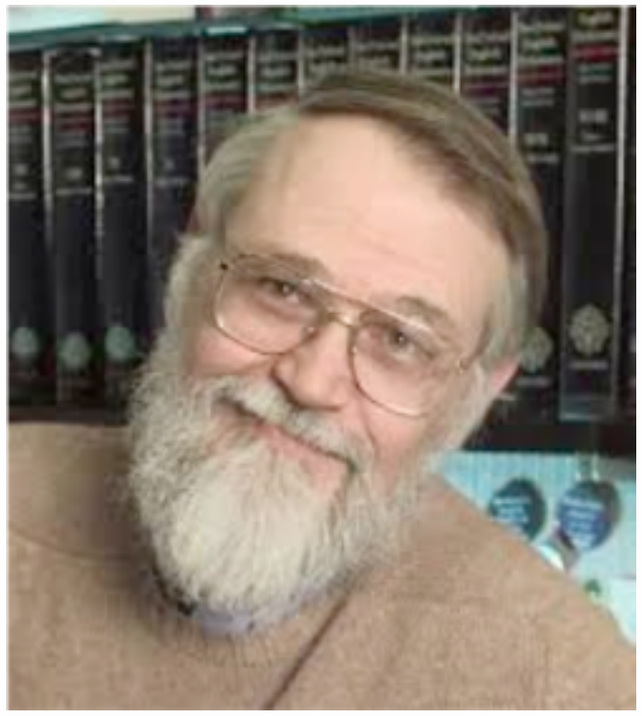
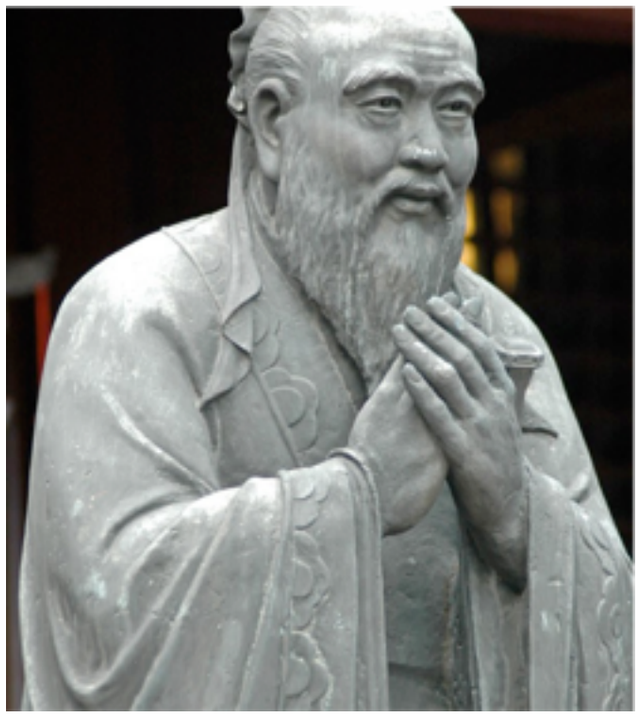
Inside Hulu's Data Platform

