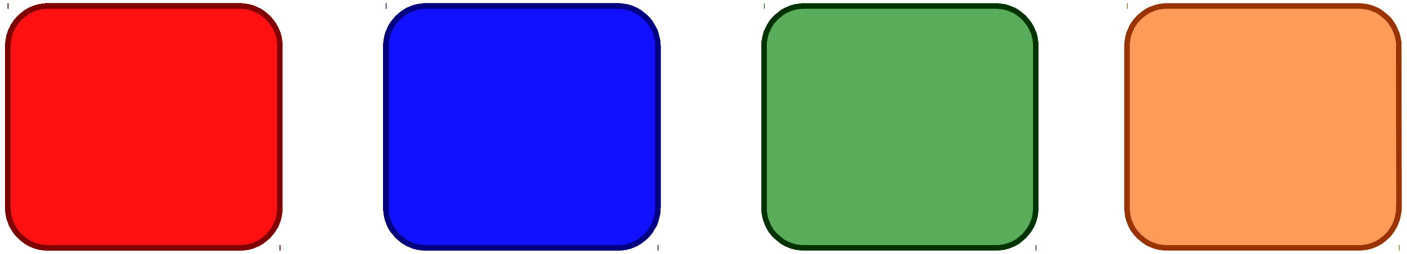
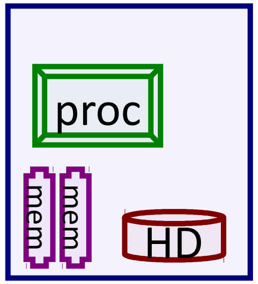
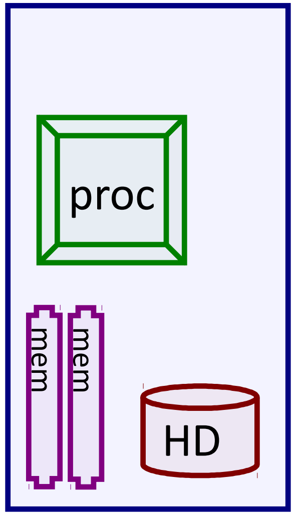
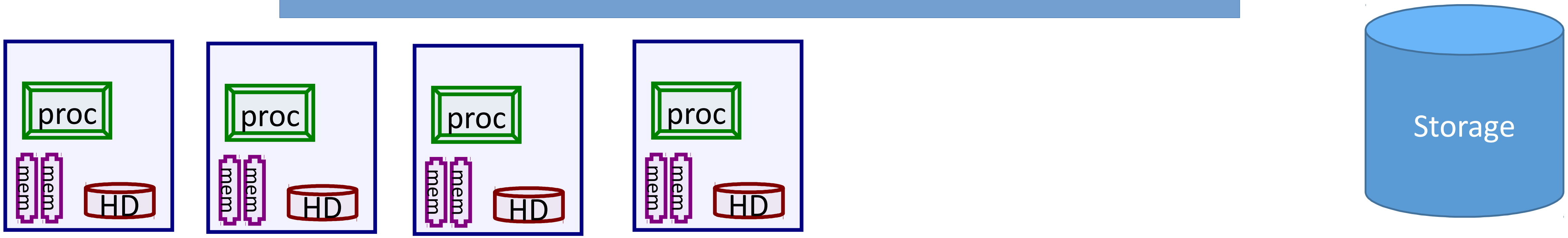



proc
mem
mem
HD


proc
mem
mem
HD

SMAC – Social Mobility Analytics Cloud
ISMAC – IoT, Social, ...



