

Idea: Application which can collect statistics from multiple sources and post it to destination applying aggregations or historical comparisons

Functional:

Collect statistic
unlimited
implemented
sources

Compare
statistics with
historical
data

Send
result to
Telegram

Non-
functional:

Online
data
collection

Idea: Application which can collect statistics from multiple sources and post it to destination applying aggregations or historical comparisons

Functional:

Collect statistic unlimited implemented sources

It can be many sources of the same type

Collect statistic from YouTube

Collect statistic from telegram

Compare statistics with historical data

Data should be stored

Comparisons

Aggregations

Trends

Send result to Telegram

When data should be posted?

Universal messaging mechanism to replace concrete system

Where? In channel or in message? And how, from user, from bot?

How, how the message should look, change of one value or some group of metrics

Data is collected by some trigger

Internal or external trigger?

Webhook or cronjob like

Non-functional:

Online data collection

No more than 1min timelag

Implement on 1 server

Downtime is acceptable if it is less than 1min

Java/Kotlin + Linux

Rollout pipeline

Questions to ask:

Ask about each point directly, what it should include

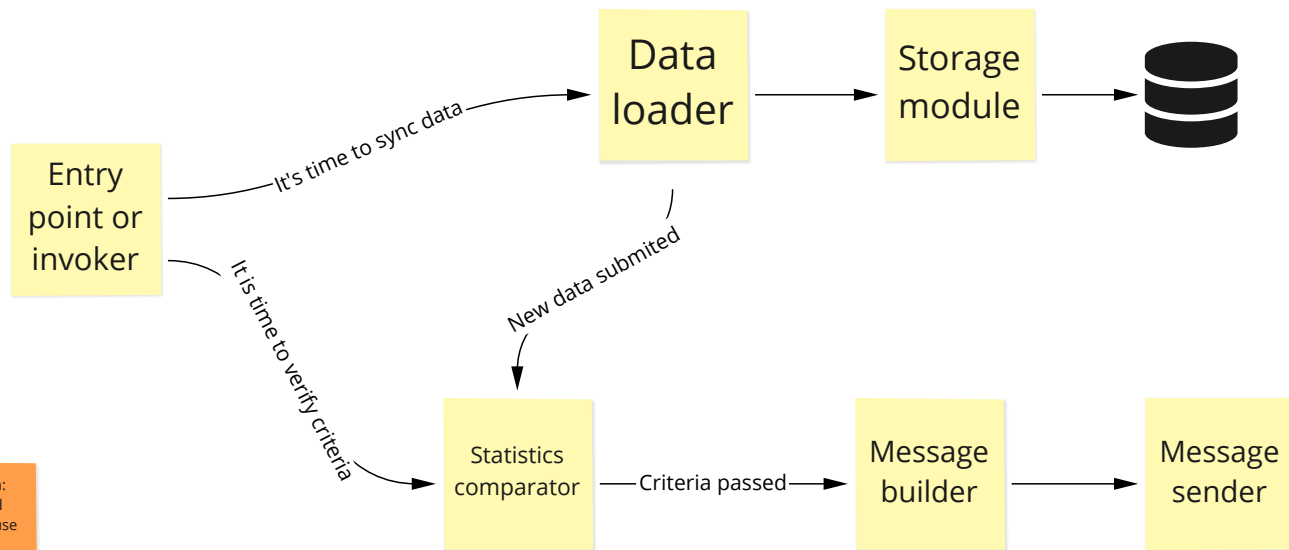
What about hardware limitations?

software limitations?

Possible load?

What aspects of CAP should be taken into account? https://en.wikipedia.org/wiki/CAP_theorem

