Tags in Hadoop

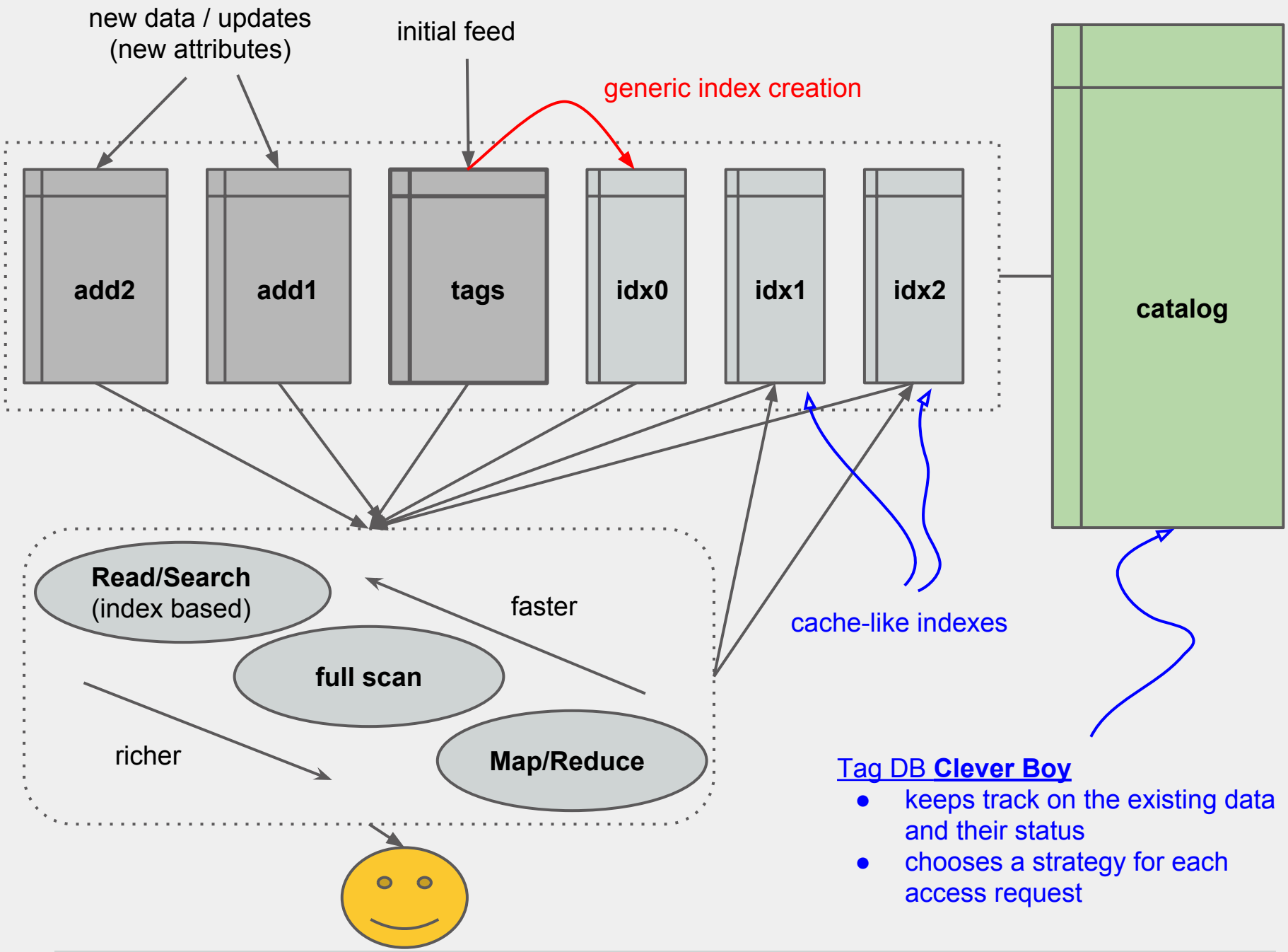
How to store and read data

based on (almost one year of) playing with Hadoop

Hadoop is very flexible. It allows many different ways of storing and access data. Data access goes from direct read (already quite fast), through index-based searches (instantaneous) to full Map/Reduce tasks.

Capabilities quickly evolve.

To fully profit from its functionality, our system should be able to handle a variety of access patterns. Also it should be opened to evolution.


new data / updates
(new attributes)
initial feed
generic index creation
add2
add1
tags
idx0
idx1
idx2
catalog
Read/Search
(index based)
faster
full scan
richer
Map/Reduce
cache-like indexes
Tag DB Clever Boy
keeps track on the existing data and their status
chooses a strategy for each access request

Creation

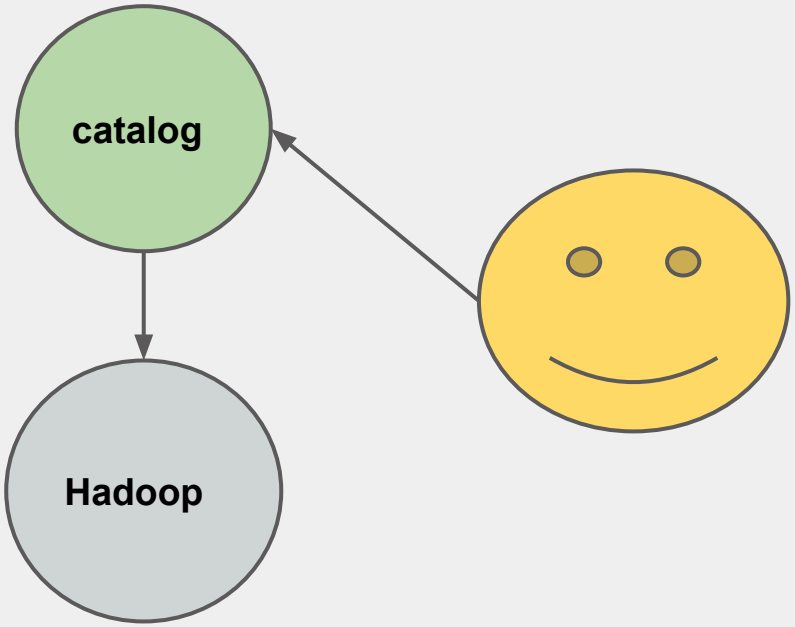
1. initial tags are fed into Hadoop (files)
2. generic index in created (i.e. creation of *map files*)
3. if needed, other data is added (following existing generic index - if possible)
4. new (cache-like) indexes are created all during the data lifetime as needed

catalog is updated after each step

Use

- search/read - if access following generic (or other) index
- full scan (i.e full search/read) or Map/Reduce if no suitable index
  - a new index created, catalog updated

Catalog is the key element. It keeps track of the tag/data status and all indexes. Based on HBase (or some SQL db).



It seems to me, that we should now stop playing around and **start building a real** (even if only a prototype-level) **system.**

<u>lessons from the existing system</u>

- communication between services should be clearly and strictly defined
- service authentication/authorisation (= certificate management) should be well handled