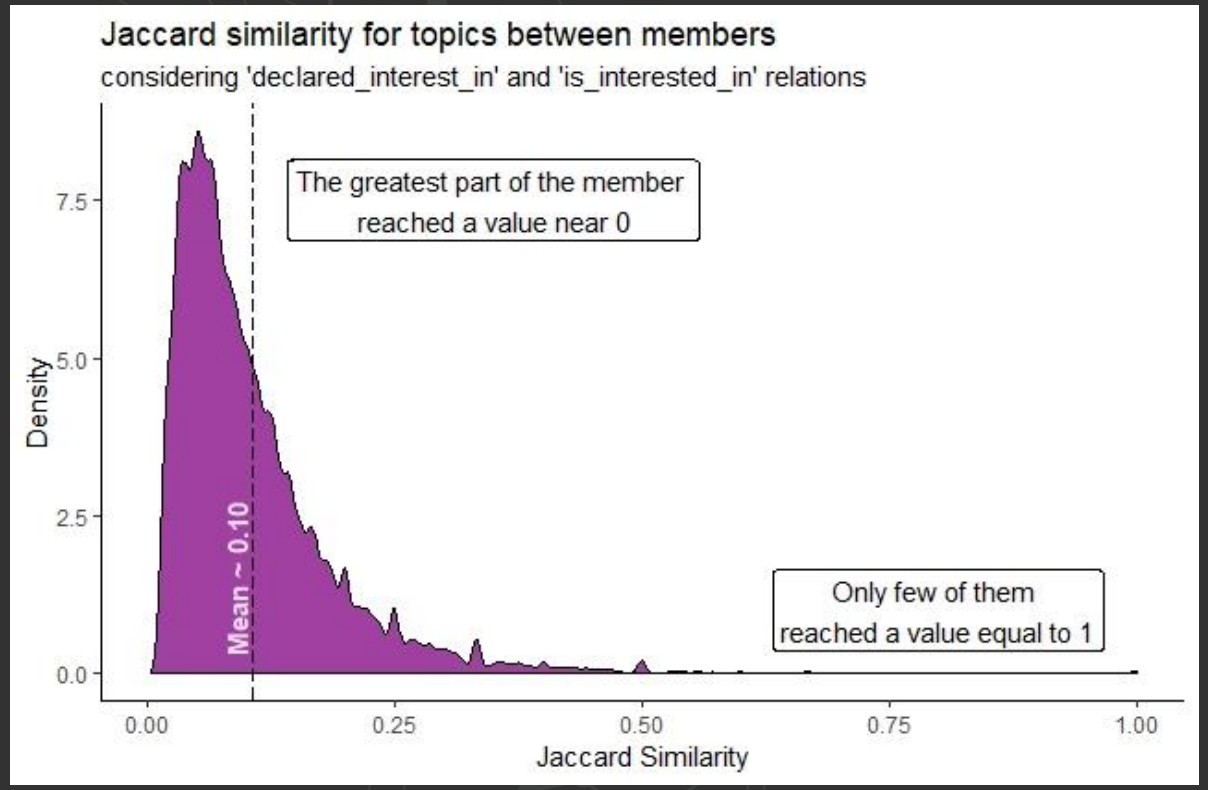
displays useful informations

about events, including a

measure of the area of

influence radius for each event

recommender system analysis



evaluate the recommender system

efficiency investigating similarity between

the group topics and those topics the user

is interested in

it is more like an idea!

challenges


1 symbolic links
2 import Cypher vs Py2Neo
3 temporal metadata conversion
4 optimization in message extraction
5 import Cypher vs Arangoimp
6 scalability
7 live streaming ingestion

