

# Timeline Service v.2: next milestones

**Bold** items are design required.

## [Alpha 2]

1. Li/Varun - UI integration (pending YARN-3368)
2. More types of applications: learning
  - o MapReduce flows
  - o Li/Varun - Tez migration
  - o Vrushali - Hive, cascading: plumbing/connection to tez/MR
3. **Joep/Li - Fault tolerance - storage level**
4. **Sangjin/Vrushali - (Security) authentication (kerberos)**
5. Joep/Vrushali - Deployment ideas for a single-node HBase setup

## [Beta]

6. JMX: reader, collectors (logs and metrics), RM/NM JMX
7. Flow activity handling for long running applications (not necessarily services)
8. **Migration**
  - o **Can we import v.1.x data?**
  - o **How can frameworks cut over to v.2?**
9. Separate process / **Containers for collectors** - stability, ability to roll new code with minimal disruption
  - o Impact on timeline client (limits on latency and memory usage)
10. Timeline Service working through RM failover (HA) (basic acceptability)
11. Command line support
12. **Off-application clients/collectors: different data schemas, security (use cases)**
  - o **Hive use case**
13. Test driver setup/improvements

## [Post-beta]

14. (Security) authorization - timeline domains like in v.1
15. **Offline aggregation**
16. Timeline service correctness and completeness through RM failover (HA)
17. Fault tolerance - collector-level fault tolerance
  - o At minimum, the timeline client should not crash/keep crashing due to issues with the timeline collector. The application should be able to run irrespective of issues with collector.
  - o If it crashes, we need a redo log for recovery the collector-state
18. **Working with YARN federation**
19. **Better support for services** as opposed to short-running applications
  - o Timeline information from first class services
  - o NodeManagers sending its own stats directly
  - o RM writing cluster level metrics [YARN-3881](#)

**20. Multi-DC setup**

21. Not ready for alternate storage yet, but will need to do work to make that happen

22. HDFS stats

23. Pooling readers

24. YARN first-class flow API with lifecycles