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

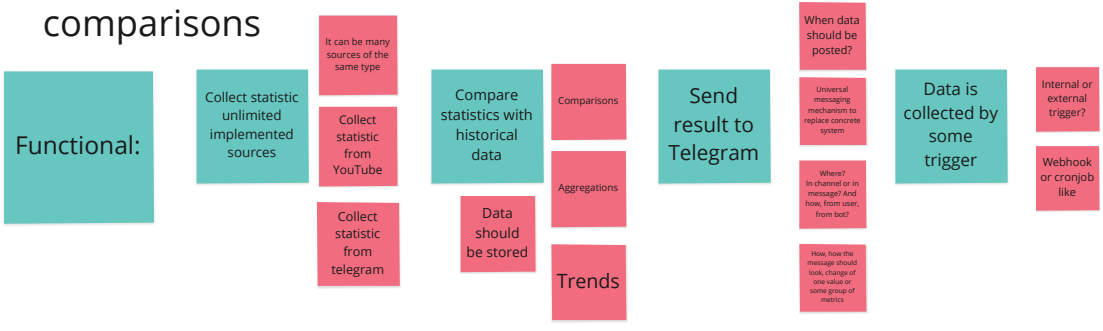
Functional:

- Collect statistic unlimited implemented sources
- Compare statistics with historical data
- Send result to Telegram

Non-functional:

- Online data collection

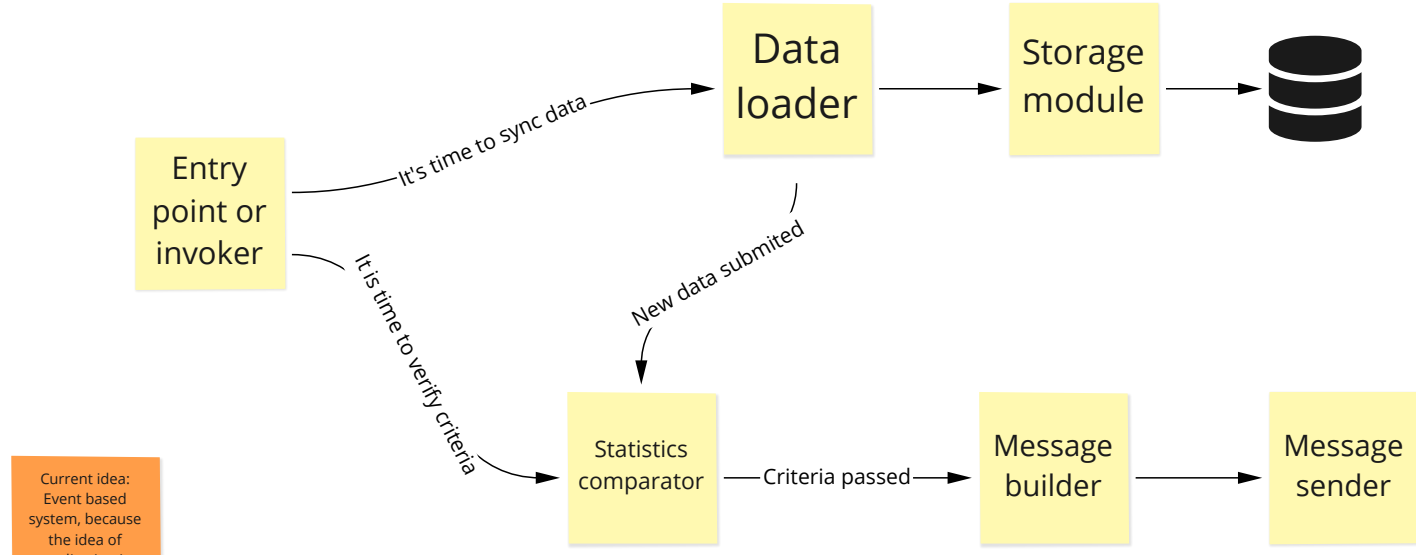
Idea: Application which can collect statistics from multiple sources and post it to destination applying aggregations or historical comparisons