Modules definition

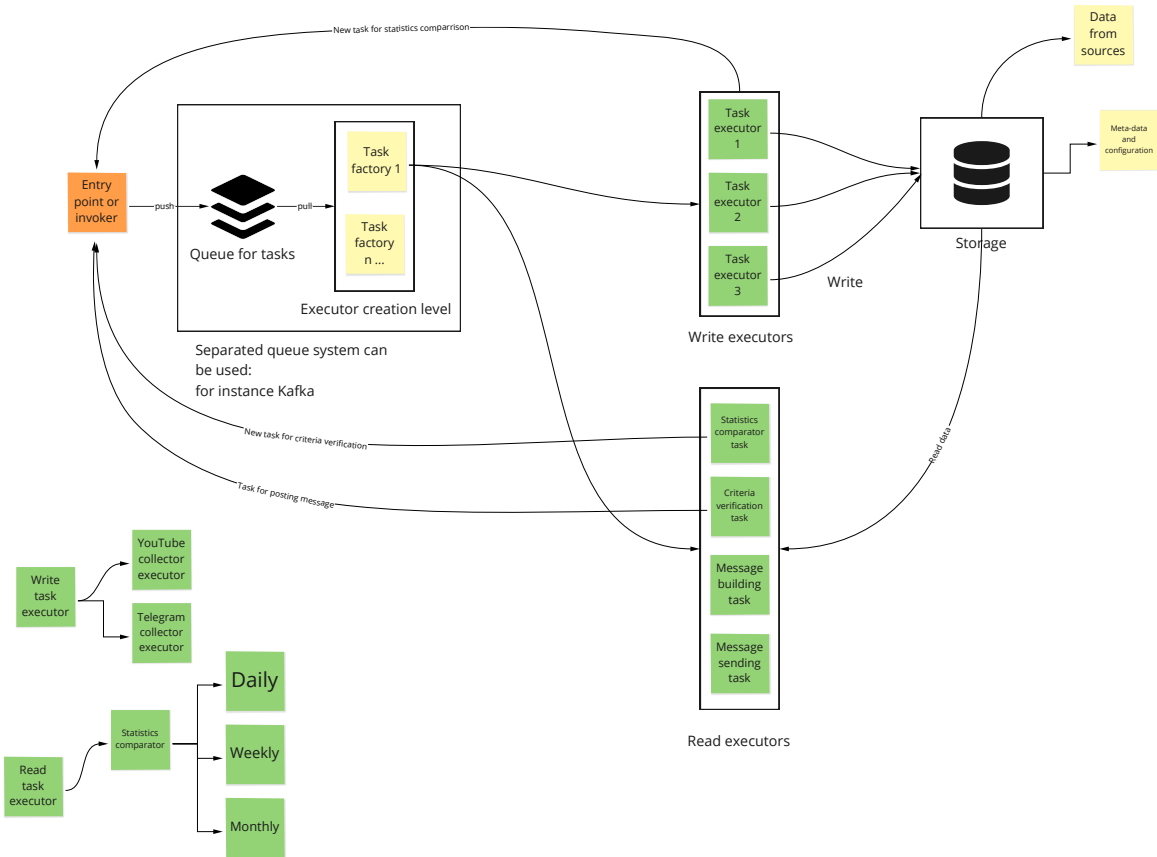


Current idea:
Event based
system, because
the idea of
application is
based on event

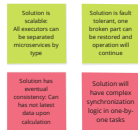
As a plus it has
possible
decoupling for
microservices in
case of high load

For con:
eventual
consistency
and not latest
data

Solution definition

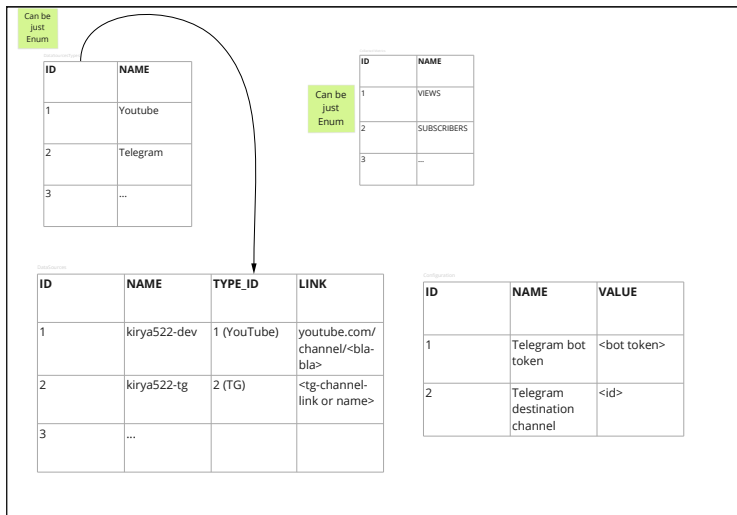
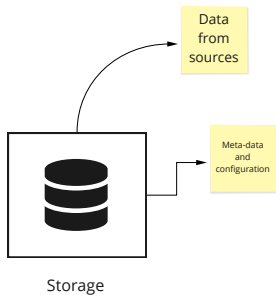


pros/cons



Observations:





Relational storage

Series values

ID	DATE	DATA_SOURCE_ID	METRIC_ID	VALUE
1	21:36 14/08/2022	1 (kirya522-dev)	1 (views)	1338
2	21:35 14/08/2022	2 (kirya522-tg)	2 (subscribers)	320
3	...			

Time-series storage