1 TB – _____ hrs

1 PB – _____ days

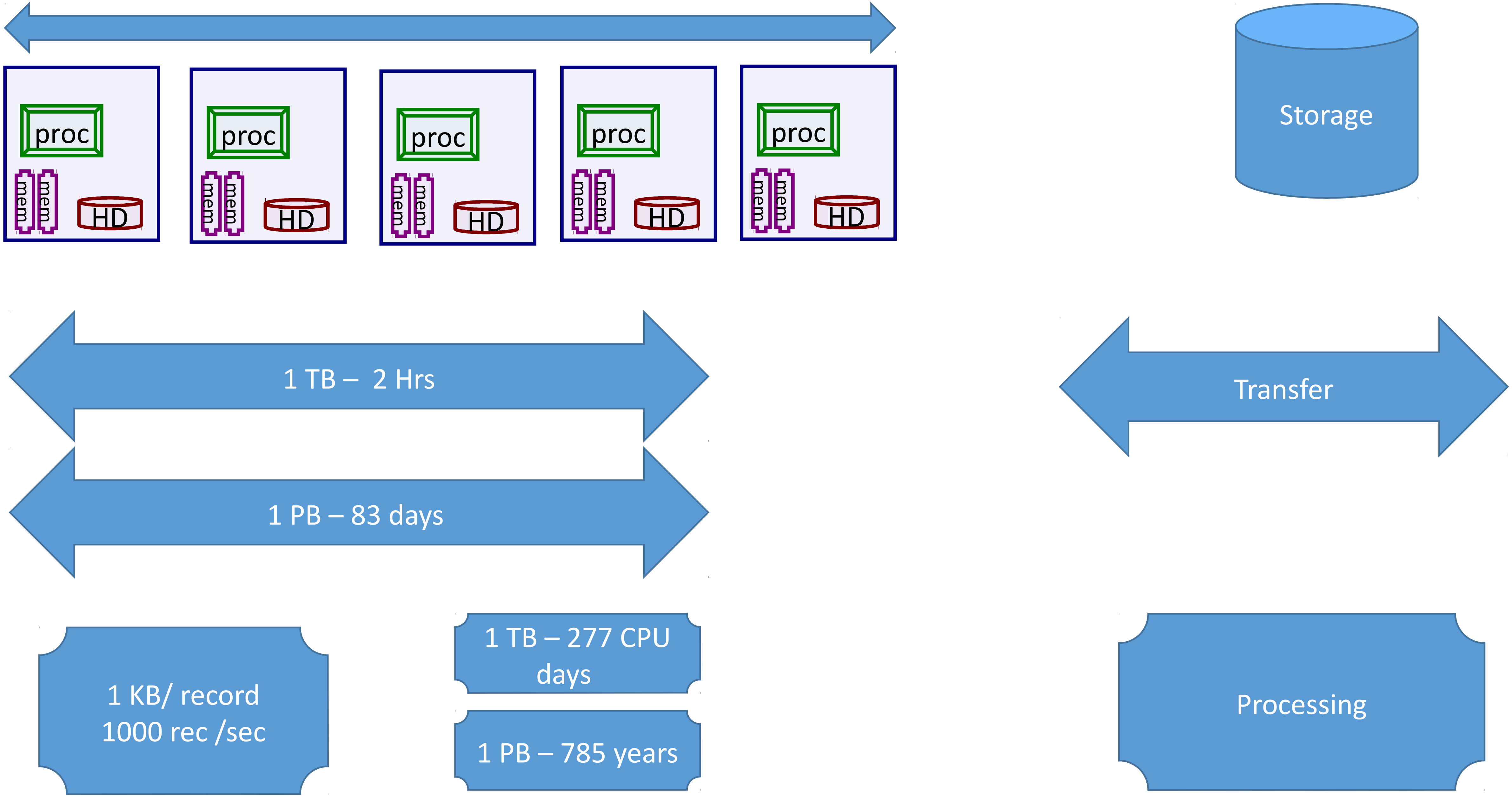
Transfer

1 KB/ record
1000 rec /sec

1 TB - _____
days

1 PB – _____
years

Processing



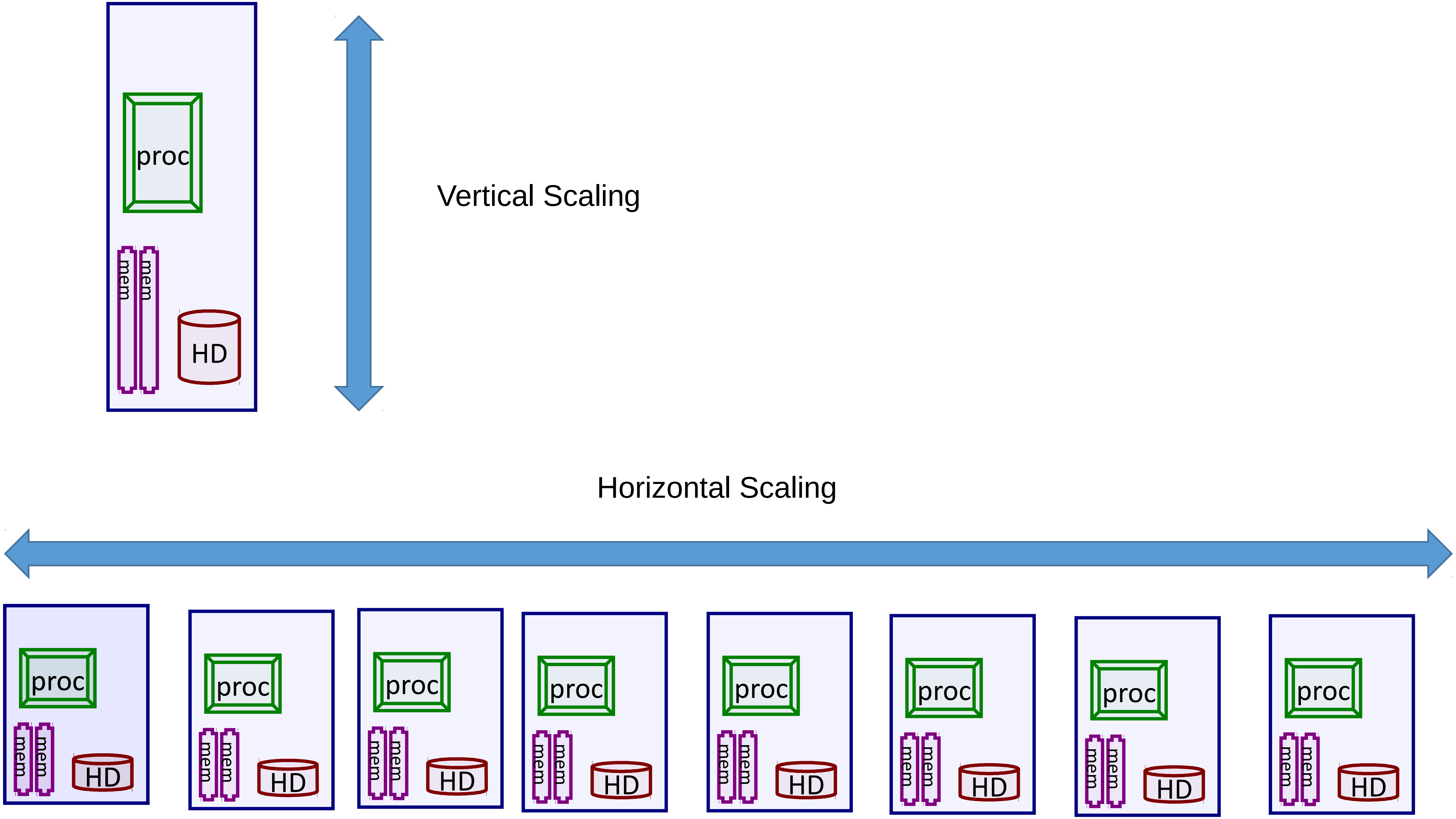
1 TB – 2 Hrs

1 PB – 83 days

1 KB/ record
1000 rec /sec

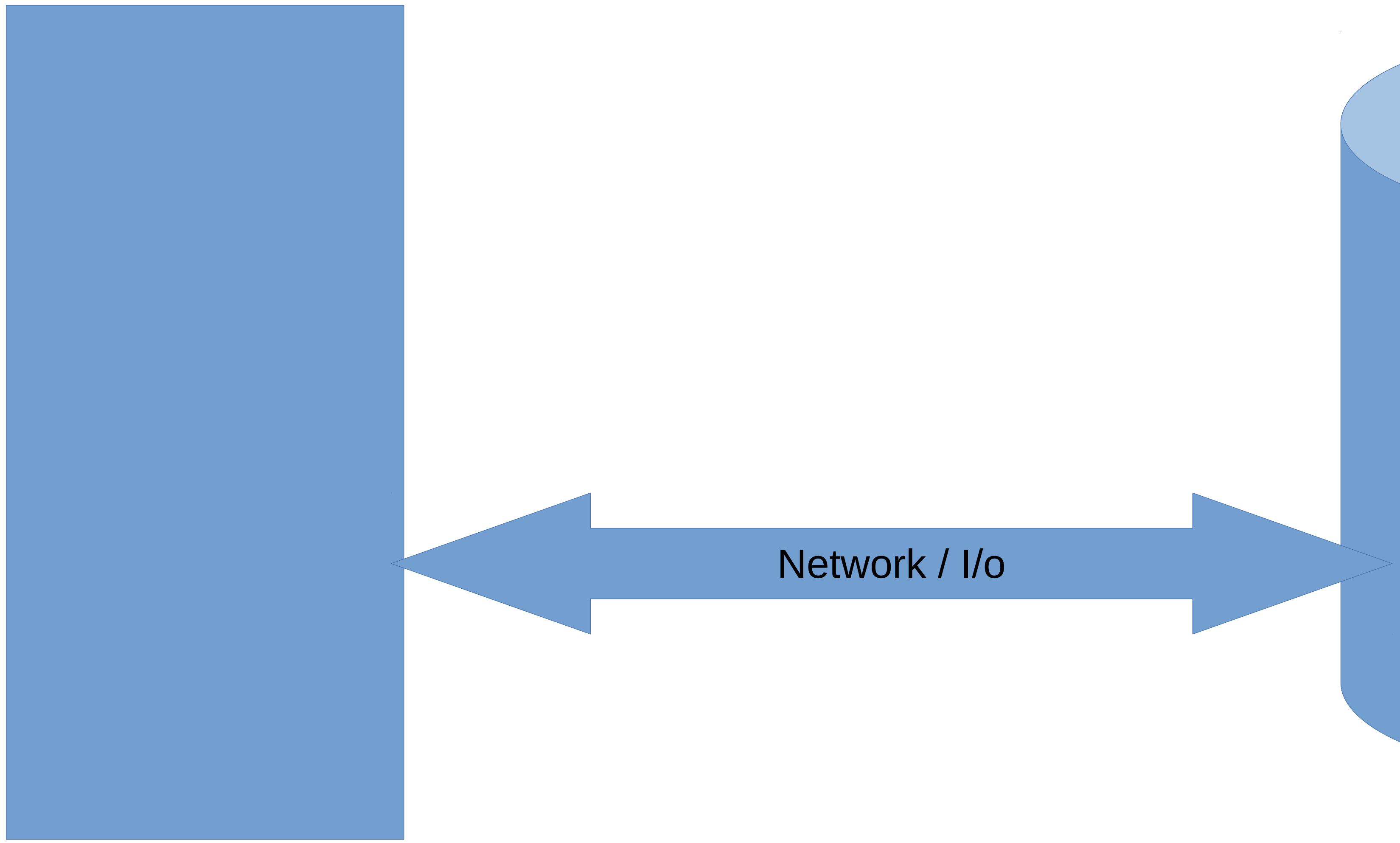
1 TB – 277 CPU days

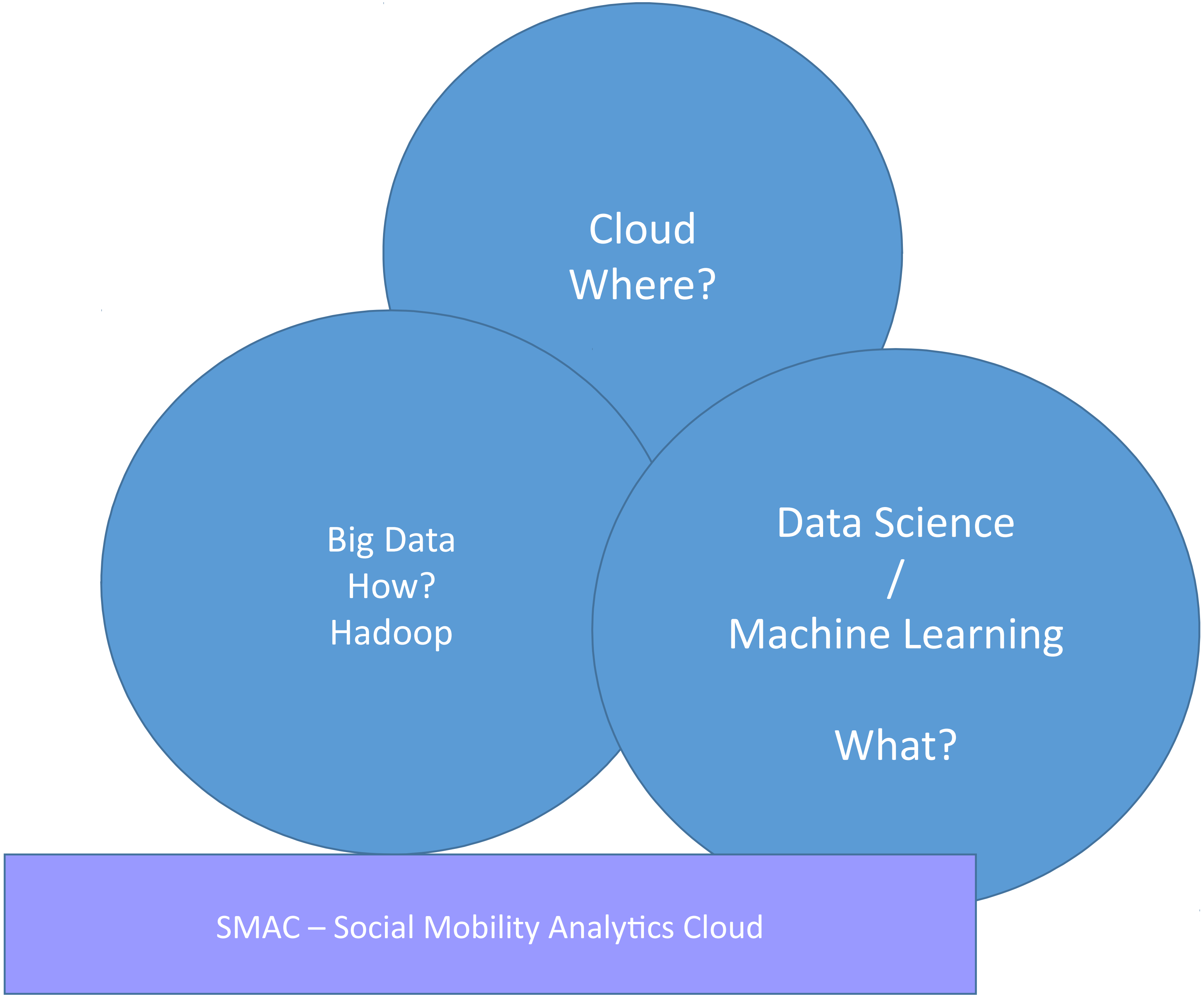
1 PB – 785 years



Vertical Scaling

Horizontal Scaling


Database
TB
Network / I/o



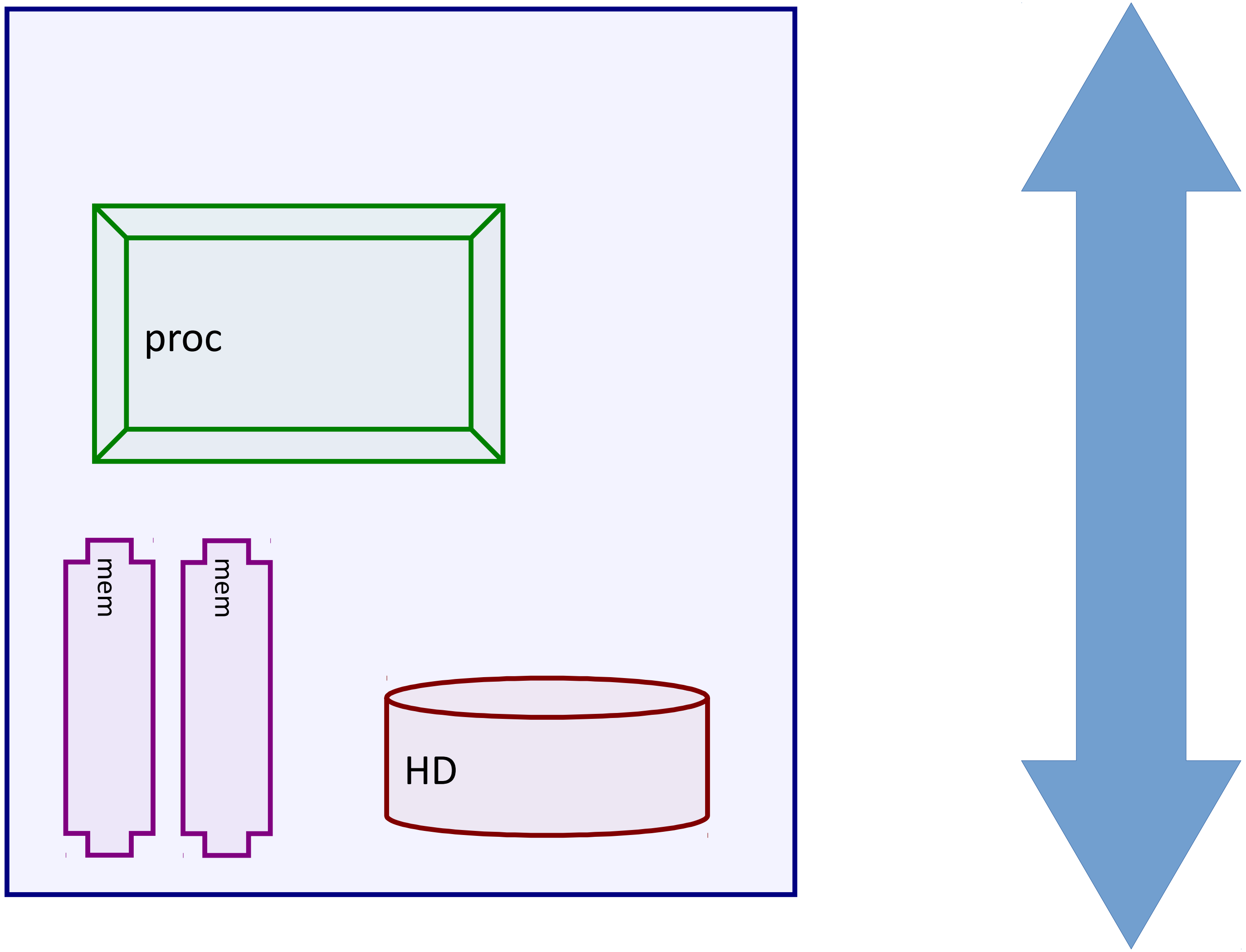
Cloud
Where?