Prasan Samtani
prasan.samtani@hulu.com

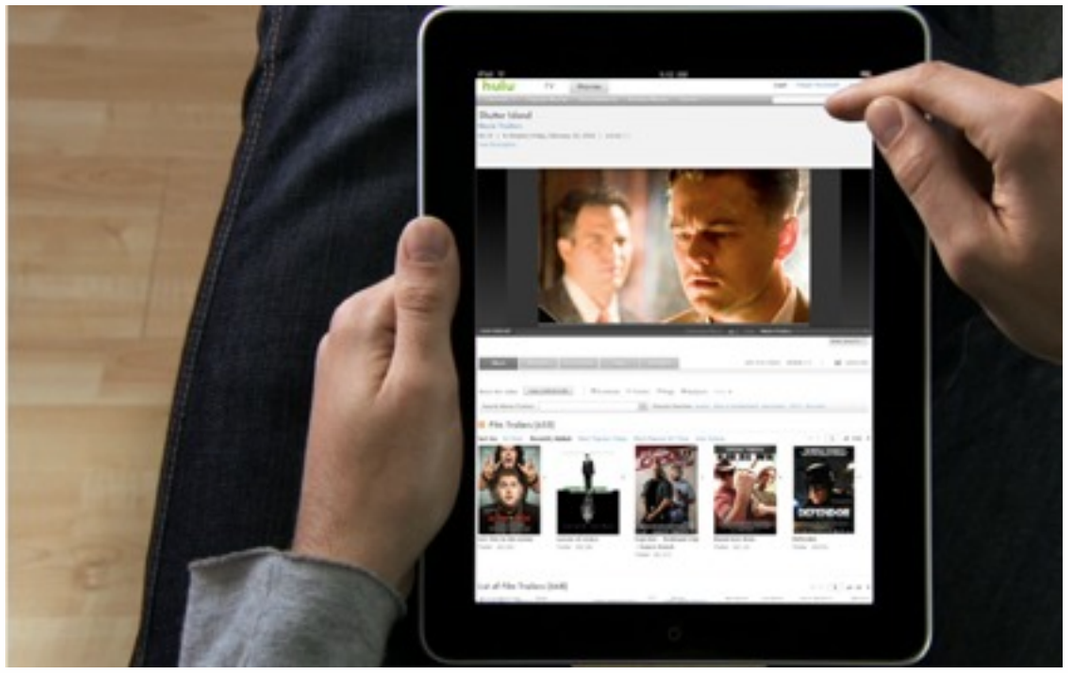
Disclaimer

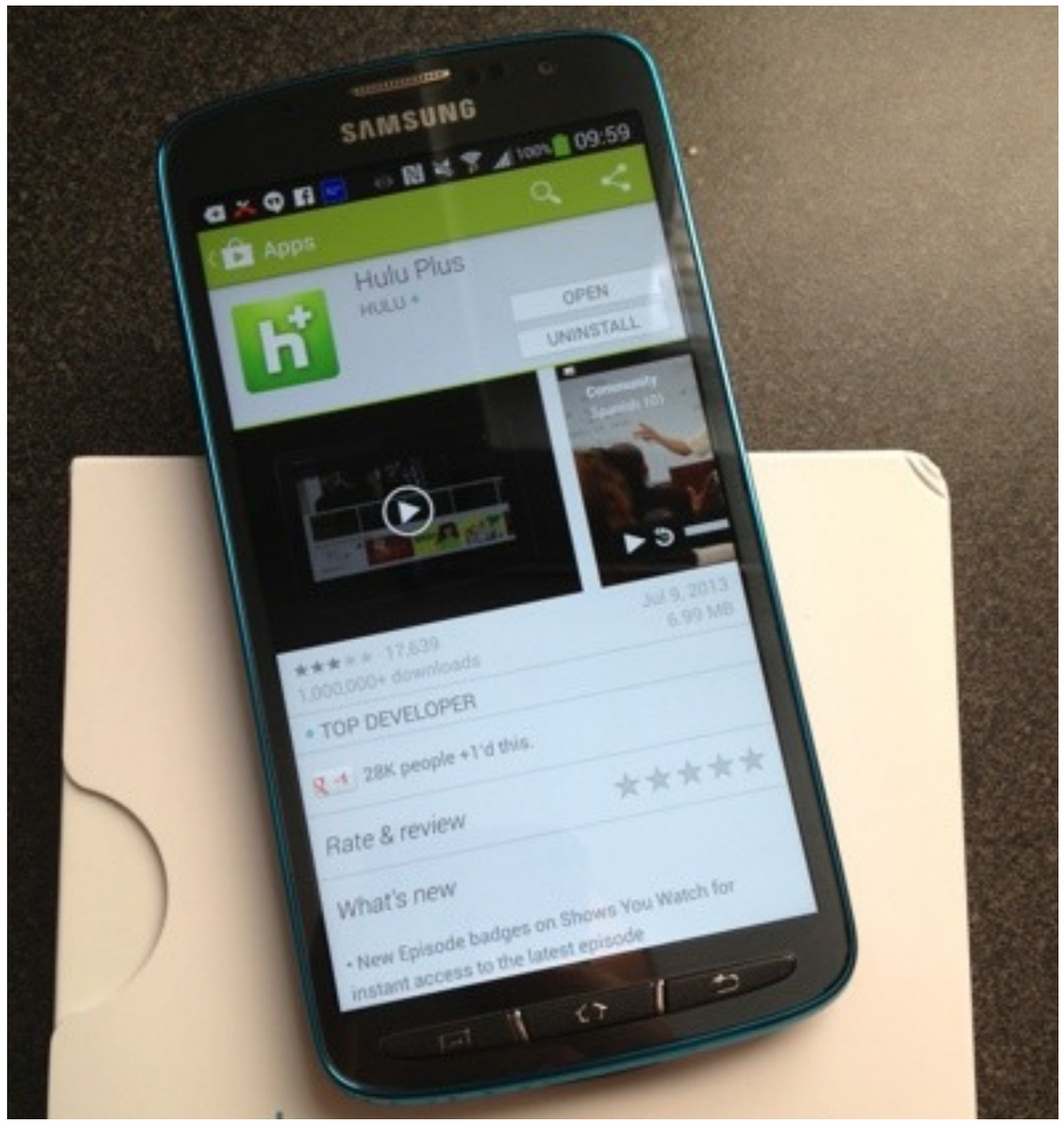




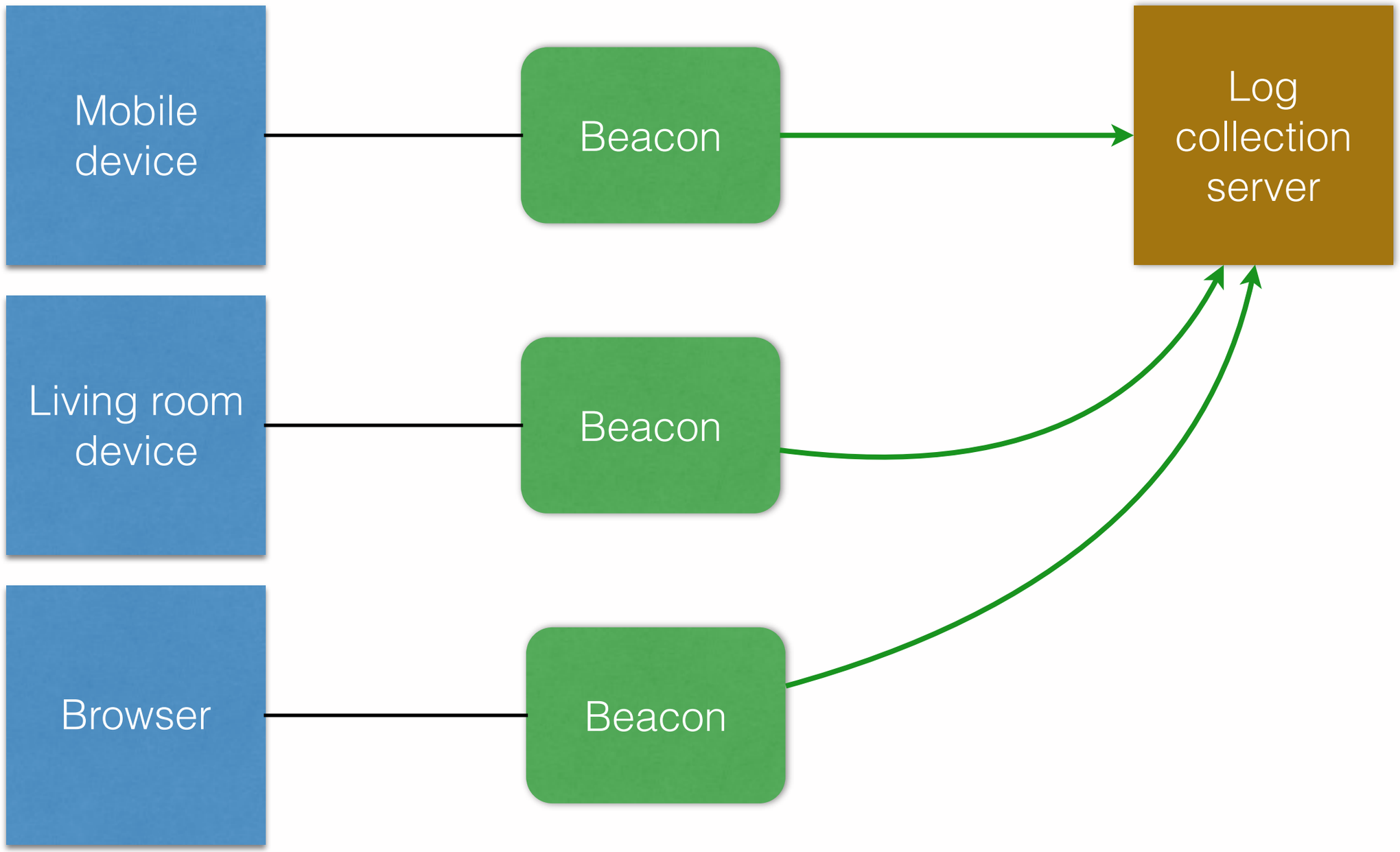
hulu™

- Streaming video service
- Next-day same-season TV
- ~4.5 million subscribers
- ~25 million unique visitors per month
- > 1 billion ads/month





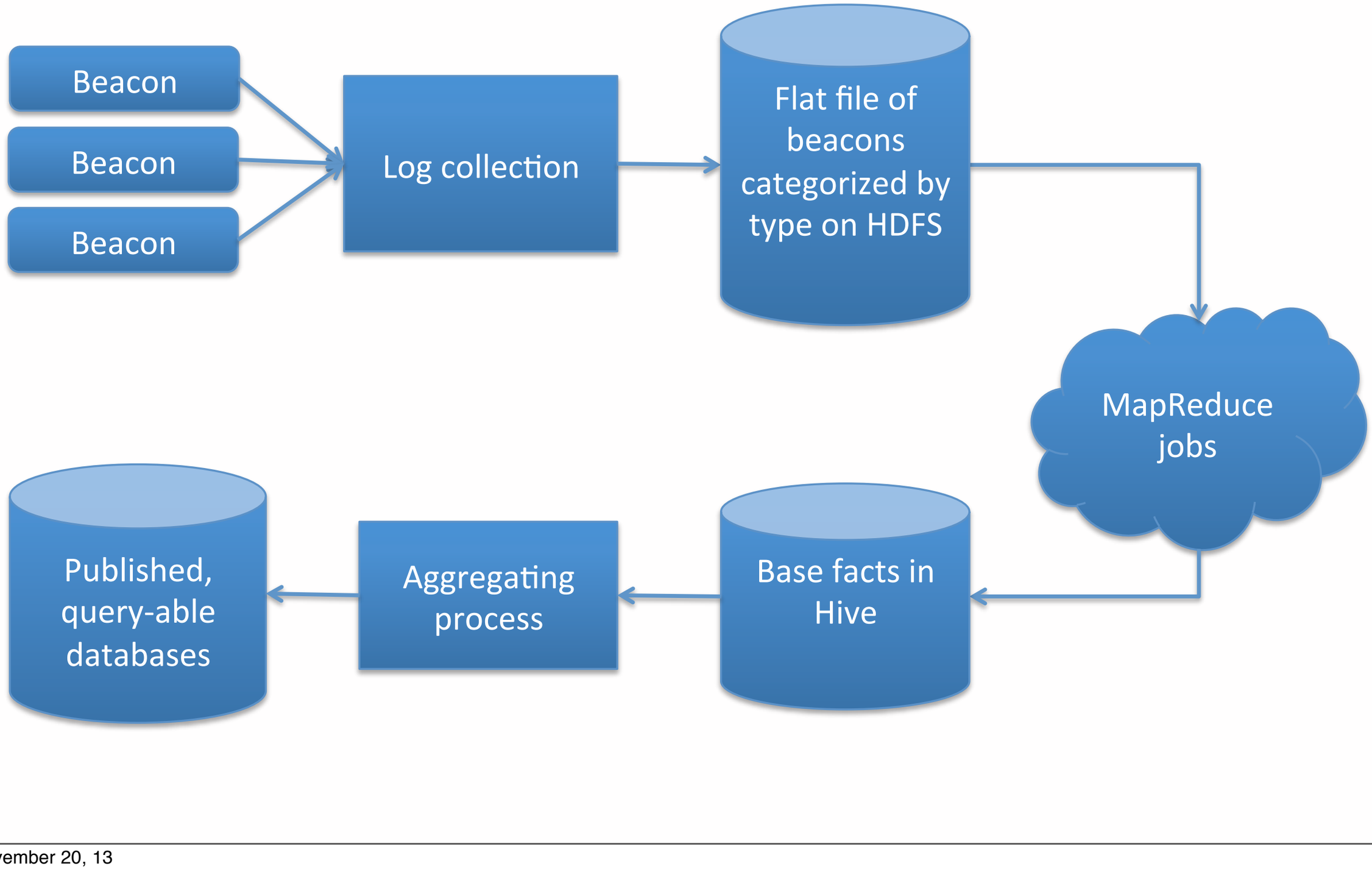
Beacons


Mobile device
Beacon
Log collection server
Living room device
Beacon
Browser
Beacon

What's in a beacon

80 2013-04-01 00:00:00
/v3/playback/start?
bitrate=650
&cdn=Akamai
&channel=Anime
&clichéent=Explorer
&computerguid=EA8FA1000232B8F6986C3E0BE55E9333
&contentid=5003673
…