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

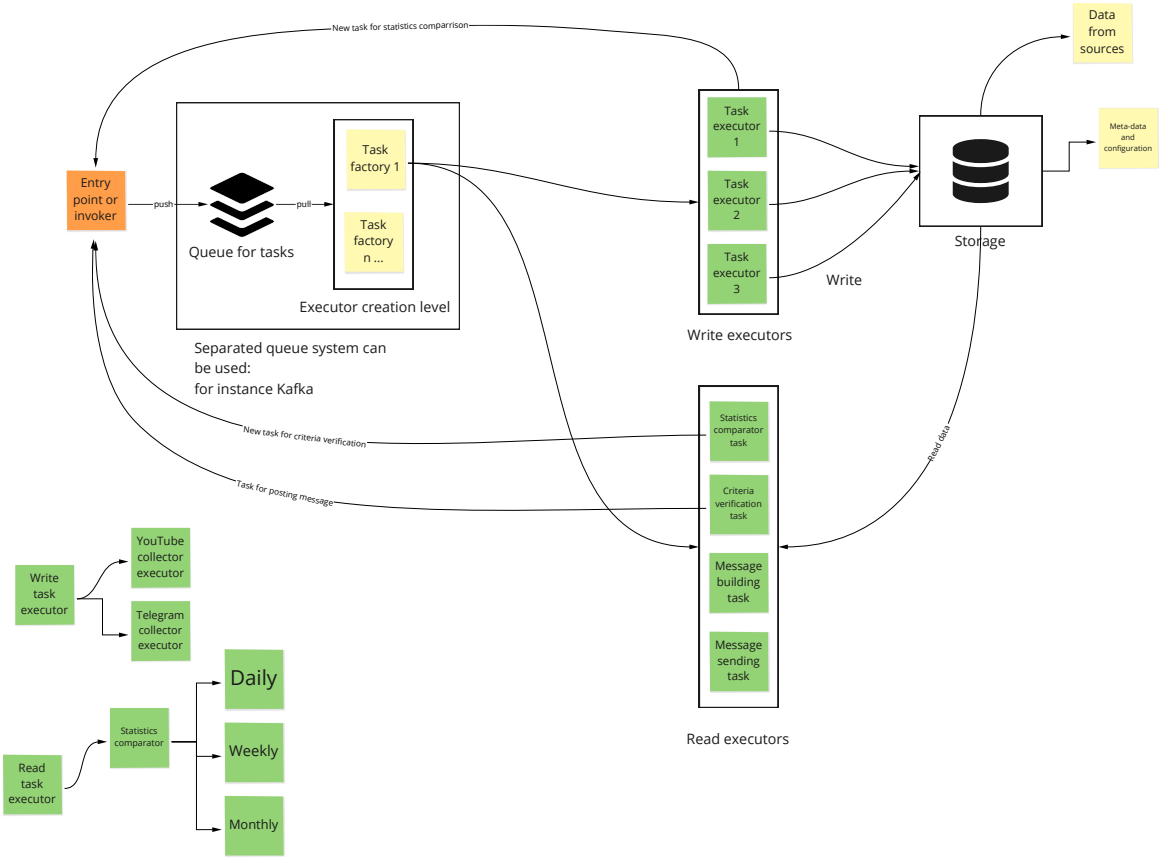


Current idea: Event based system, because the idea of application is based on event

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

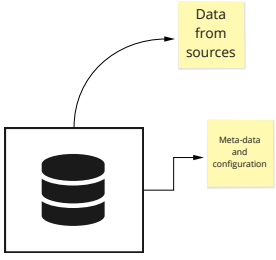
For con: eventual consistency and not latest data

Solution definition



pros/cons

Observations:



Storage

Can be just Enum

ID	NAME
1	Youtube
2	Telegram
3	...

Can be just Enum

ID	NAME
1	VIEWS
2	SUBSCRIBERS
3	...

ID	NAME	TYPE_ID	LINK
1	kirya522-dev	1 (YouTube)	youtube.com/channel/<bla-bla>
2	kirya522-tg	2 (TG)	<tg-channel-link or name>
3	...		

ID	NAME	VALUE
1	Telegram bot token	<bot token>
2	Telegram destination channel	<id>

Relational storage

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