Big Data
How?
Hadoop

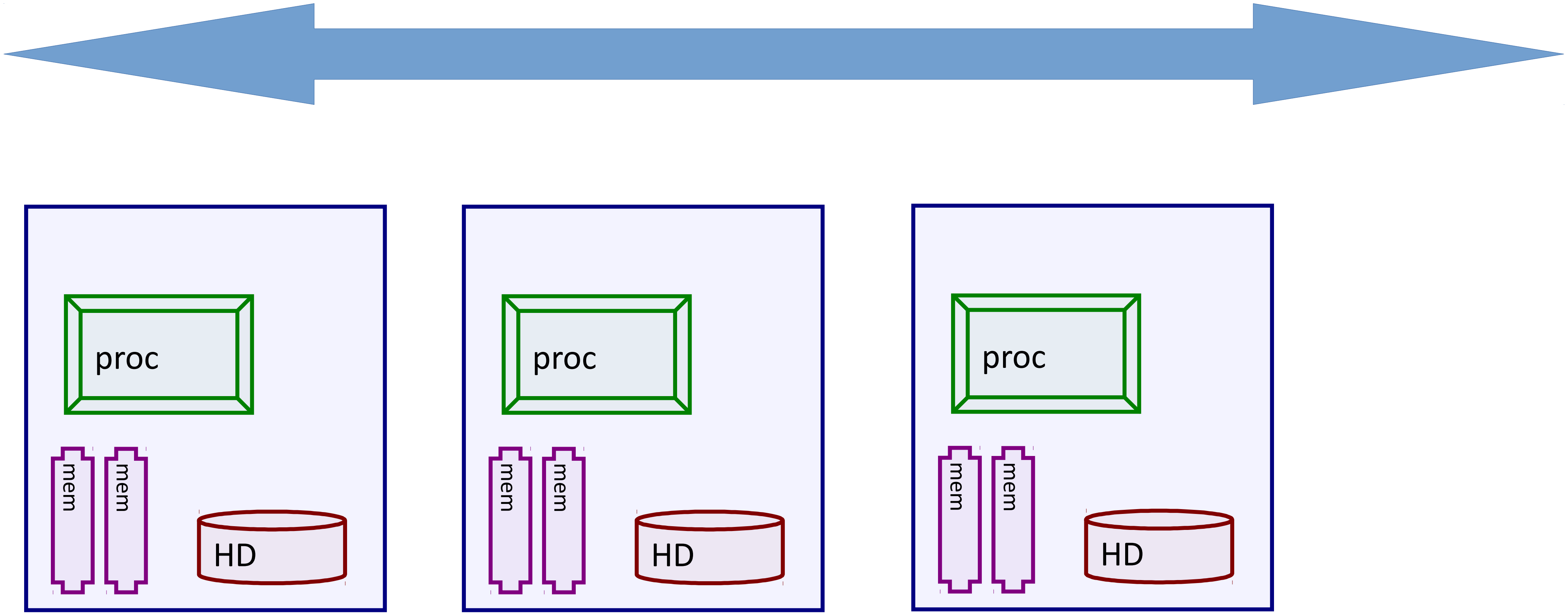
Data Science
/
Machine Learning

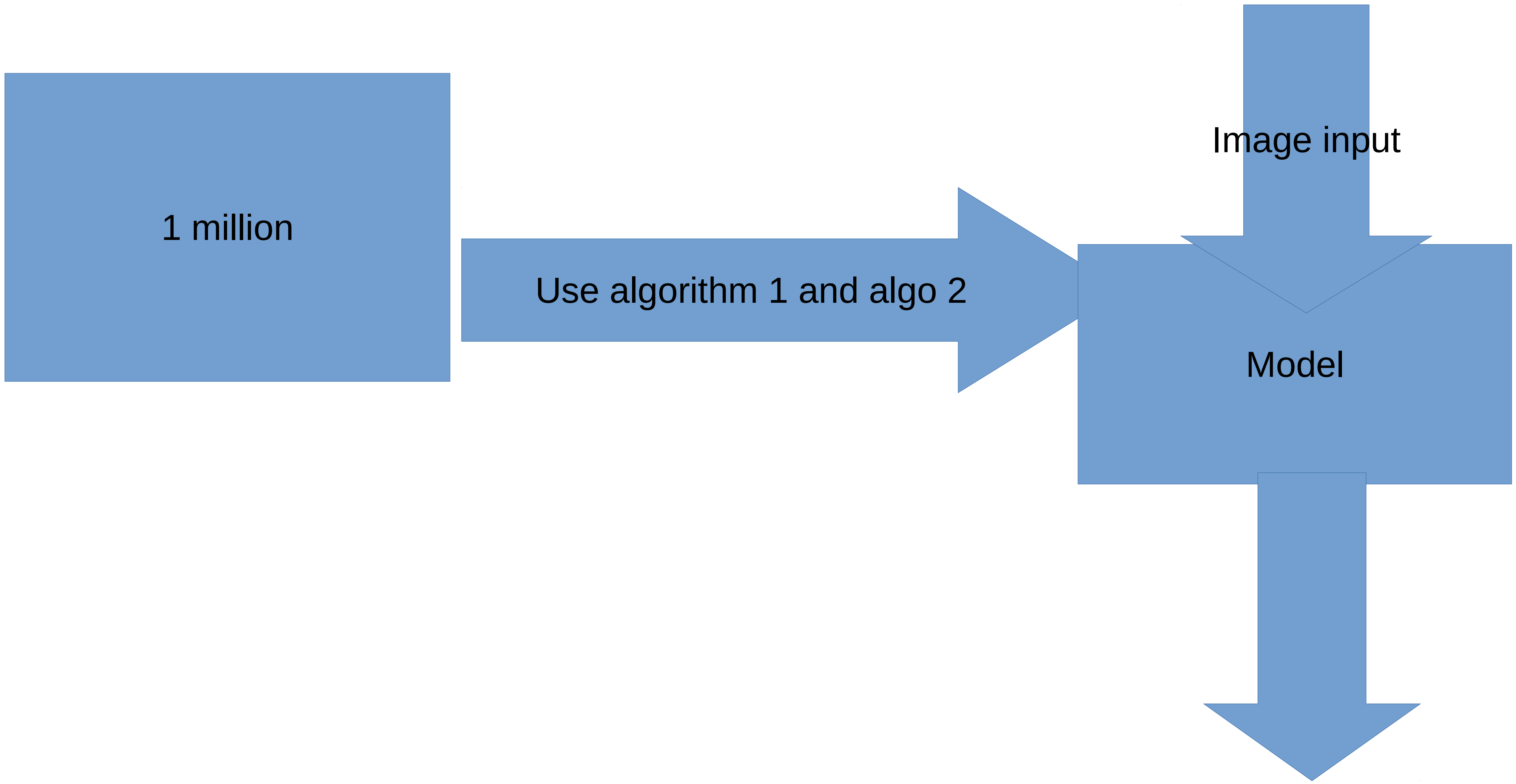
What?

SMAC – Social Mobility Analytics Cloud


proc
mem
mem
HD

IaaS



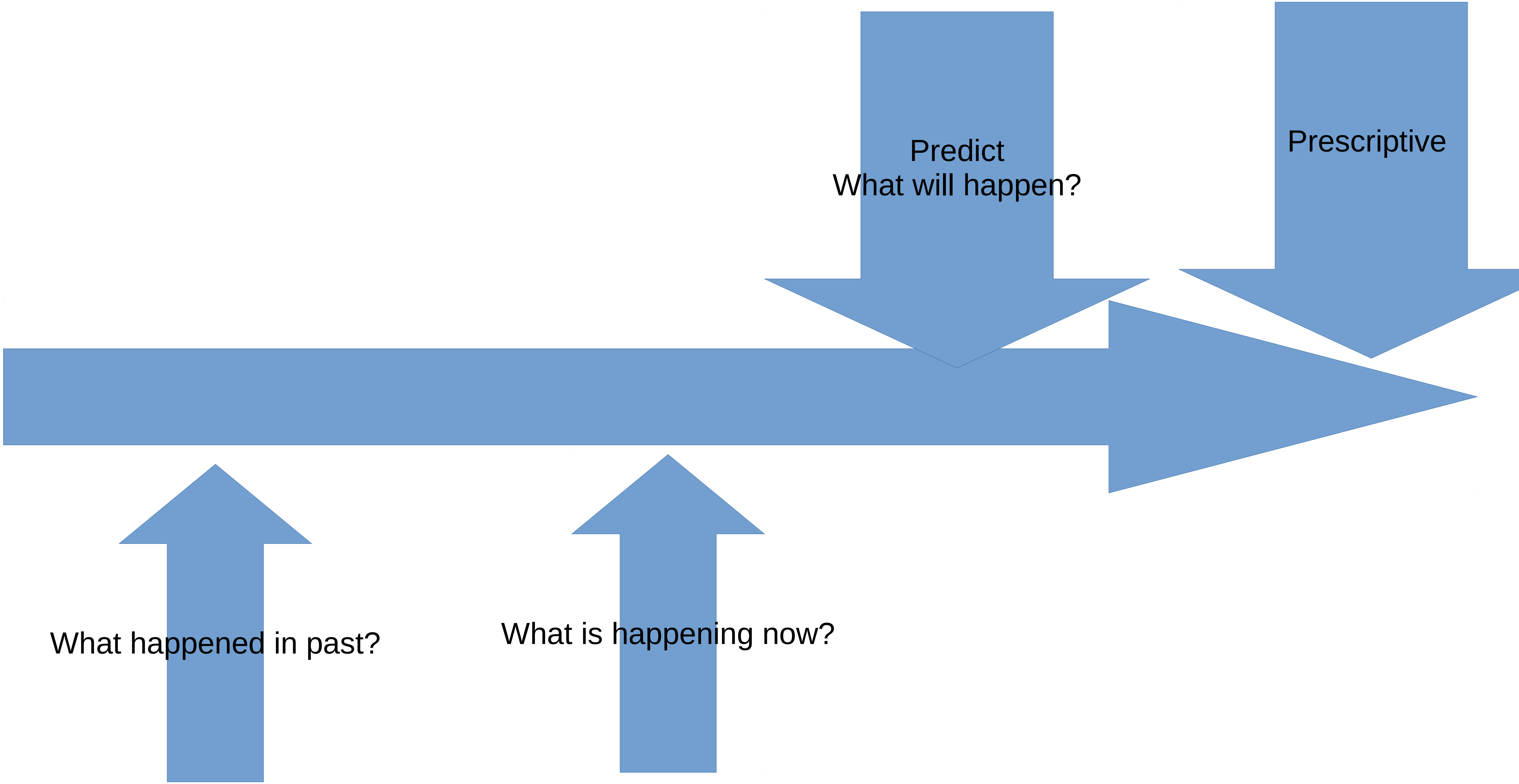


1 million

Use algorithm 1 and algo 2

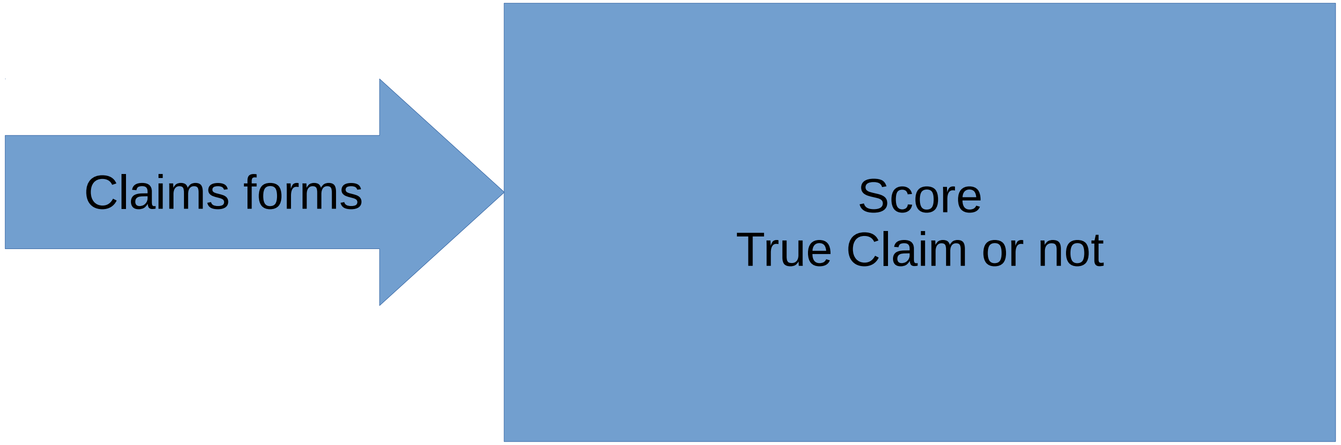
Image input

Model


Predict
What will happen?
Prescriptive
What happened in past?
What is happening now?