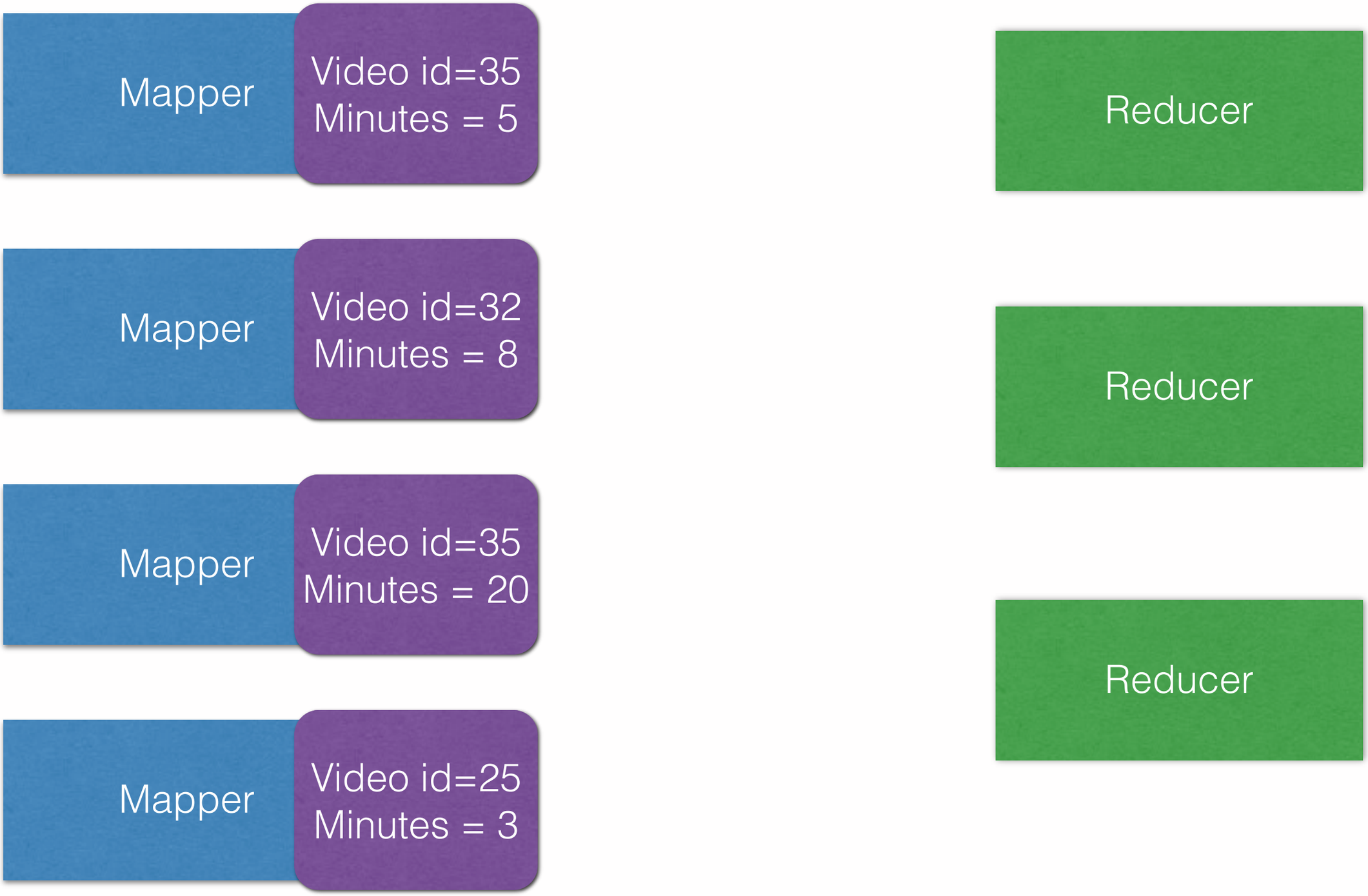
Data pipeline



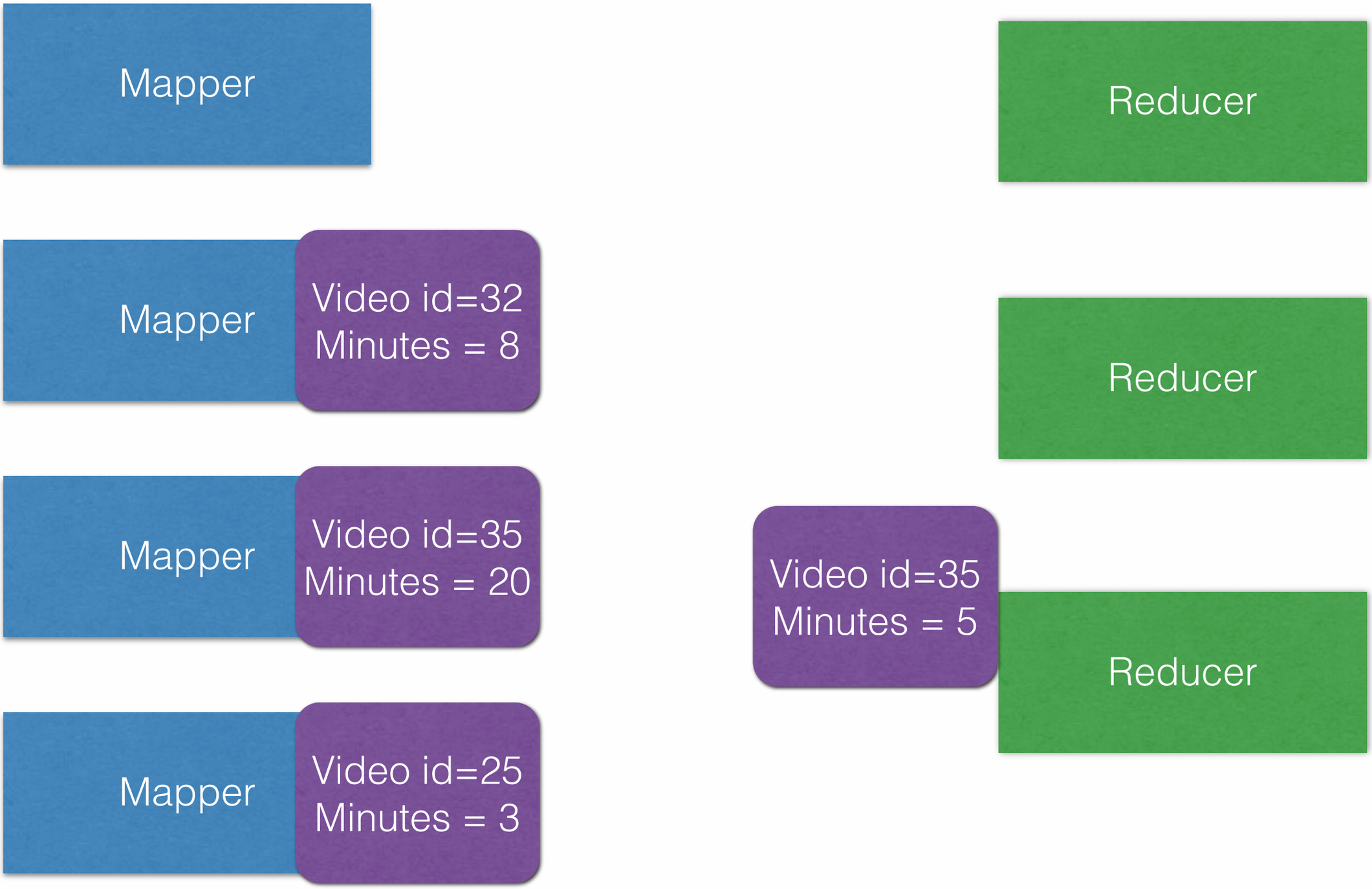
Base-facts

computerguid	***00-755925-925755-925***
userid	5238518
video_id	***289696***
content_partner_id	398
distribution_partner_id	***602***
distro_platform_id	14
is_on_hulu	***0***
onhulu_offhulu_plus_id	2
package_id	***1000***
…	
hourid	***383149***
watched	76426

