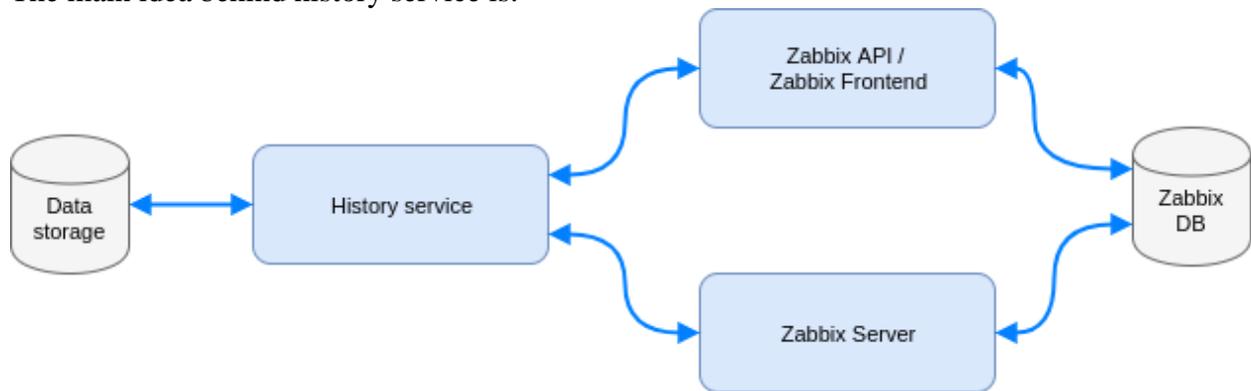


Here is a quick recap on our history service meeting.

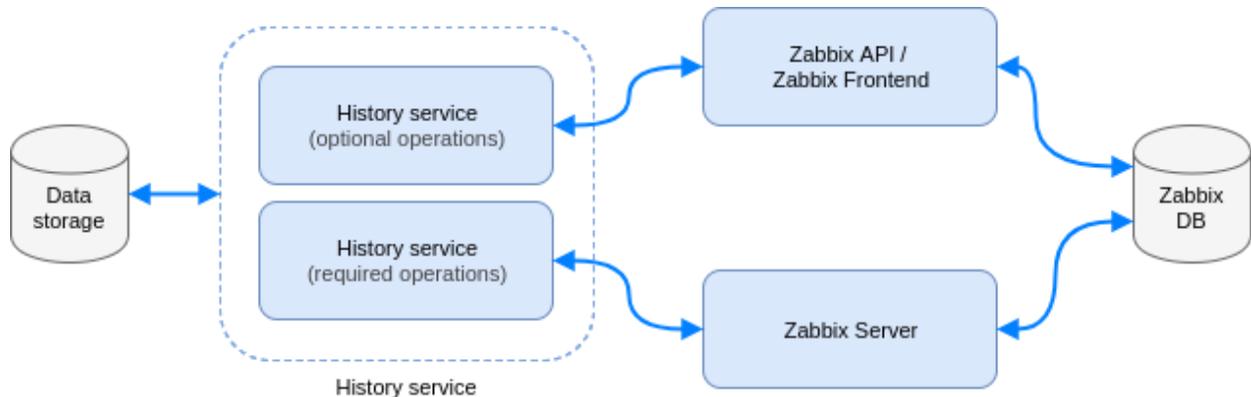
Questions discussed:

1. What actions (operations) are required from history service?
2. Should history service perform data aggregations and what are aggregation options required?
3. Should history service support actions required by specification of Zabbix API, but not used in frontend (like history.get search calls)?
4. Possible solutions to create flexible, reliable and scalable solution for history service.
5. Overall performance.

The main idea behind history service is:



To simplify creation and support of history **services decision was made to split history service into required** (required for Zabbix Server to operate) **and optional part** (required by Zabbix Frontend and Zabbix API):

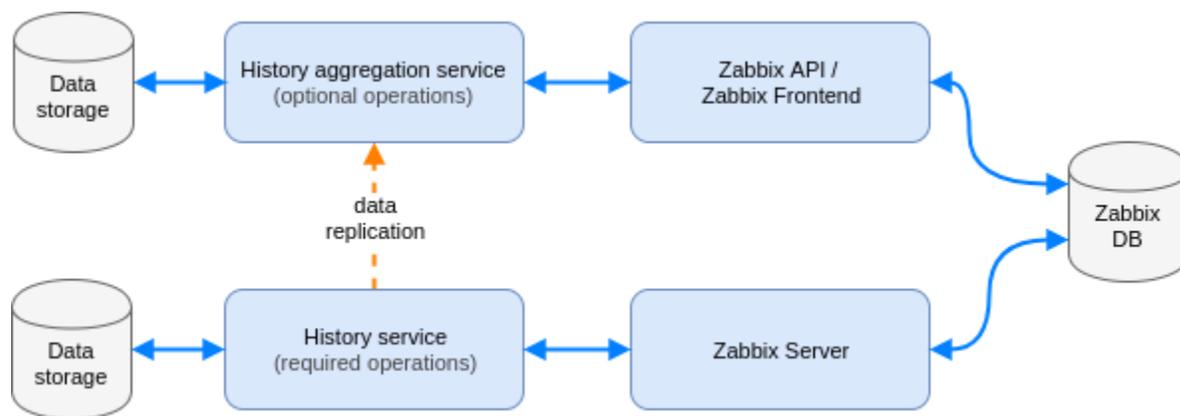


Third parties can implement required operations for Zabbix Server to work and implement optional operations needed for visualization (Zabbix Frontend). If, for example, visualization layer (Zabbix Frontend) will require additional data aggregation from history service at some point (for example, new type of graph is introduced), data aggregation action is placed into

optional part of history service operations so existing history service will work with new versions of Zabbix Frontend, just not showing the new type of graph. This separation should be described in spec and documentation allowing quick implementation of history service for cases when visualization is not that important.

As a reaction to adding of optional operations, Zabbix Frontend should handle "not supported" responses and display appropriate message to user.

To minimize load on data storage and history service there is additional option to split things needed for Zabbix to operate from things needed to visualize data:



This solution relies on data replication that can be configured to minimize load on data storage. This solution introduces additional component "History aggregation service" that allows us to separate "мух от котлет" and to keep history service clean and simple without a need to implement aggregation logic needed for visualization as a part of History service solution.

Decision was made to keep this solution as a possible improvement and to not to focus on it right now because of the additional complexity added by data replication required between history service and history aggregation service.

Decisions made (numbers match with discussed questions):

1. History service will implement all actions required by Zabbix Server (as a required part) and Zabbix Frontend / API (as an optional part).
2. History service will perform data aggregations needed by Zabbix Frontend / API (as an optional part), additional actions are described in TODO.
3. It is unclear if history service should implement complex search logic required by Zabbix API specification, but not needed in Zabbix Frontend. Question is moved to open questions.
4. Possible solutions and descriptions (including simplified diagrams) are provided within this letter and should be added to the specification. It was decided to split history service actions into required and optional without implementing additional aggregation service.
5. It was decided to perform performance (load) testing to identify the impact of requests of visualization layer (Zabbix Frontend) to overall history service performance. This is a

part of activity required to be sure that improvement with data replication should remain out of scope for this development.