7” - $13.65
100” - $345

6” - $12
8” - $15.5
10” - $19
12” - $21 - $23



Claims forms

Score
True Claim or not

Id – Score
1 – 80%
2 – 50%
3 – 70%
4 – 20%

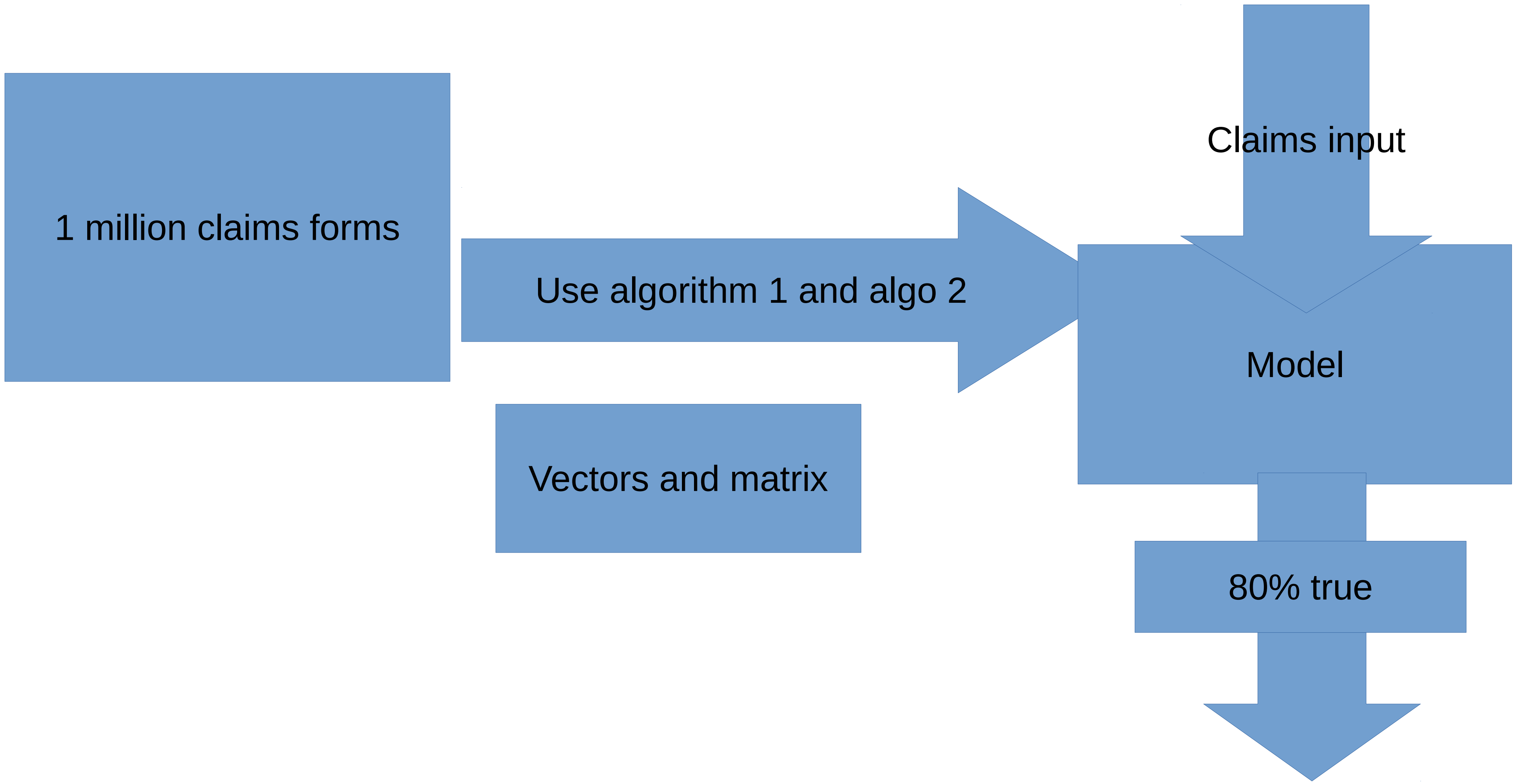
Any claim > 70% being true – Go for only theoretical verification.