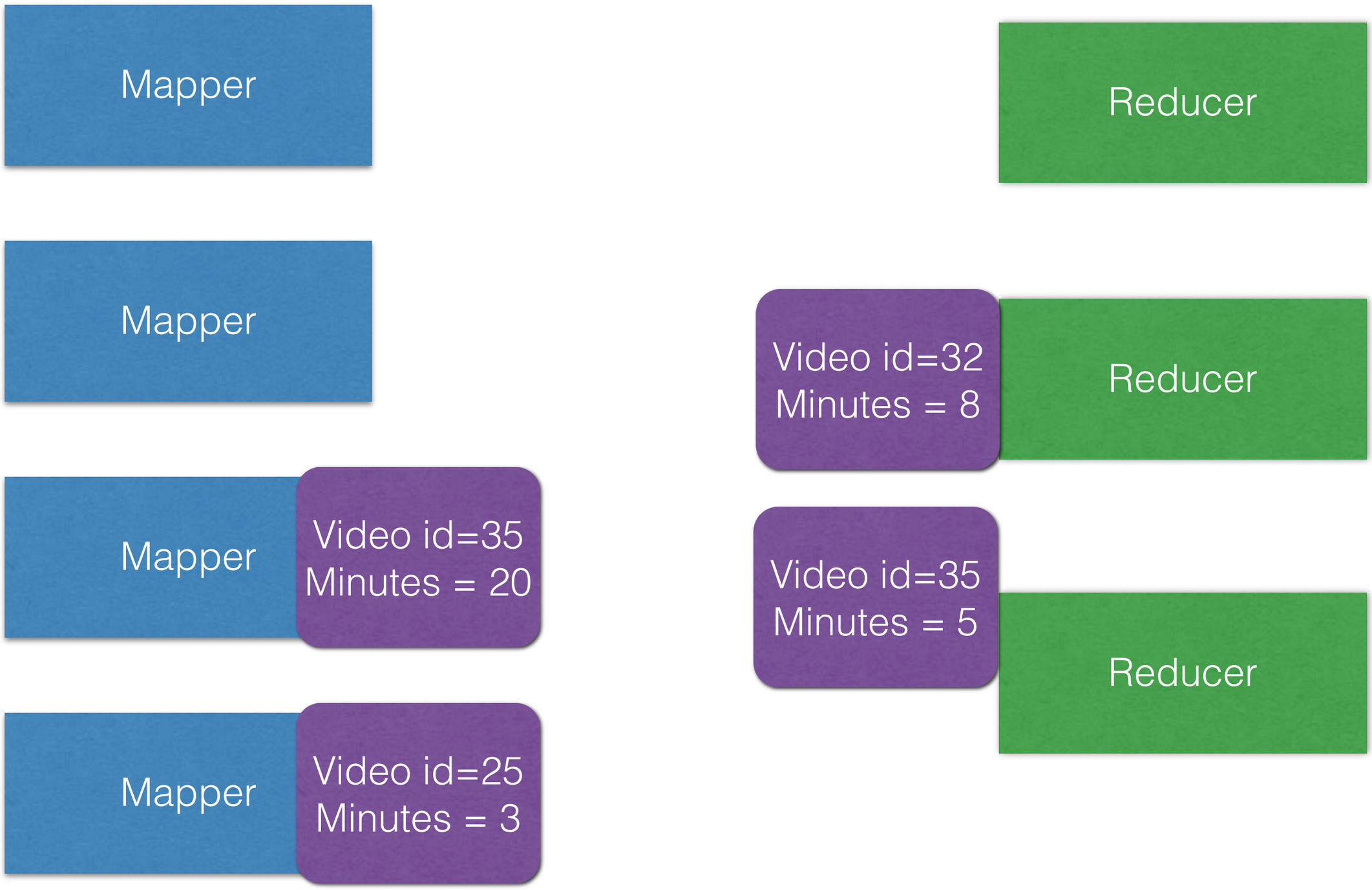
Hadoop & MapReduce



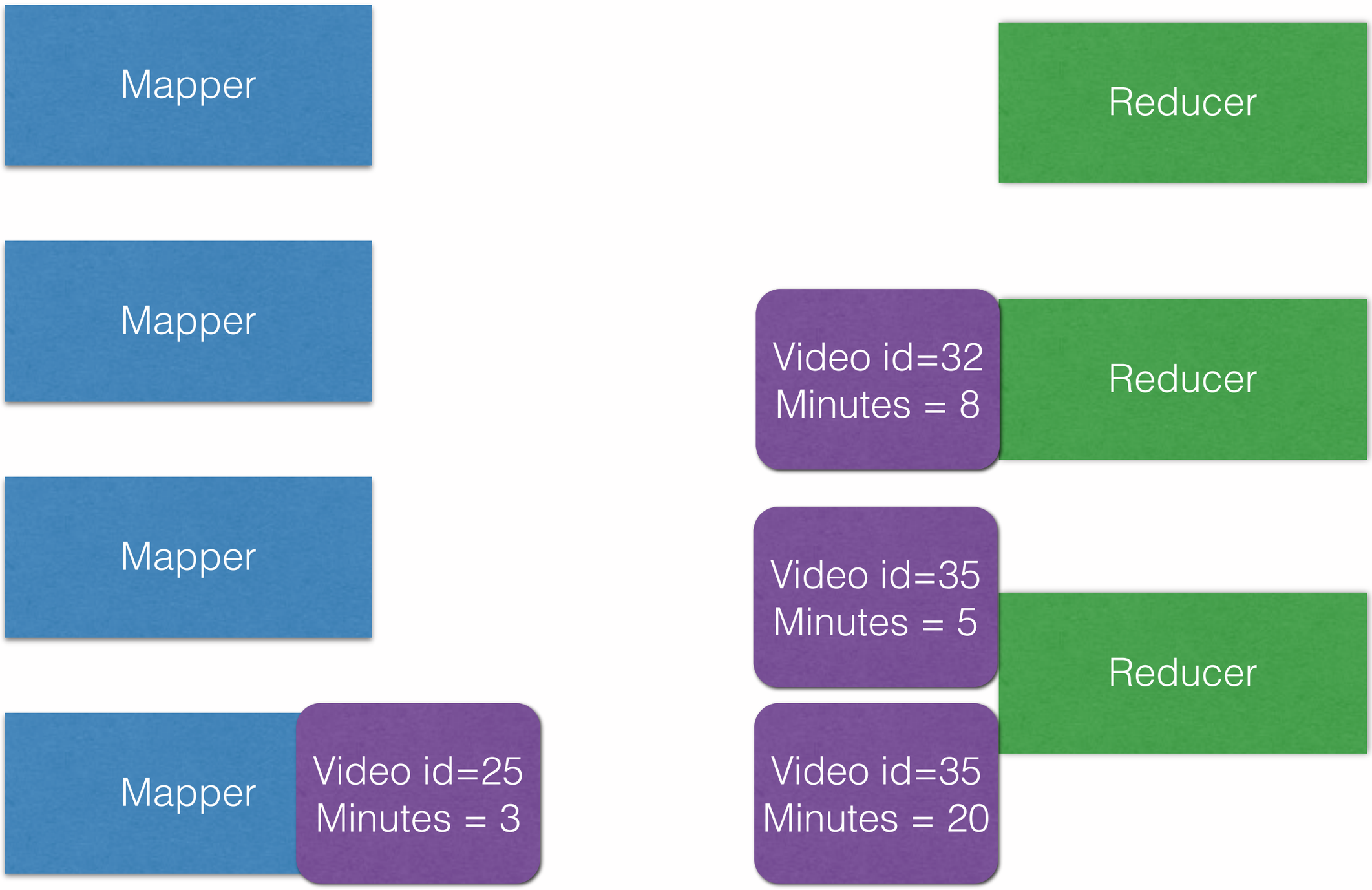
Hadoop & MapReduce



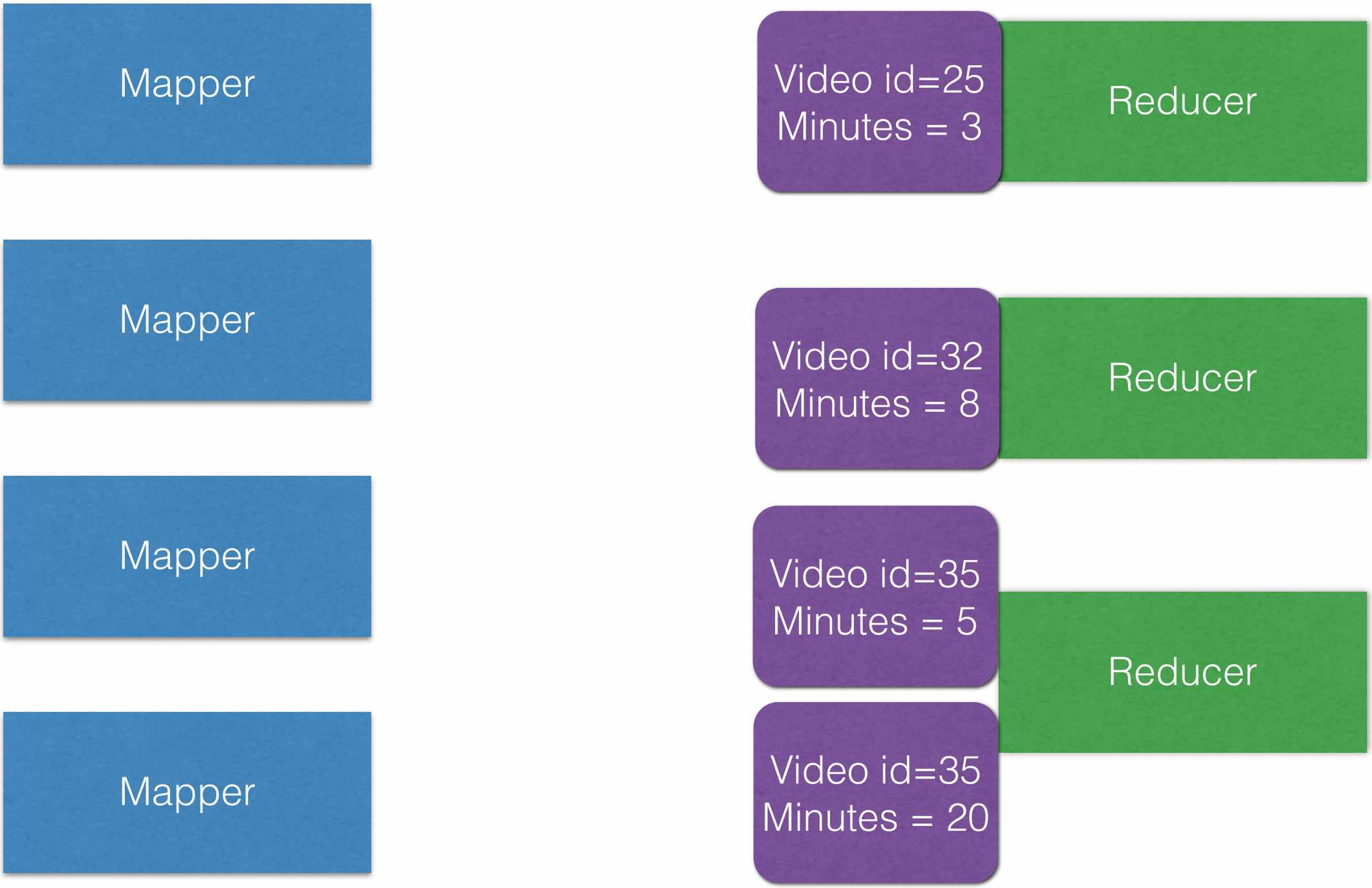
Hadoop & MapReduce



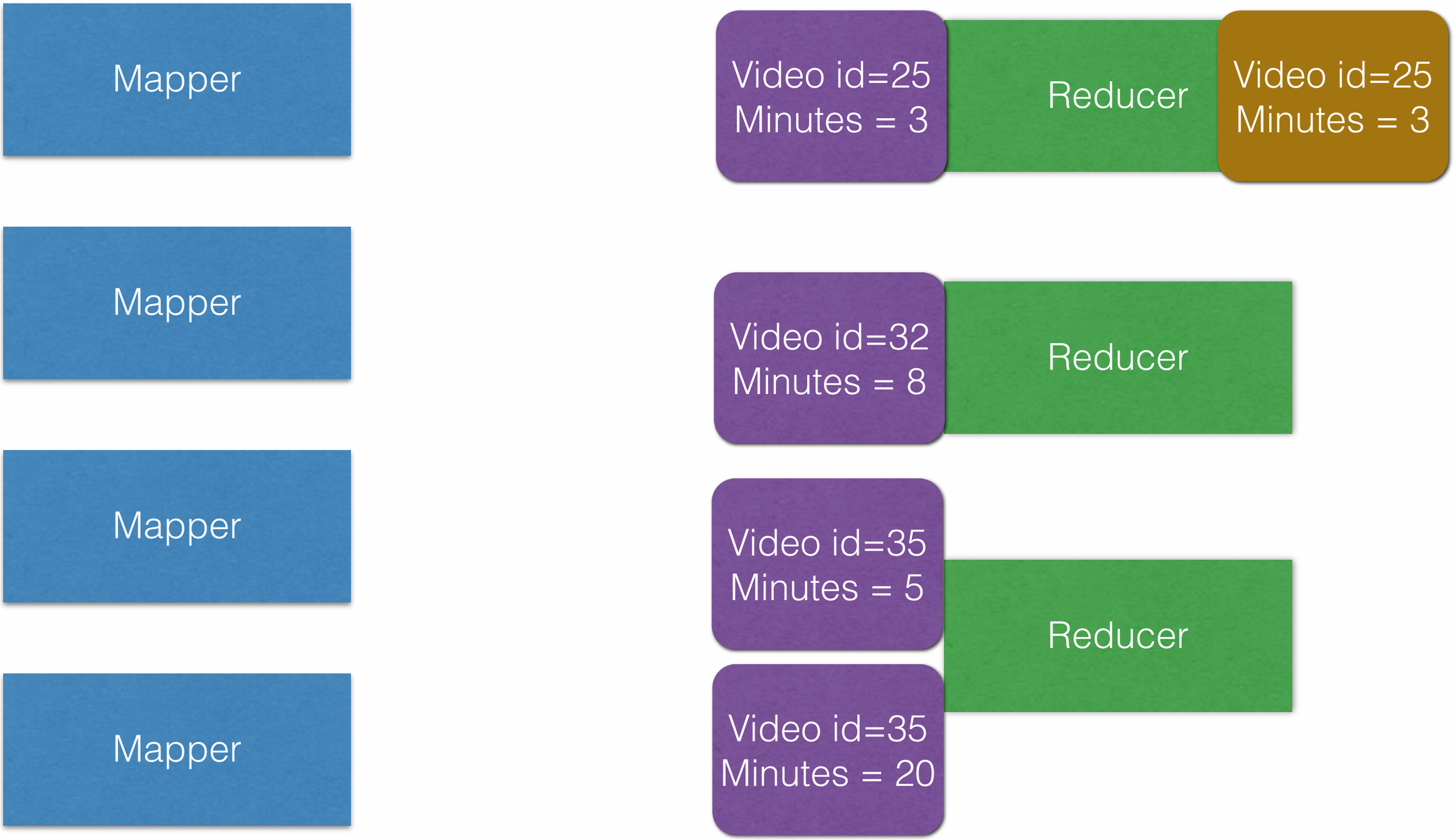
Hadoop & MapReduce



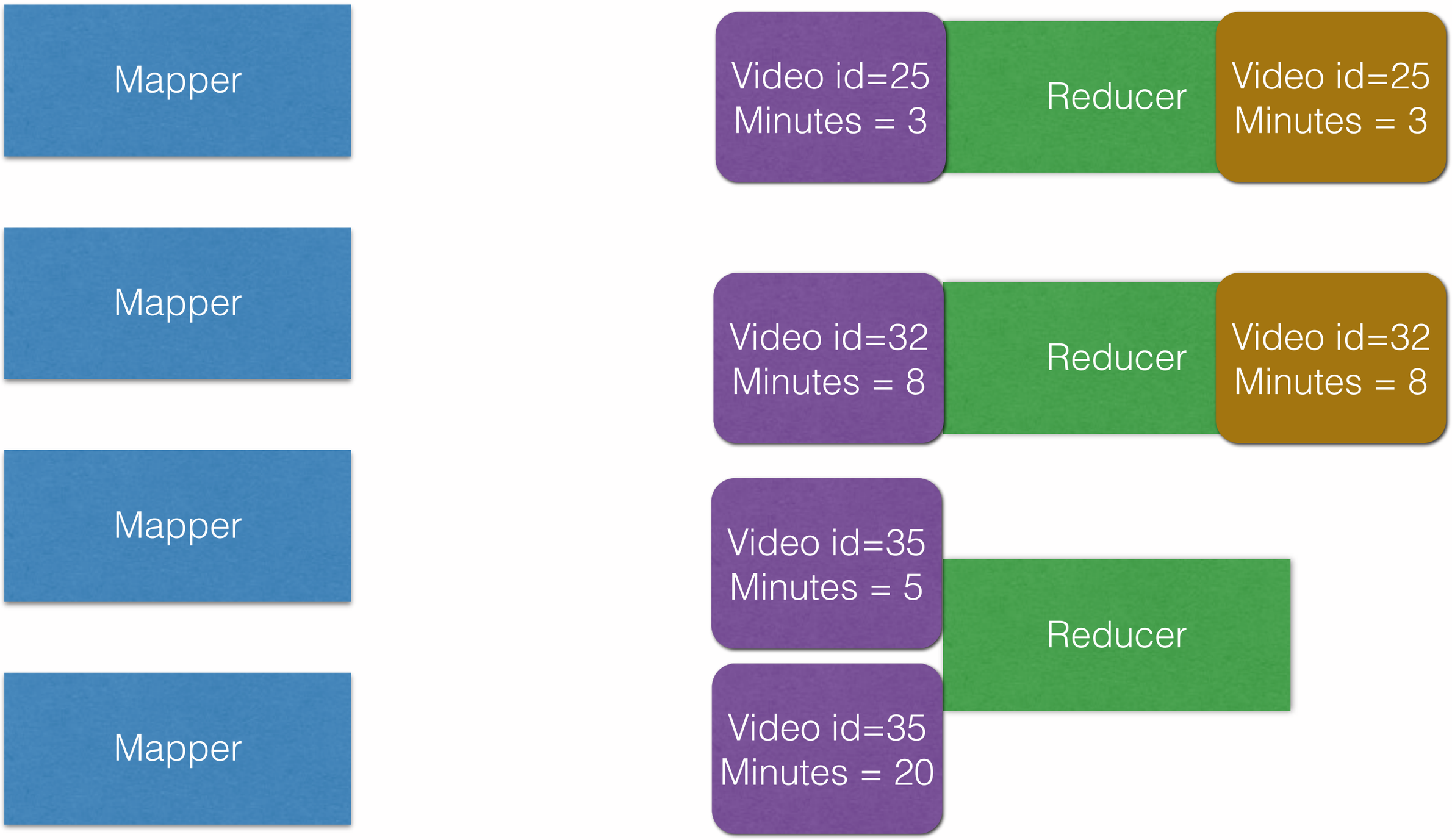
Hadoop & MapReduce