1 symbolic links

2

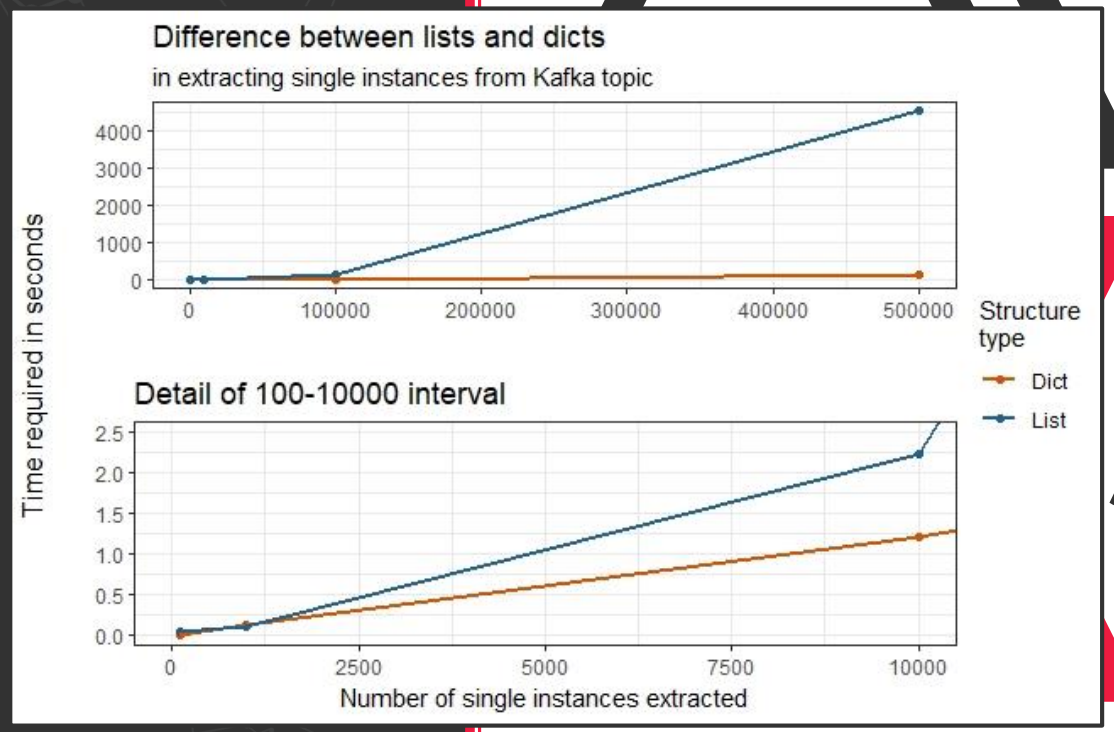
3 temporal metadata conversion

4 optimization in message extraction

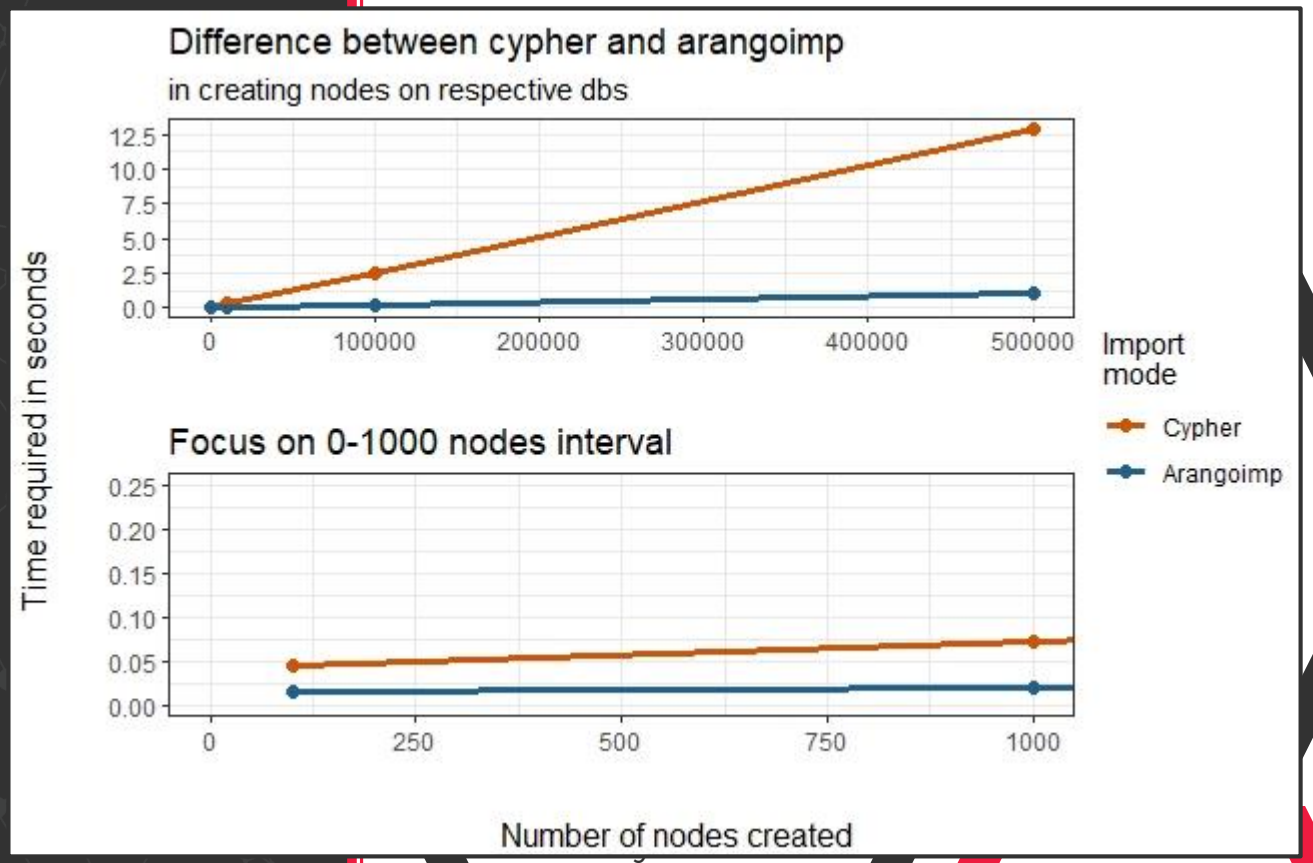
5 import Cypher vs Arangoimp

6 scalability

7 live streaming ingestion




1 symbolic links
2 import Cypher vs Py2Neo
3 temporal metadata conversion
4 optimization in message extraction
5 import Cypher vs Arangoimp
6 scalability
7 live streaming ingestion



in

5 import Cypher vs Arangoimp

6 scalability

3 temporal metadata conversion

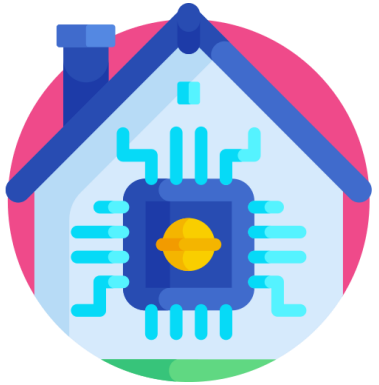
7 live streaming ingestion


1 symbolic links
2 import Cypher vs Py2Neo
3 temporal metadata conversion
4 optimization in message extraction
5 import Cypher vs Arangoimp
6 scalability
7 live streaming ingestion

conclusions

KEY POINTS

answered research questions successfully



software structure benefits from these features

improvement of personal skills

scalability

fault tolerance

consistency

THANK YOU