Claims forms

180 Param - True / False

1 million claim forms

Historical data



1 million claims forms

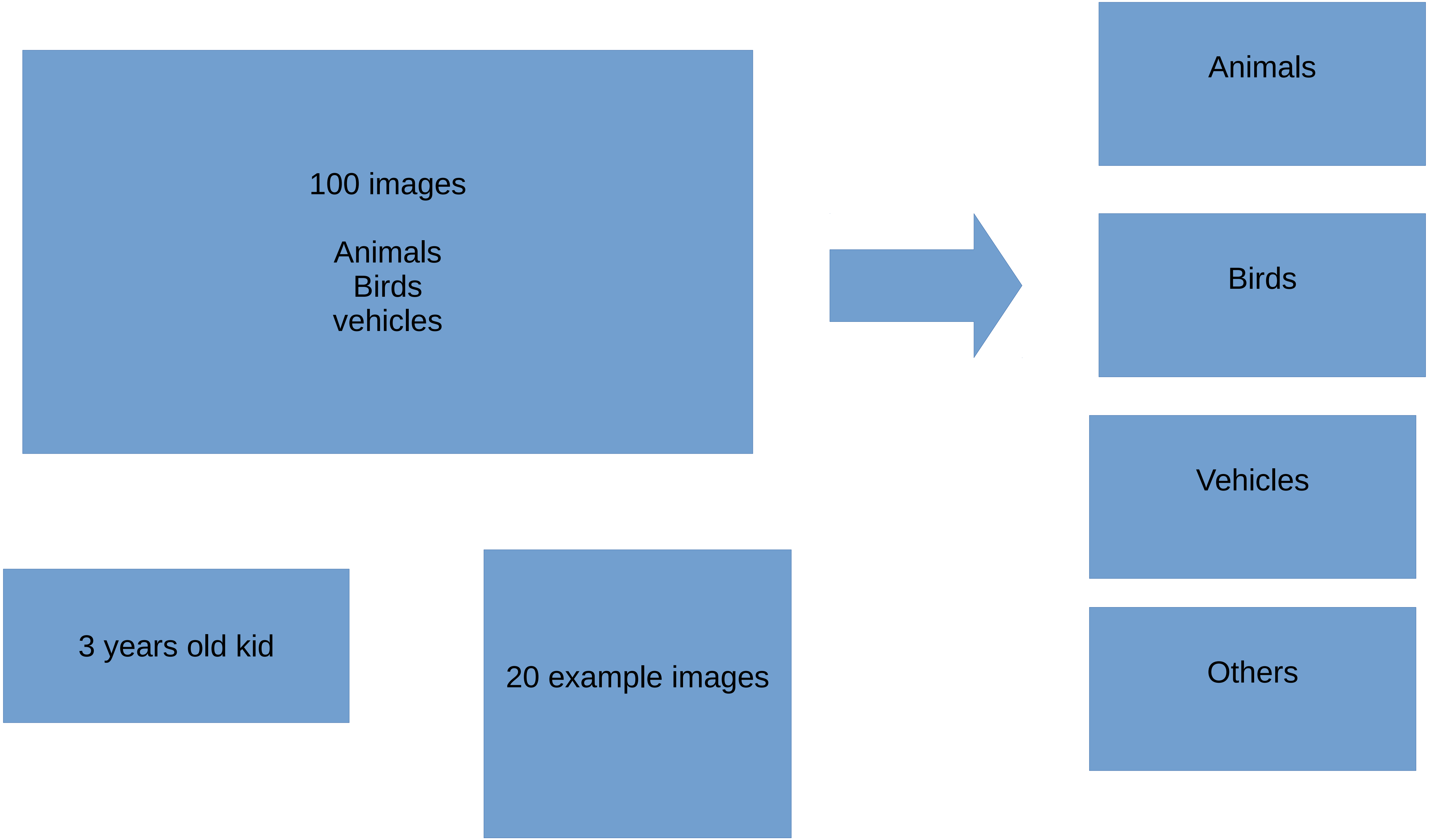
Use algorithm 1 and algo 2

Vectors and matrix

Claims input

Model

80% true



100 images

Animals
Birds
vehicles

Animals

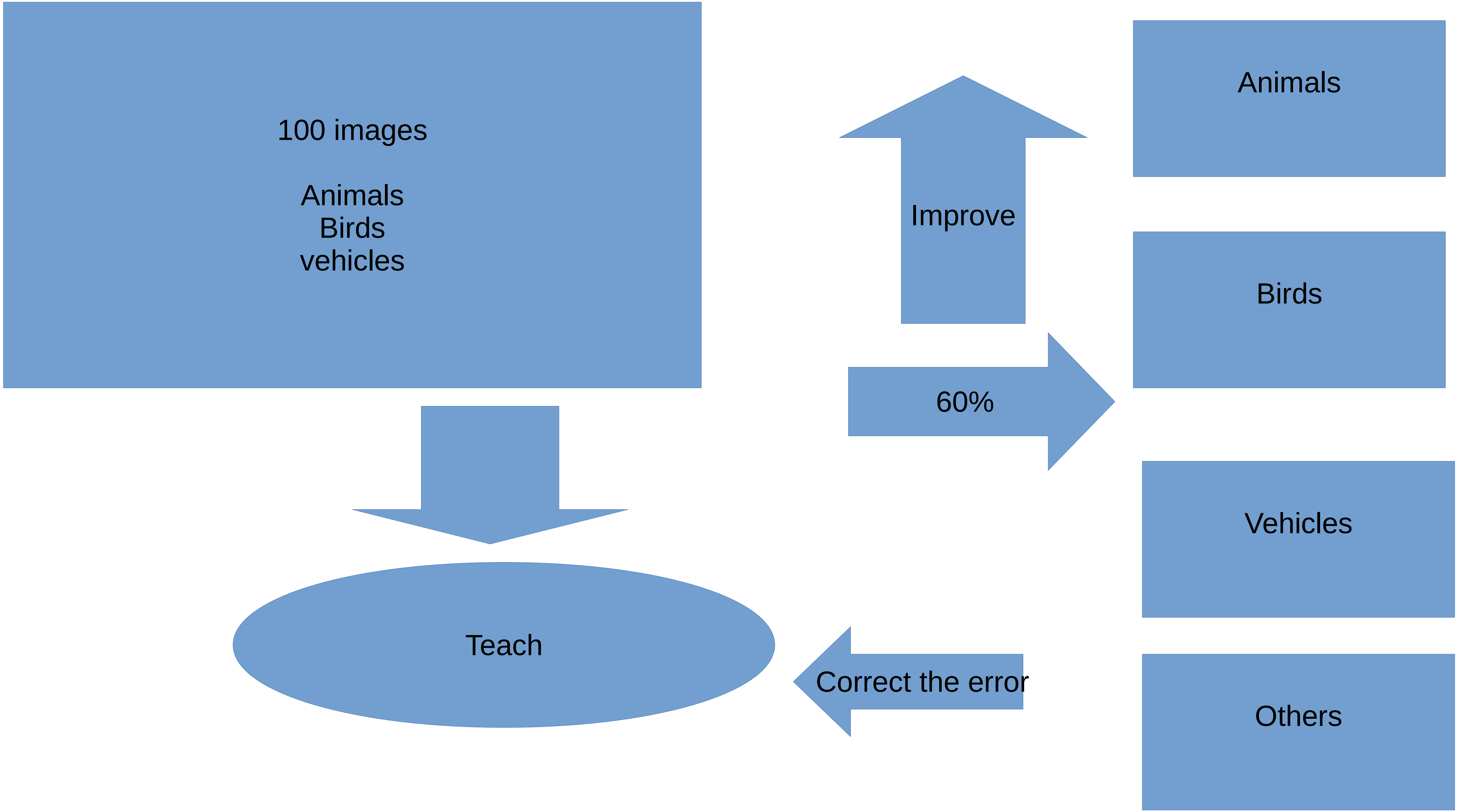
Birds

Vehicles

Others

3 years old kid

20 example images



100 images

Animals
Birds
vehicles

Teach

Correct the error

60%

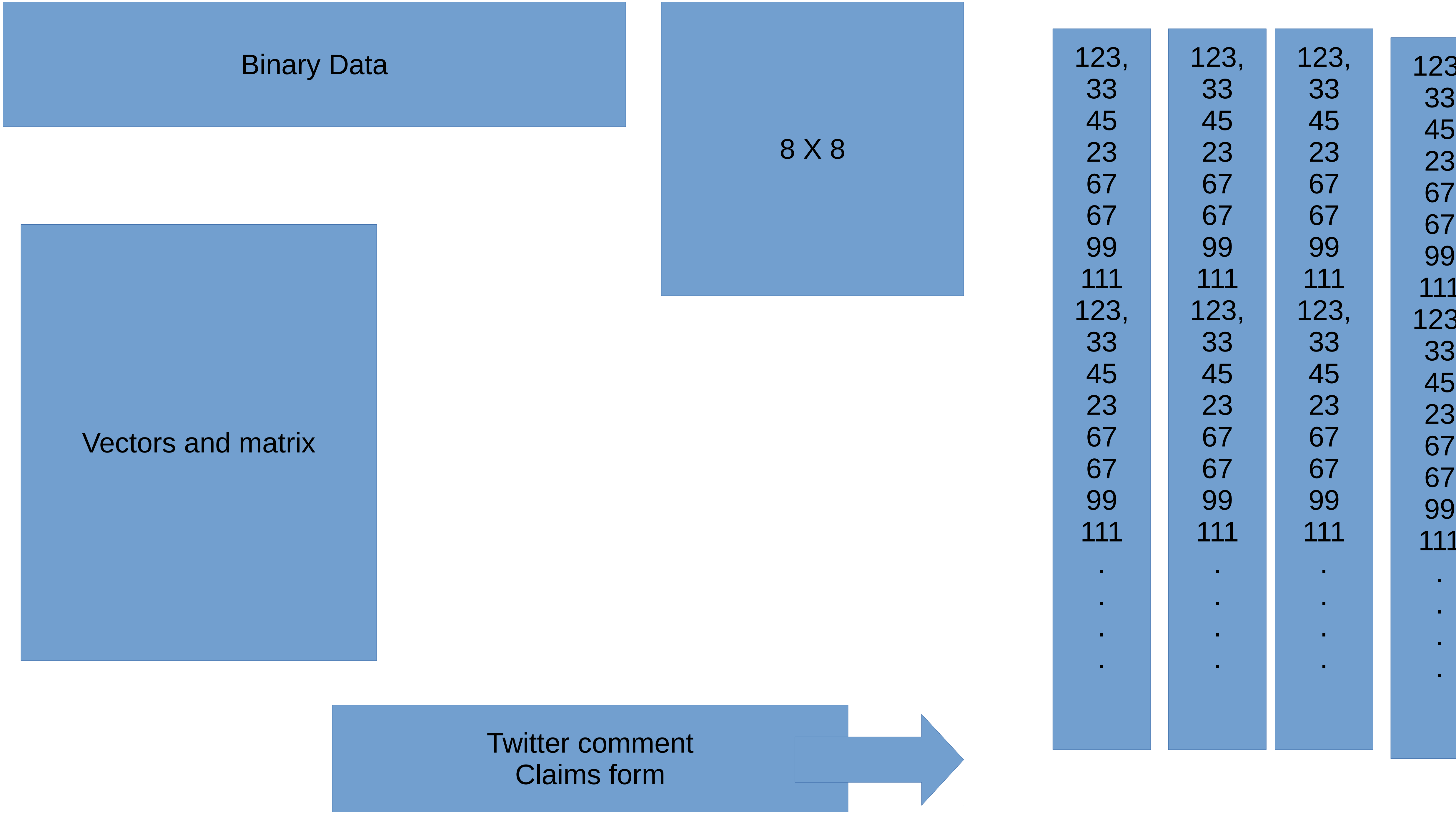
Improve

Animals

Birds

Vehicles

Others



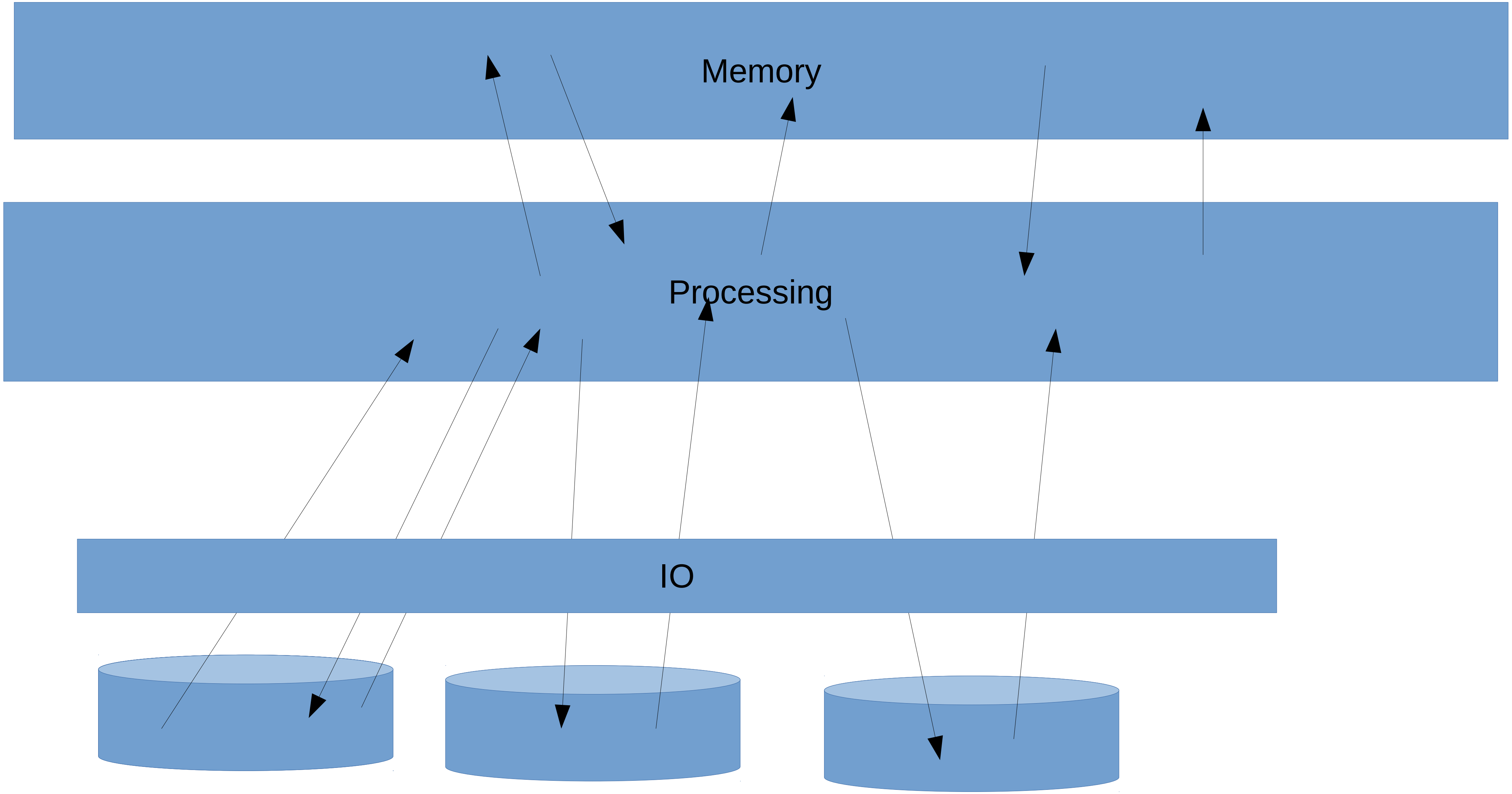
Binary Data

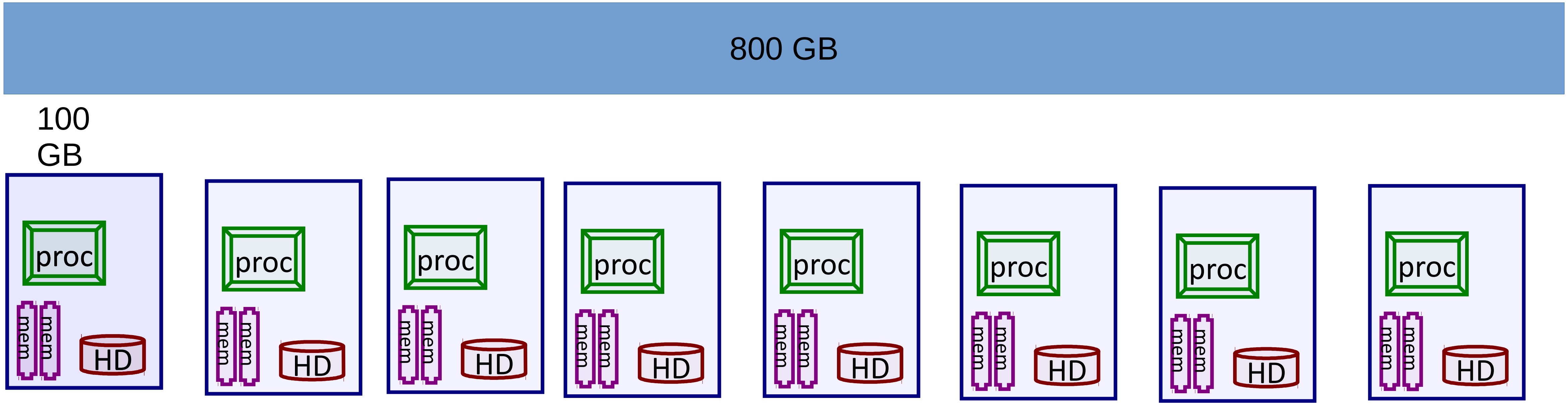
8 X 8

Vectors and matrix

Twitter comment
Claims form

123,	123,	123,	123
33	33	33	33
45	45	45	45
23	23	23	23
67	67	67	67
67	67	67	67
99	99	99	99
111	111	111	111
123,	123,	123,	123
33	33	33	33
45	45	45	45
23	23	23	23
67	67	67	67
67	67	67	67
99	99	99	99
111	111	111	111
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