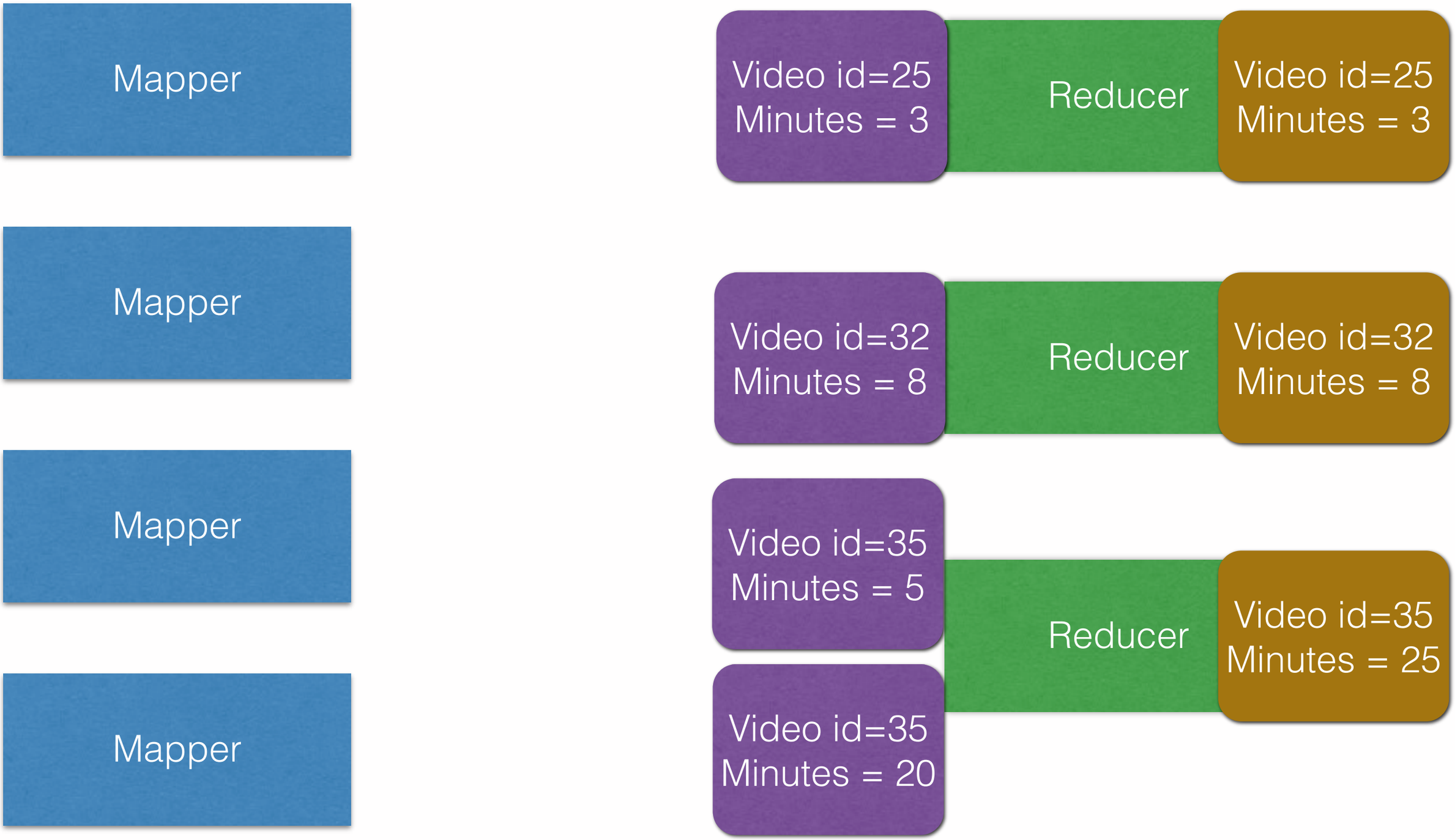
Hadoop & MapReduce



Hadoop & MapReduce

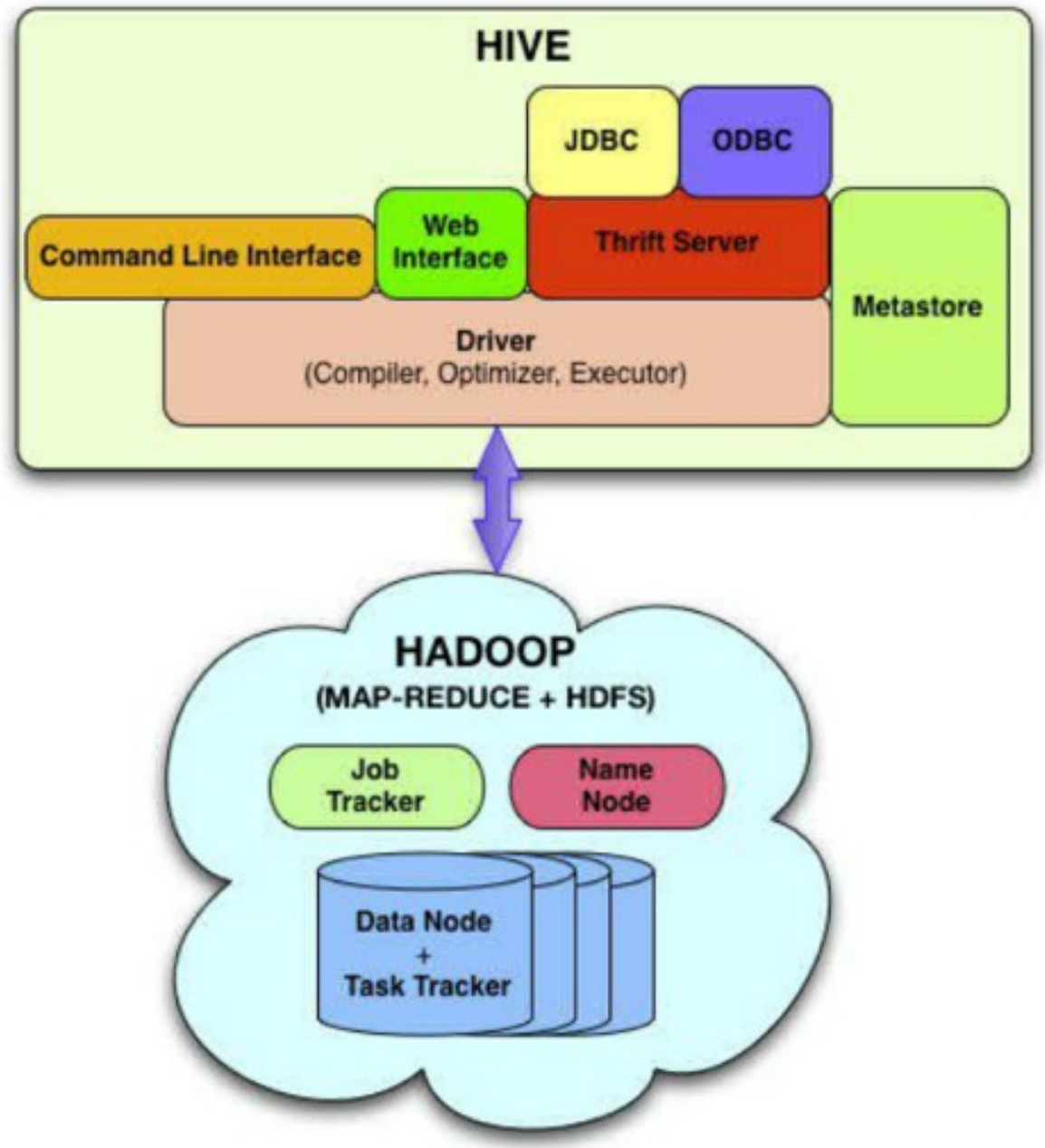


Hadoop & MapReduce



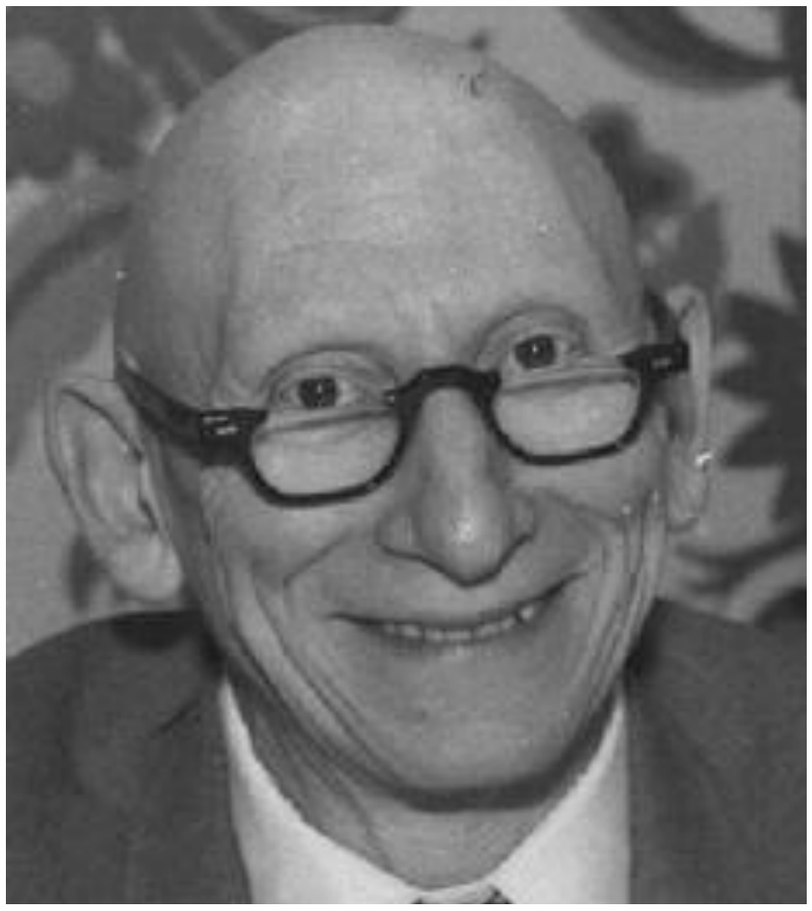
Aggregations

- Base-facts live in relational tables in HIVE
- Aggregations are now a matter of performing the right HiveQL queries once data is ready



Scale vs complexity

"If a program manipulates a large amount of data, it does so in a small number of ways"
- Alan Perlis



Going from beacons to base-facts

- So what do the vast majority of our MapReduce jobs do?
  - Select a set of dimensions we are concerned about
  - Clean up any malformed beacons
  - Perform joins against metadata tables
  - Group by the dimensions and aggregate on some attribute (for example, minutes watched)

ck_Start_UniquesByPlan.java ×

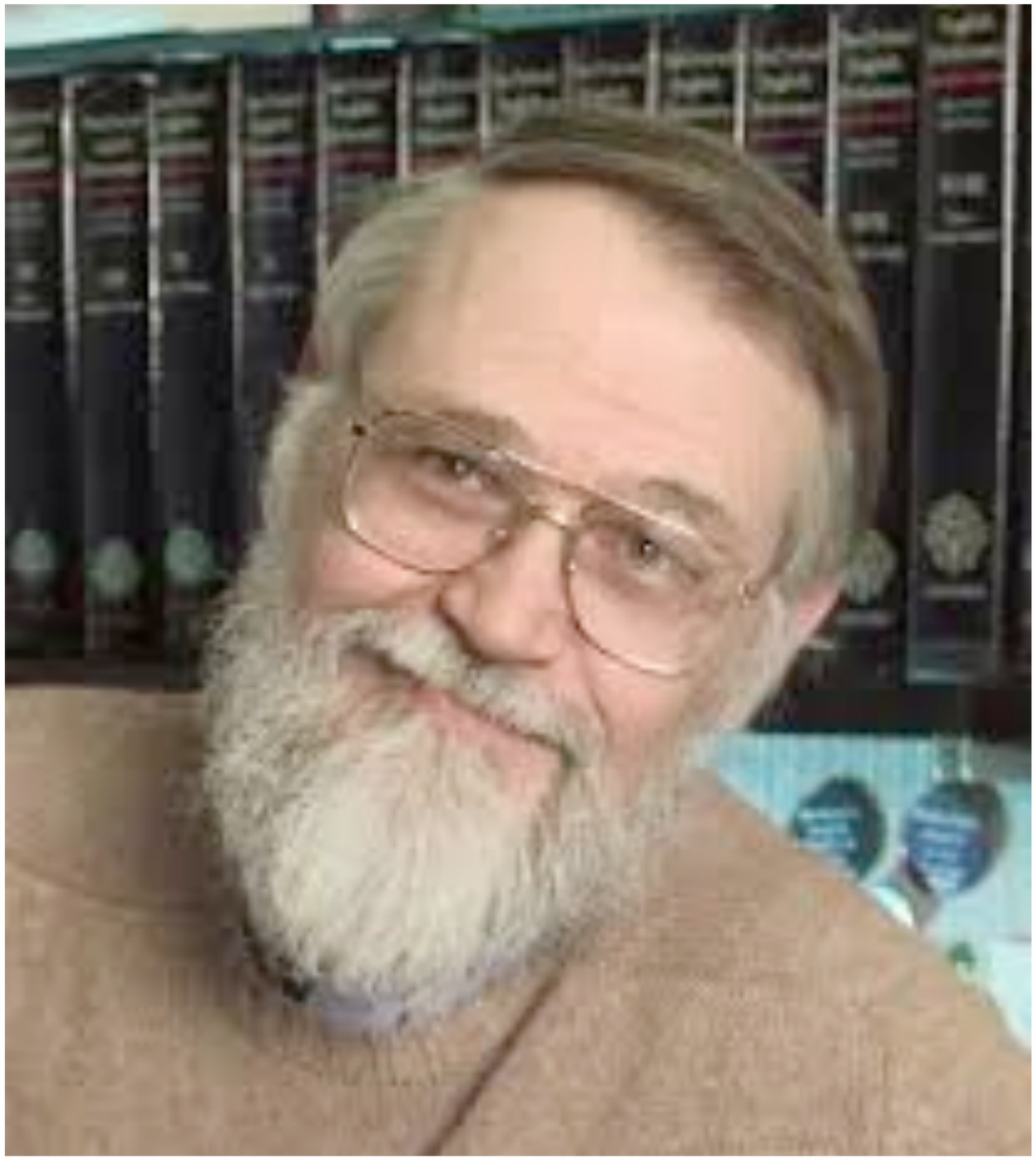
```
public void map(LongWritable inKey, Text inValue, OutputCollector<KeyYMDPlanComputerGuid, LongWritable> output, Reporter reporter) throws IOExceptio
{
    //get the date and URL from value

    String[] parts = inValue.toString().split("\t", -1); //date, time, IP, event
    if (parts.length < 4)
    {
        reporter.incrCounter("Task-Parent", "Invalid line - not enough parts", 1);
        return;
    }

    String eventPart = parts[3];
    String datePart = parts[0];

    if (true == eventPart.matches(".*playback/start.*"))
    {
        reporter.incrCounter("Task-Parent", "Included line - Revenue Start", 1);
    }
    else
    {
        reporter.incrCounter("Task-Parent", "Non-included line", 1);
        return;
    }

    //parse the URL
    final Map<String,String> queryMap = UrlHelper.getQueryMap(eventPart);

    String planidstr = queryMap.get("planid");
    String plan = "";
    if (null == planidstr || planidstr.equals("0"))
    {
        plan = "Classic";
    }
    else
    {
        plan = "Plus";
    }

    String computerguid = queryMap.get("computerguid");
    if (null == computerguid || computerguid.length() < 32)
    {
        reporter.incrCounter("Task-Parent", "Missing computerguid", 1);
        return;
    }

    String[] dateParts = datePart.split("-", -1);
    if (dateParts.length != 3)
    {
        return;
    }
```

So what's the problem?

- Too much complexity, we are forced to reason about both concurrency and data transformations at the same time

- What we'd like to do is achieve a separation of concerns

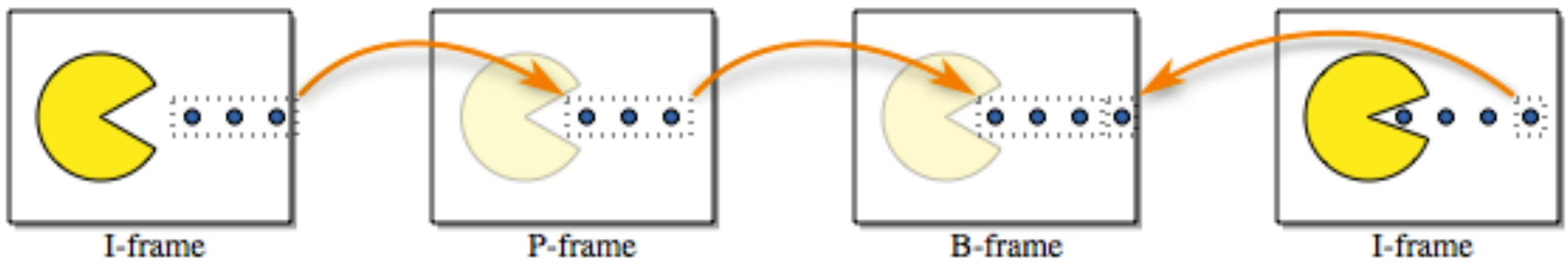
“Controlling complexity is the essence of computer programming”
-Brian Kernighan



The third way

Every domain has its own language

- Example: Video processing
  - I-frame
  - P-frame
  - B-frame



What is the vocabulary of our language?

```
basefact playback_watched_uniques from playback/(position|end) {
    dimension harpyhour.id as hourid;
    dimension computerguid as computerguid;
    dimension userid as userid;
    required dimension video.id as video_id;
    required dimension contentPartner.id as content_partner_id;
    required dimension distributionPartner.id as distribution_partner_id;
    required dimension distributionPlatform.id as distro_platform_id;
    dimension packageAvailability.chosen as package_availability;
    …
    dimension siteSessionId.chosen as site_session_id;
    dimension facebook.isfacebookconnected as is_facebook_connected;
    fact sum(watched.out) as watched;
}
```

Why didn't we just use Pig?

- Dataflow language - runs on Hadoop
- Pig philosophy (taken from Apache's website):
  - Pigs eat anything
  - Pigs live anywhere
  - Pigs are domestic animals
  - Pigs can fly