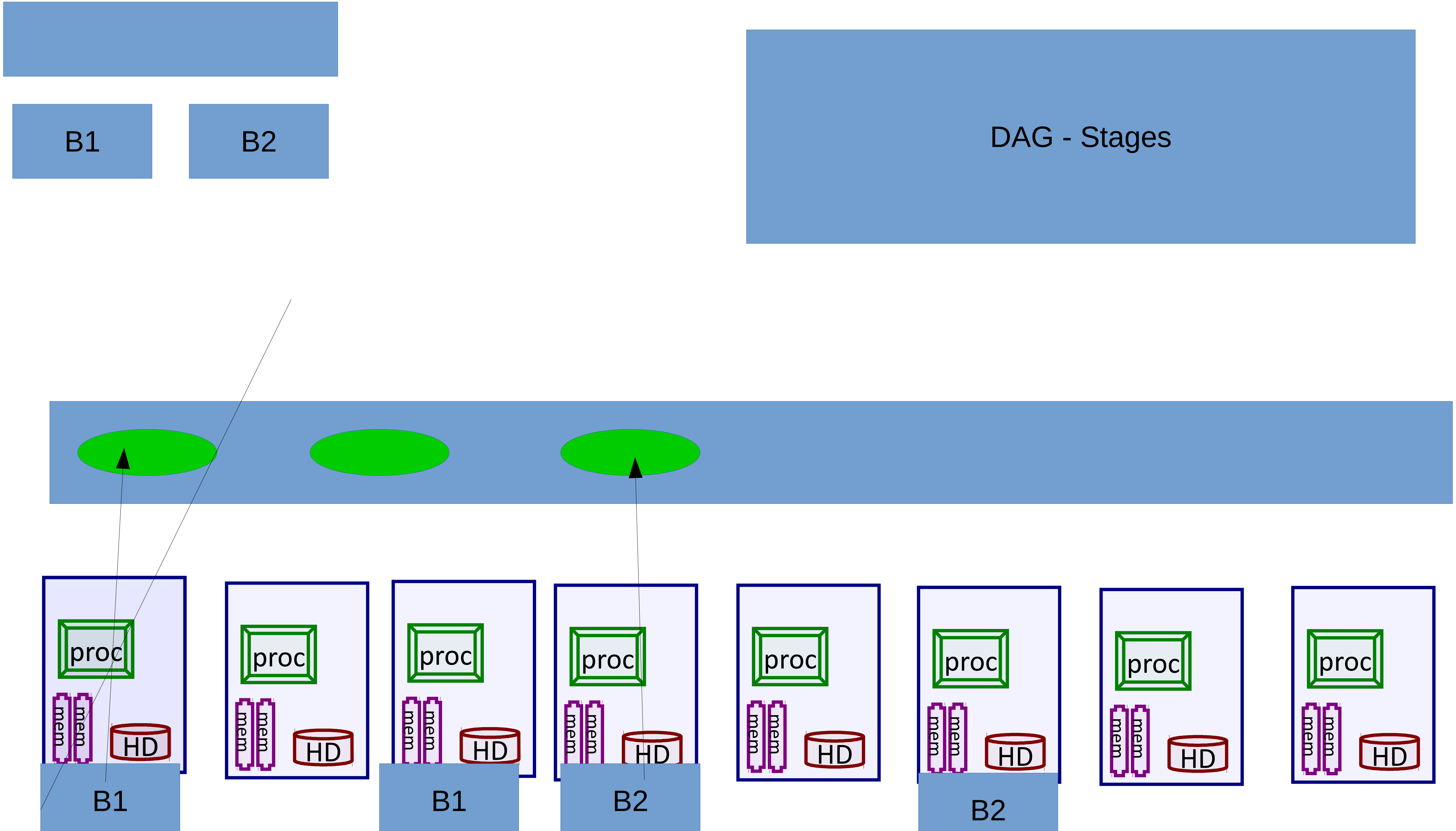

Memory
Processing
IO

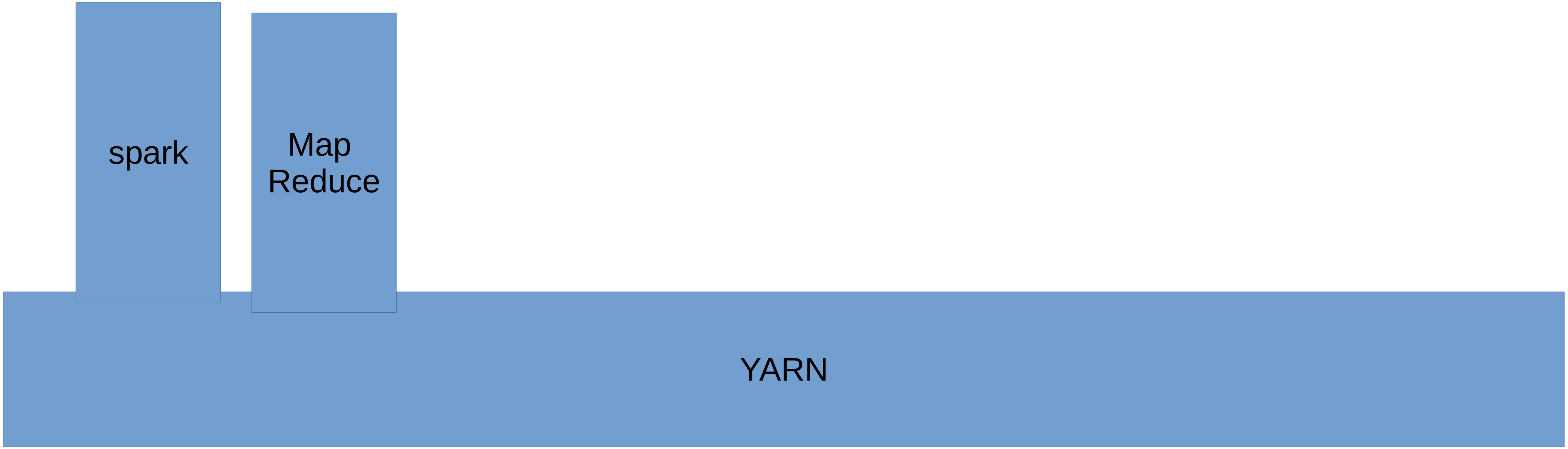


800 GB

100 GB

proc
mem
mem
HD
proc
mem
mem
HD
proc
mem
mem
HD
proc
mem
mem
HD
proc
mem
mem
HD
proc
mem
mem
HD
proc
mem
mem
HD
proc
mem
mem
HD

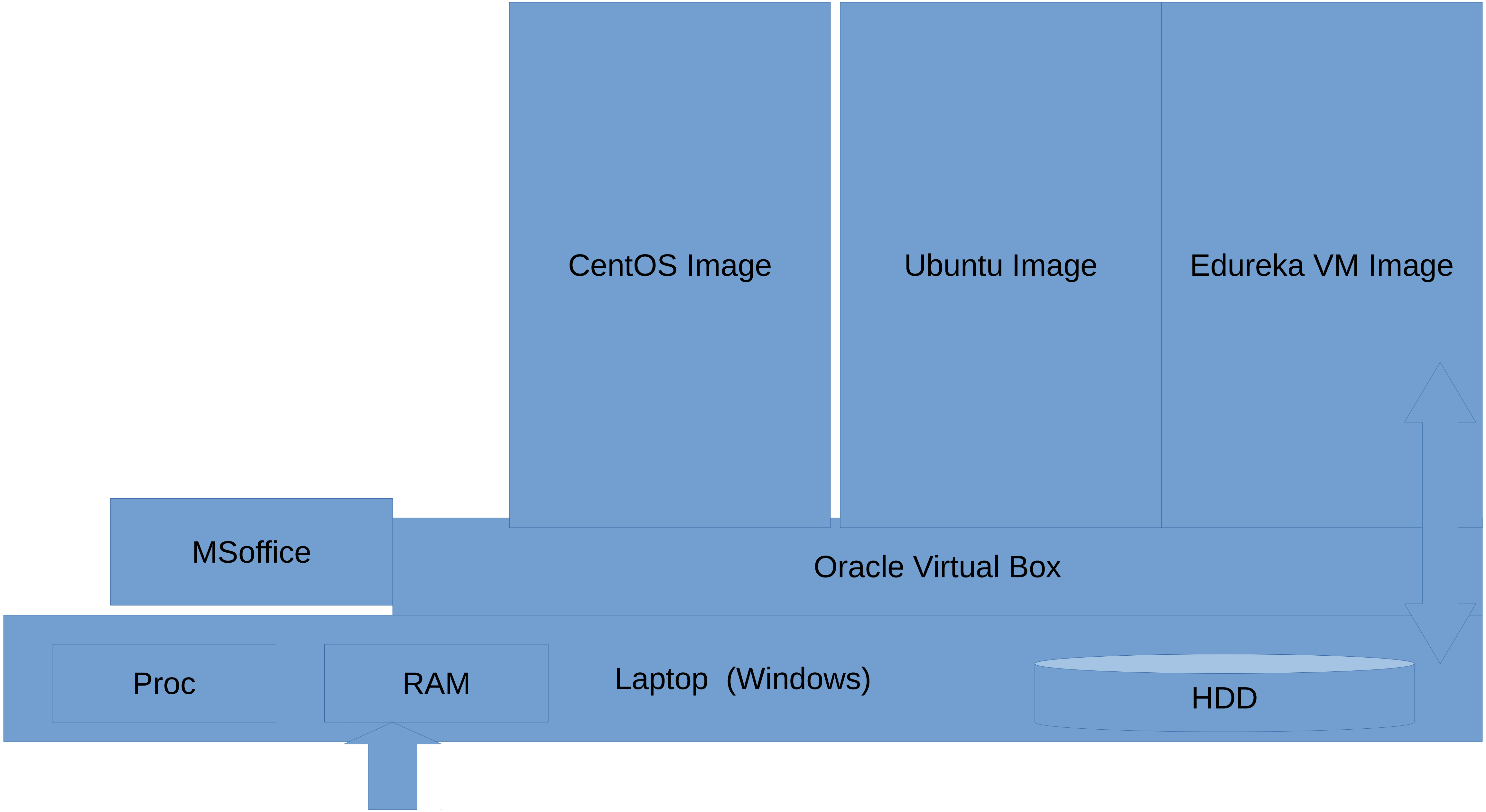

B1
B2
DAG - Stages
proc
mem
mem
HD
B1
B1
B2
B2



spark

Map
Reduce

YARN



CentOS Image

Ubuntu Image

Edureka VM Image

MSoffice

Oracle Virtual Box

Proc

RAM

Laptop (Windows)