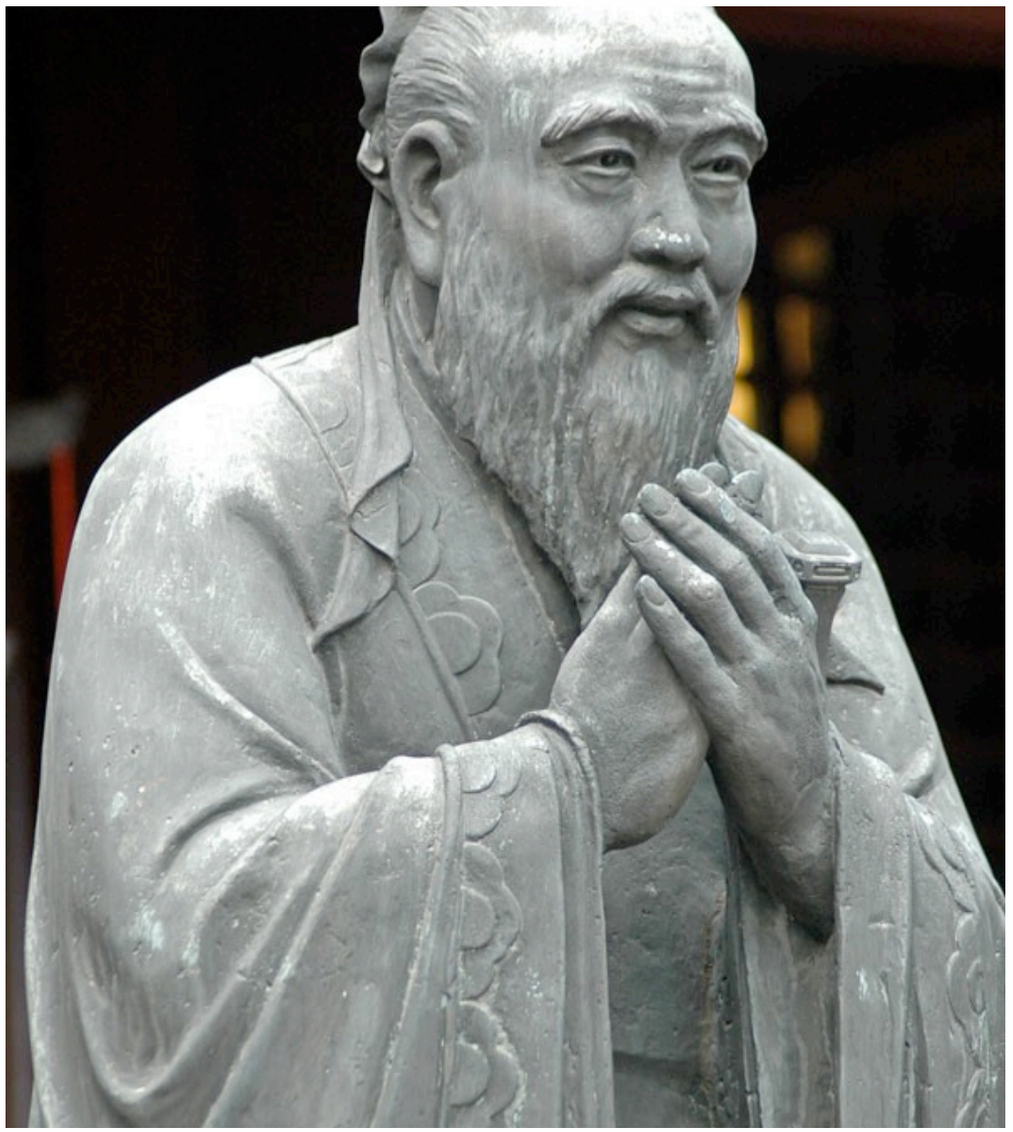
```
/* myscript.pig
My script is simple.
It includes three Pig Latin statements.
*/

A = LOAD 'student' USING PigStorage() AS (name:chararray, age:int, gpa:float); -- loading data
B = FOREACH A GENERATE name;  -- transforming data
DUMP B;  -- retrieving results
```

“Beware of the Turing tar-pit, in which everything is possible, but nothing of interest is easy”
-Alan Perlis

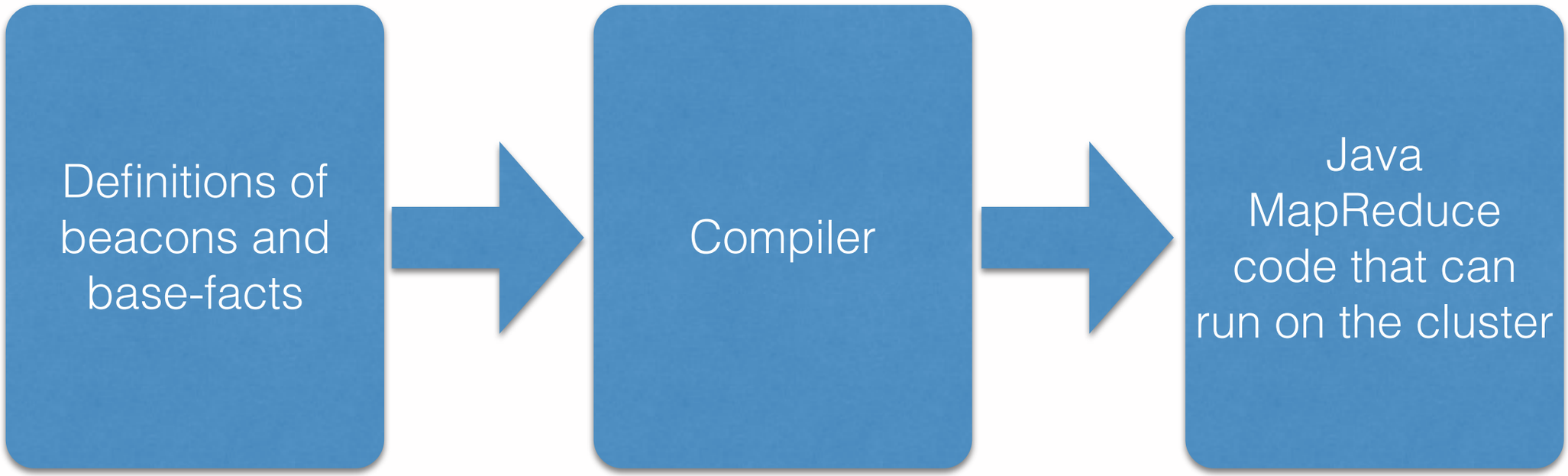
The Tao of Program Design

Confucius say:
"The superior [program] cultivates itself so as to give rest to others"

(paraphrased)



The beaconspec compiler



Simple vs complex dimensions

- Simple dimensions are those that exist in the beacon itself

- Complex dimensions are those that require processing a subset of the input values of the beacon to produce a new value (for example, metadata lookups)

Complex dimension example: campaign tracking

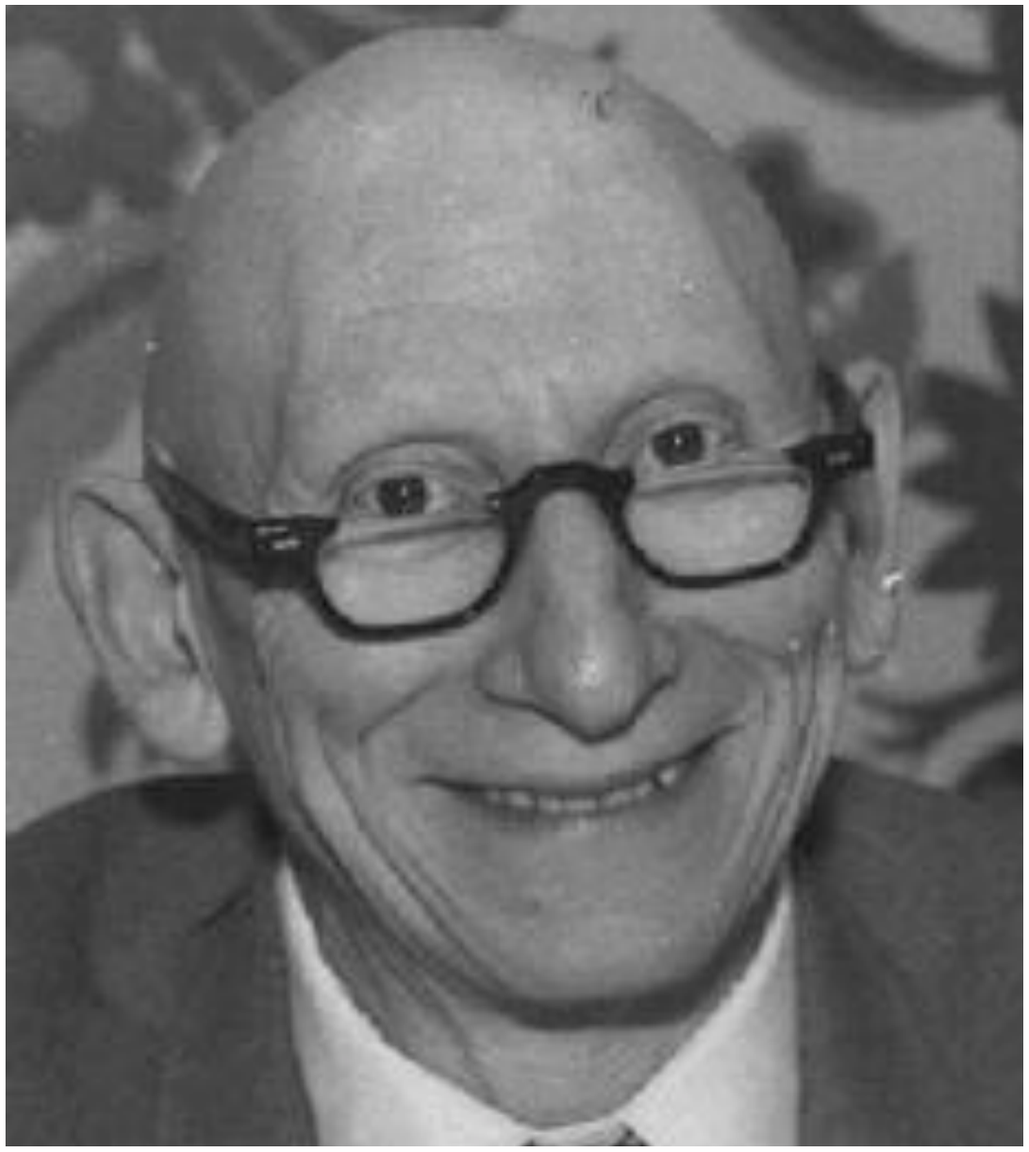
- When a user arrives at our site through an advertising campaign, we associate the user with that campaign

- For the next 90 days, any activity that user performs is linked to that advertising campaign