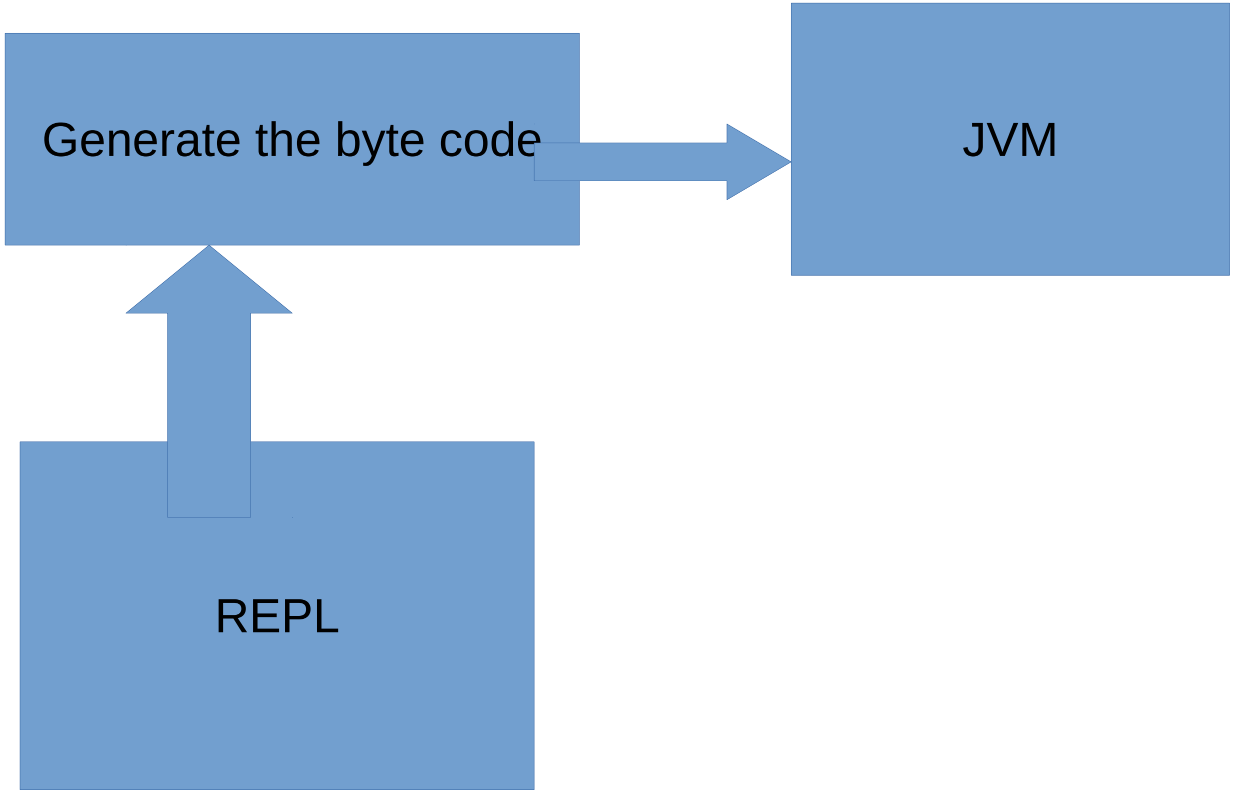
HDD

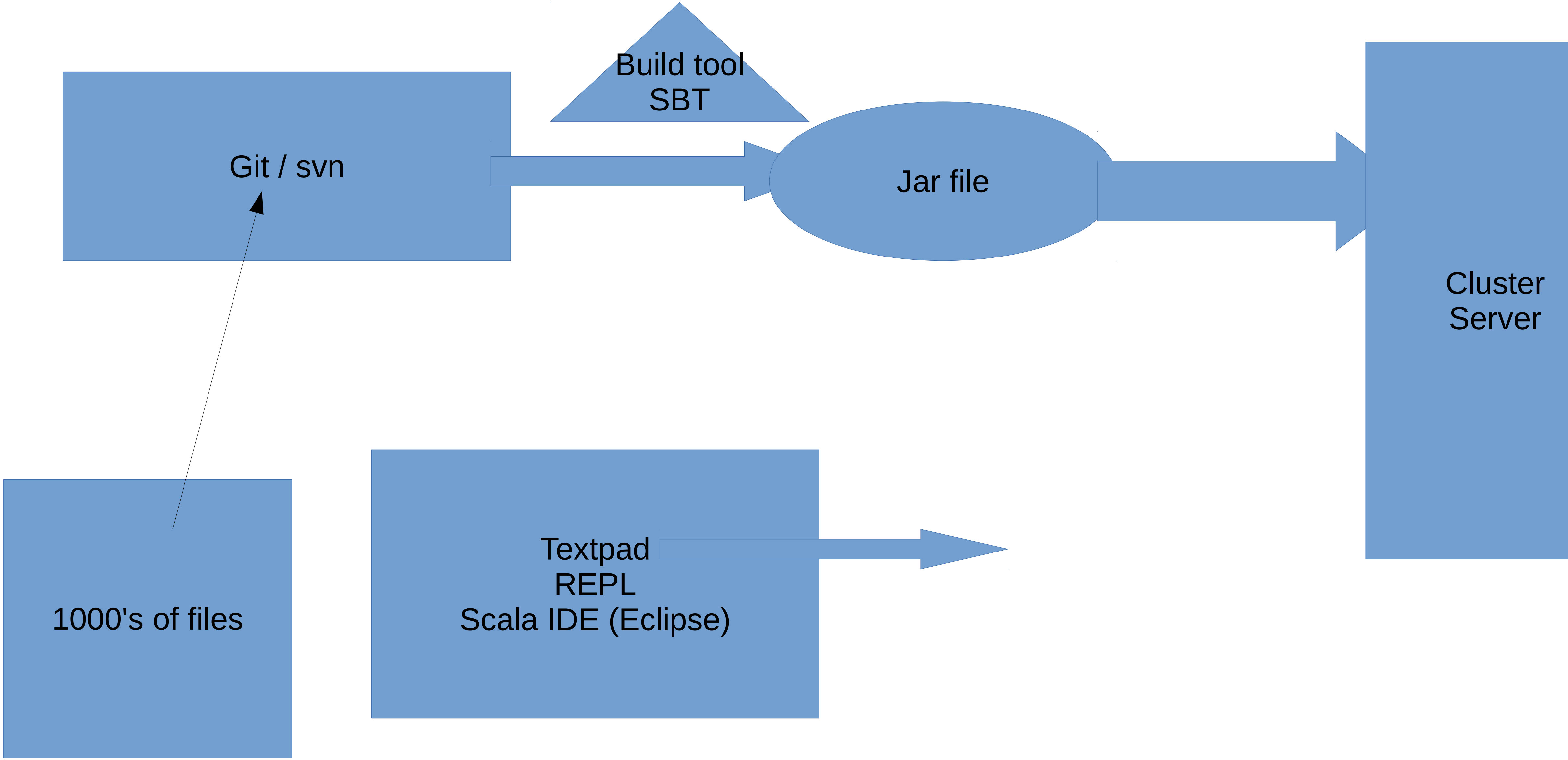
Object{

int = 5

Assign a function to a Variable
Pass a function as an argument to another function
Get function as return type form a function.

int and Integer


Generate the byte code
JVM
REPL



Git / svn

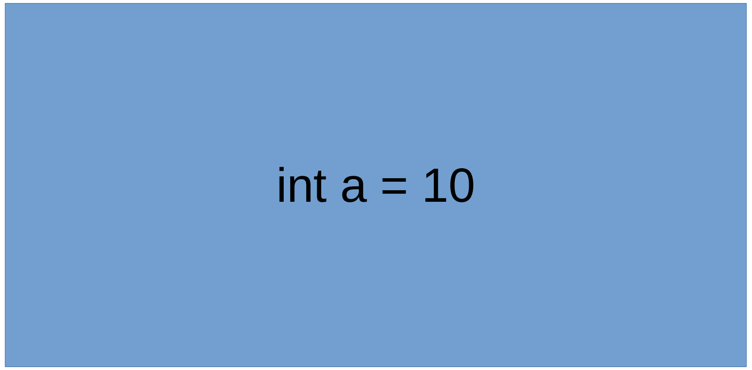
Build tool
SBT

Jar file

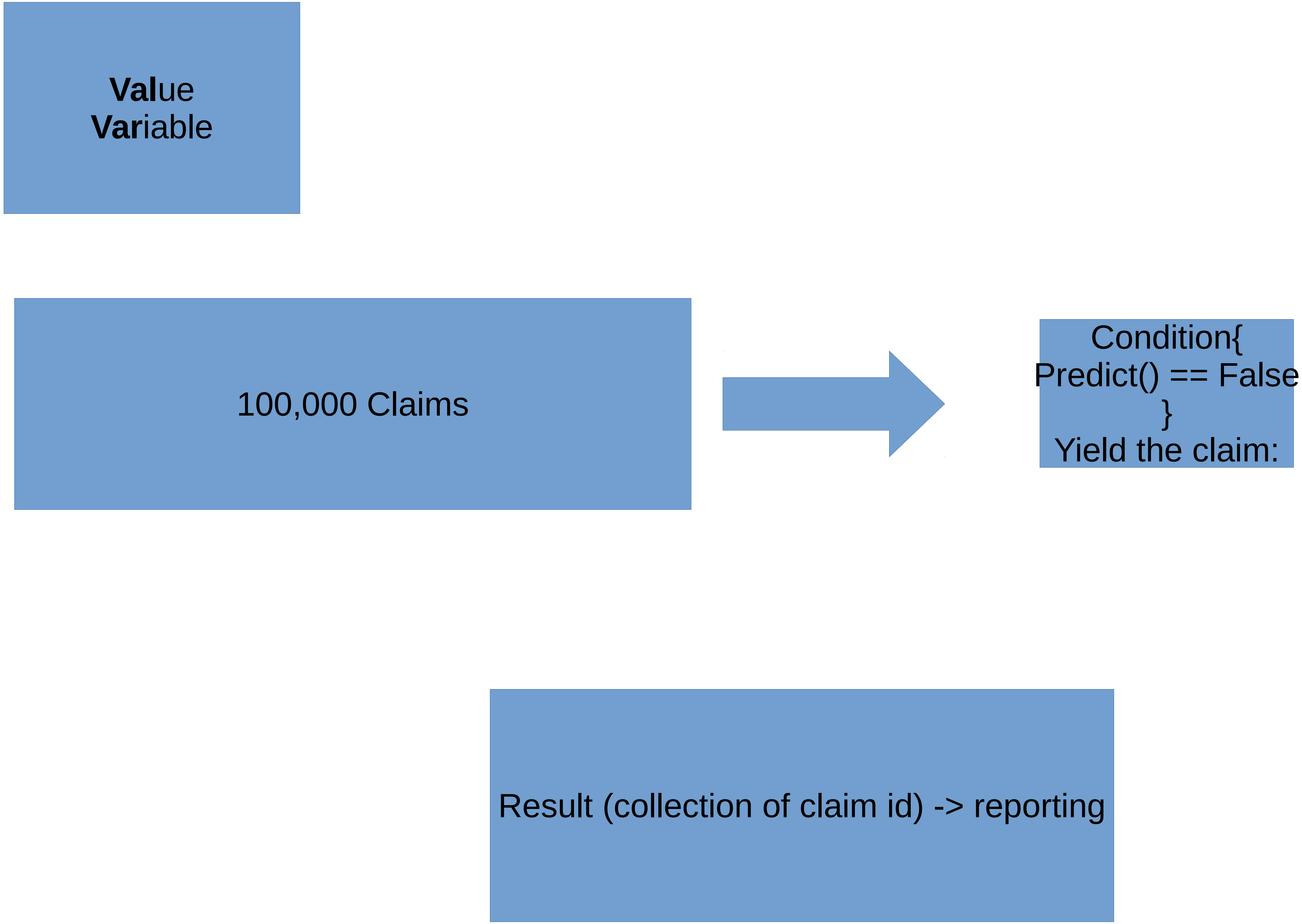
Cluster
Server

1000's of files

Textpad
REPL
Scala IDE (Eclipse)



int a = 10



Value
Variable

100,000 Claims

Condition{
Predict() == False
}
Yield the claim:

Result (collection of claim id) -> reporting

Compile and package it.