Advantages

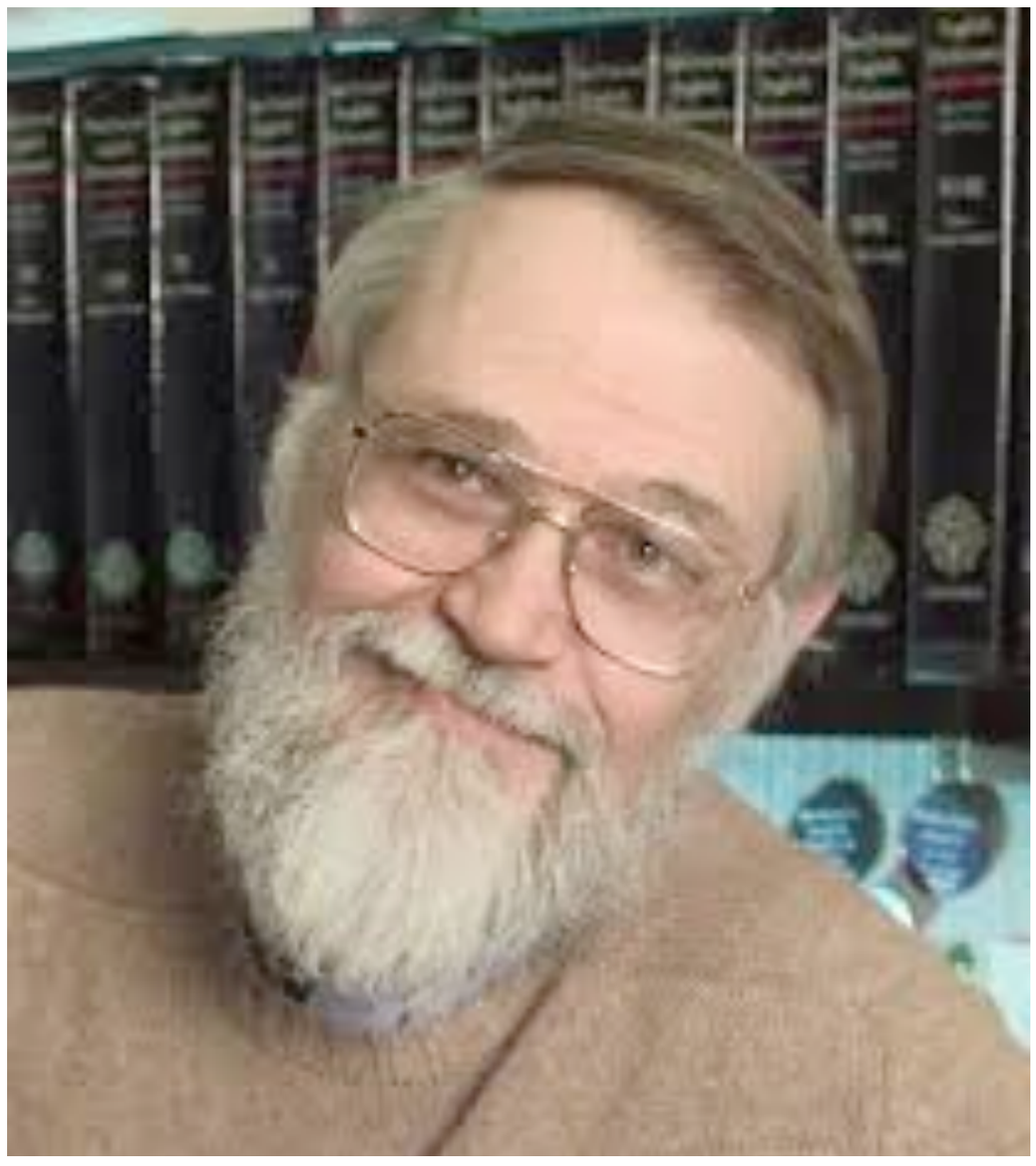
- The majority of the jobs running on our cluster are now authored in beaconspec, resulting in a significant reduction in the size of our codebase
- When upgrading to a new version of Hadoop, all we need to do is upgrade the compiler - the job code is pure logic
- Consistent way of performing routine actions

Cast (in order of appearance):



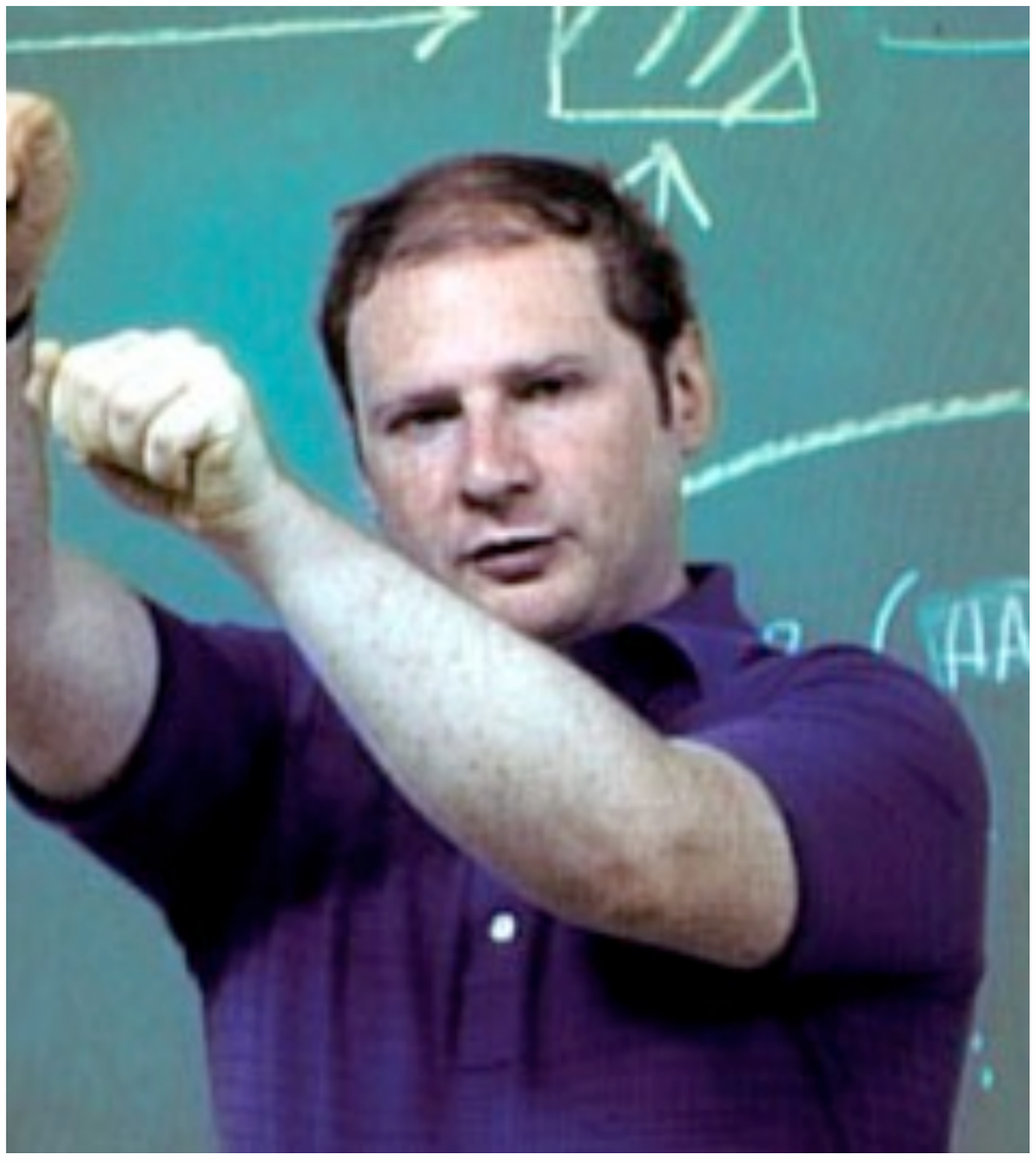
- Alan Perlis
- Founding father of computer science
- ALGOL, Project Whirlwind
- Tweeted before twitter
- “Epigrams in programming”

Cast (continued):



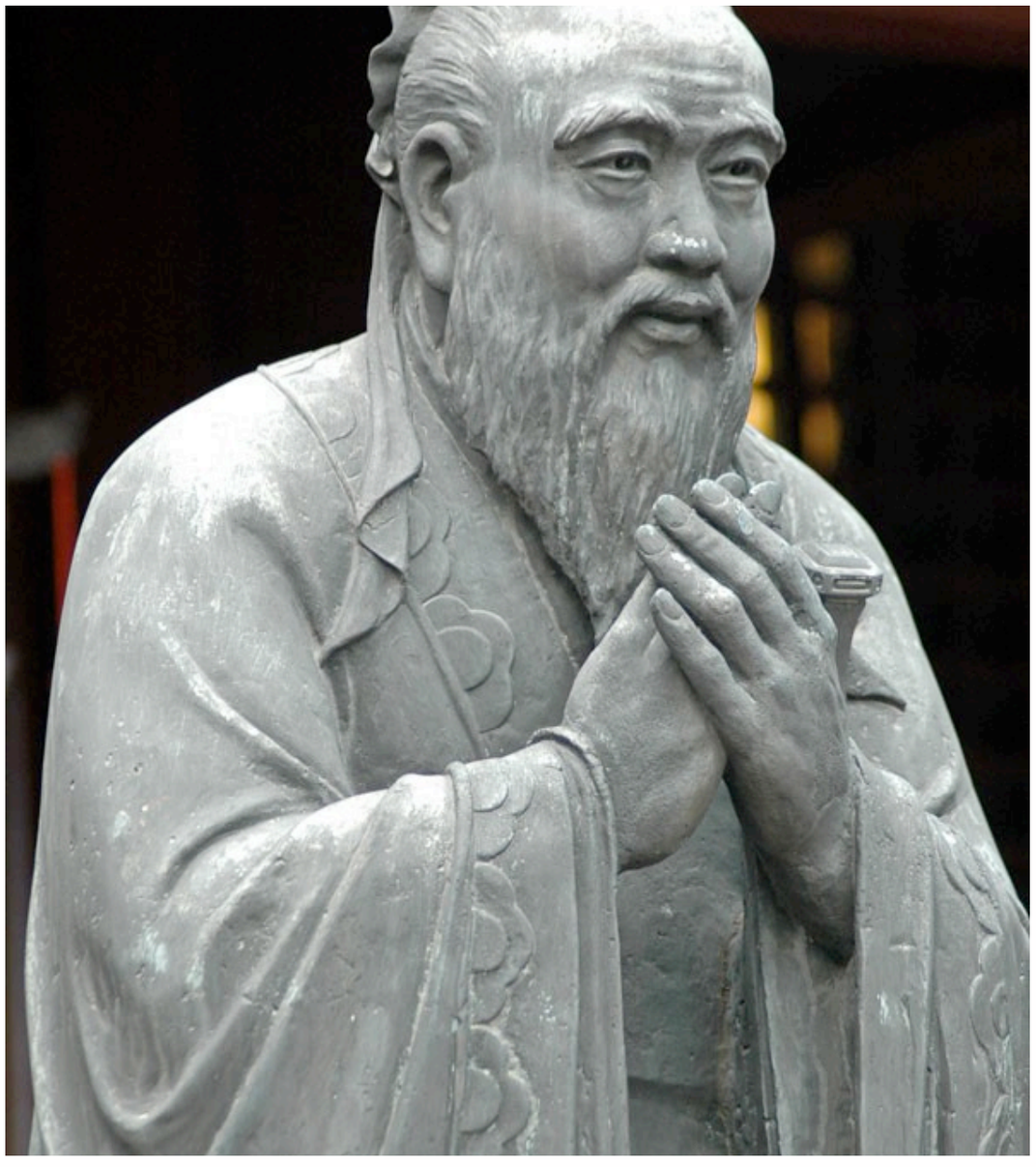
- Brian Kernighan
- Unix co-author
- Developer of ‘cron’
- Known for TCPL
- “The Practice of Programming” (w/ Rob Pike)

Cast (continued):



- Hal Abelson
- MIT Professor
- “Structure and Interpretation of Computer Programs”

Cast (continued):



- Confucius
- Transmitter of the Tao
- “The Way of the Superior Man”

Questions?

If you find yourself writing too much code to do a mundane job, or if you have trouble expressing the process comfortably, maybe you are using the wrong language

- Rob Pike and Brian Kernighan, The Practice of Programming