IDE

Some other Editor -> cmd -> execute

REPL

10 reference files

1) if the type is Person = read the person file
2) if the type is System = read the system file
2) if the type is Location = read the Location file

```
Account()

Var balance

Deposit(amount){
Balance = balance + amount}

withdraw(amount){
Balance = balance – amount}

CalcInt
possibleLoanAmount()
```

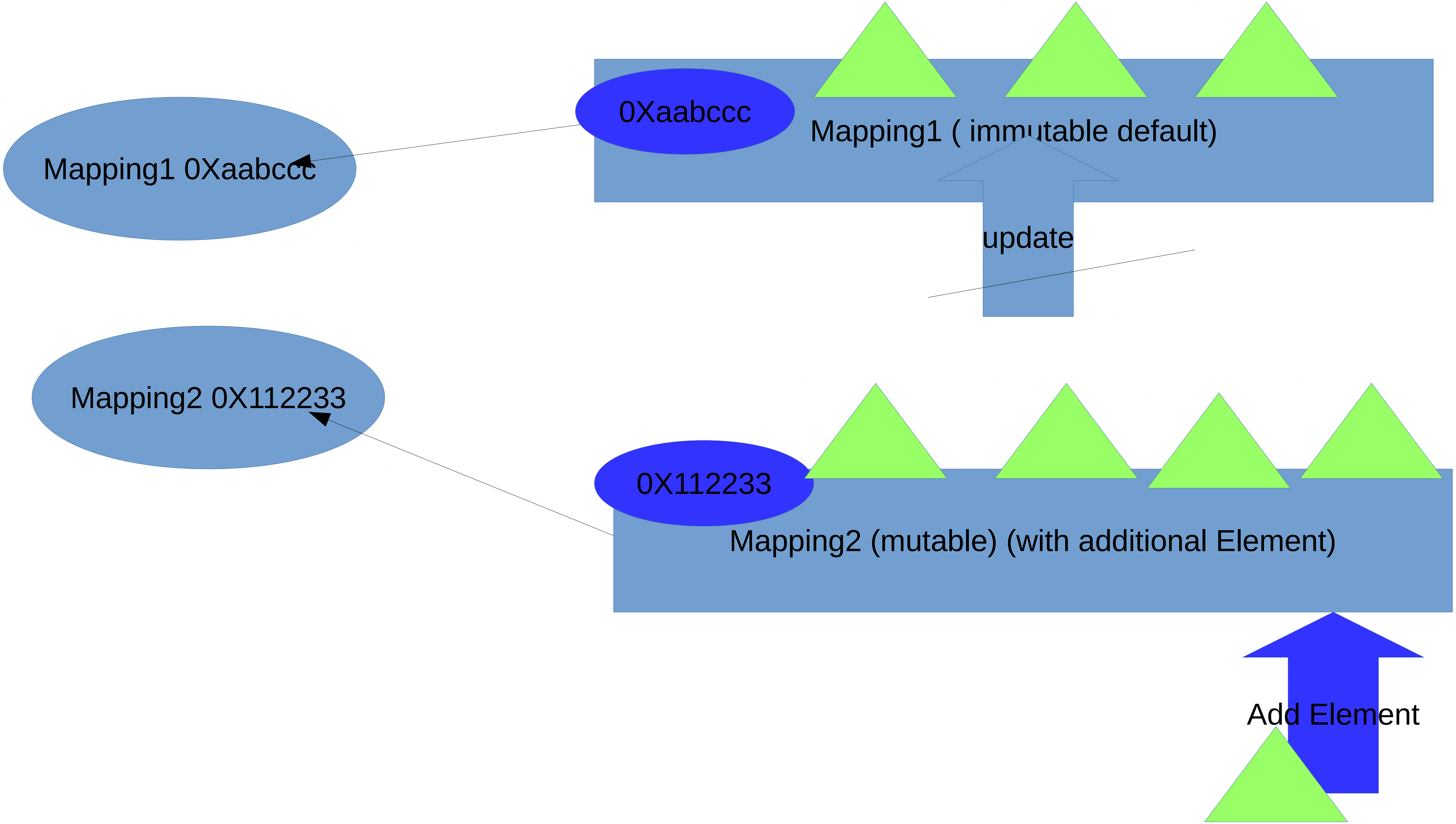
```
CalcInt($4000, 1year, 2%)
```

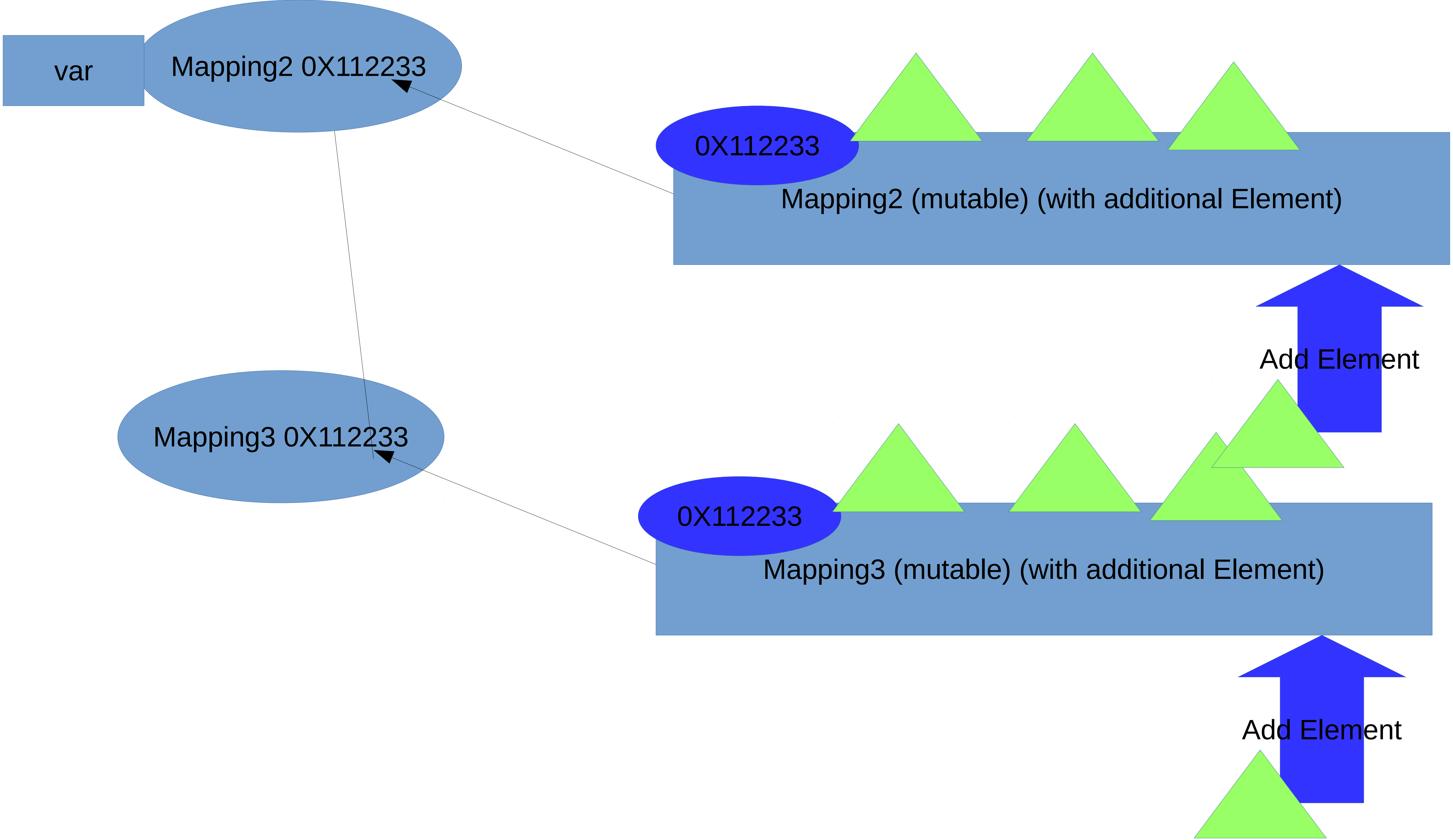
```
Account object = accobj (NarendraID)

Accobj.deposit($1000)
```

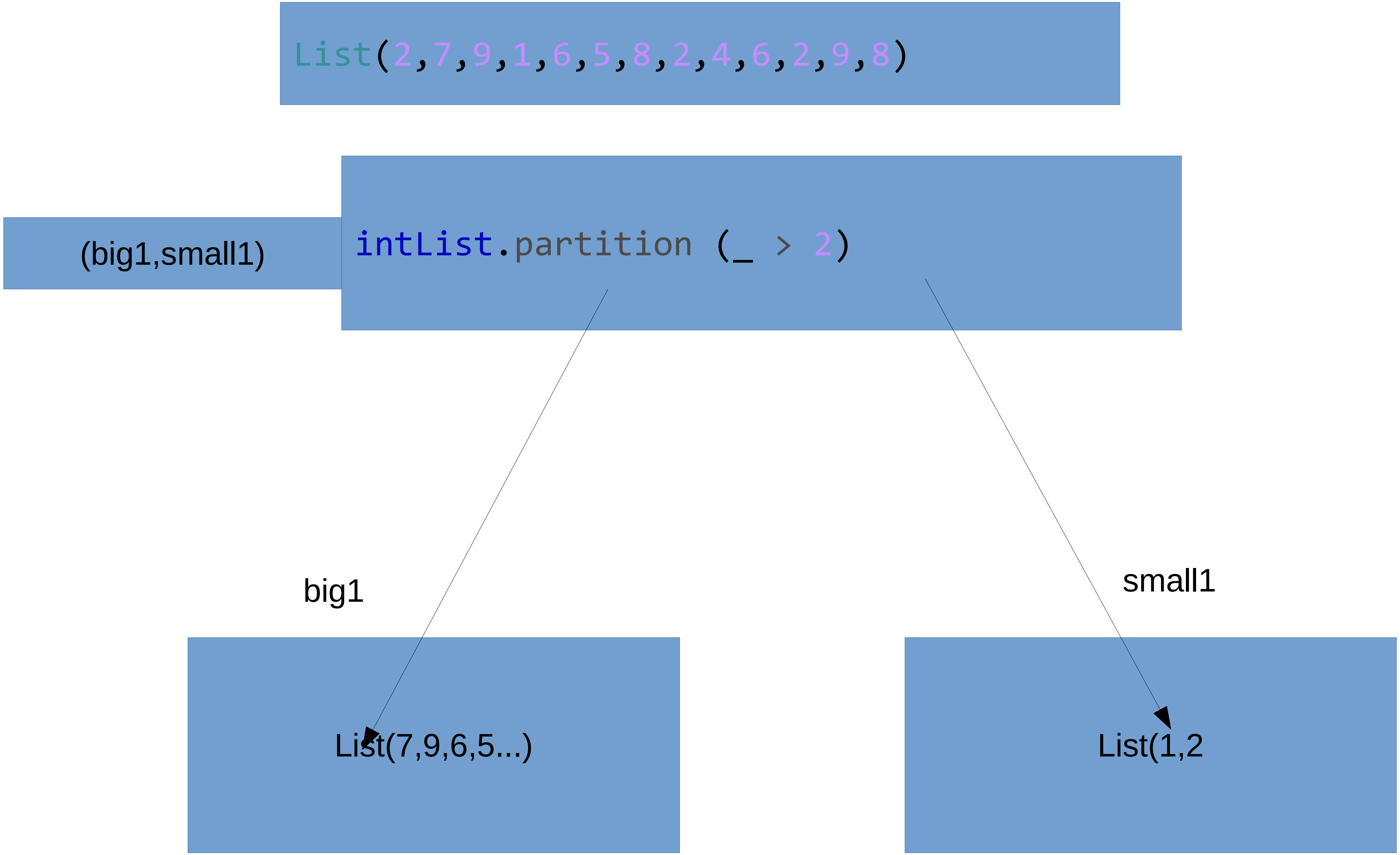
CalculateCreditScore(180 arguments)

OOPs

key(claimID) -> value(Claimobject)


0Xaabccc
Mapping1 (immutable default)
Mapping1 0Xaabccc
update
Mapping2 0X112233
0X112233
Mapping2 (mutable) (with additional Element)
Add Element


var
Mapping2 0X112233
0X112233
Mapping2 (mutable) (with additional Element)
Add Element
Mapping3 0X112233
0X112233
Mapping3 (mutable) (with additional Element)
Add Element



```
List(2,7,9,1,6,5,8,2,4,6,2,9,8)
```

(big1,small1)

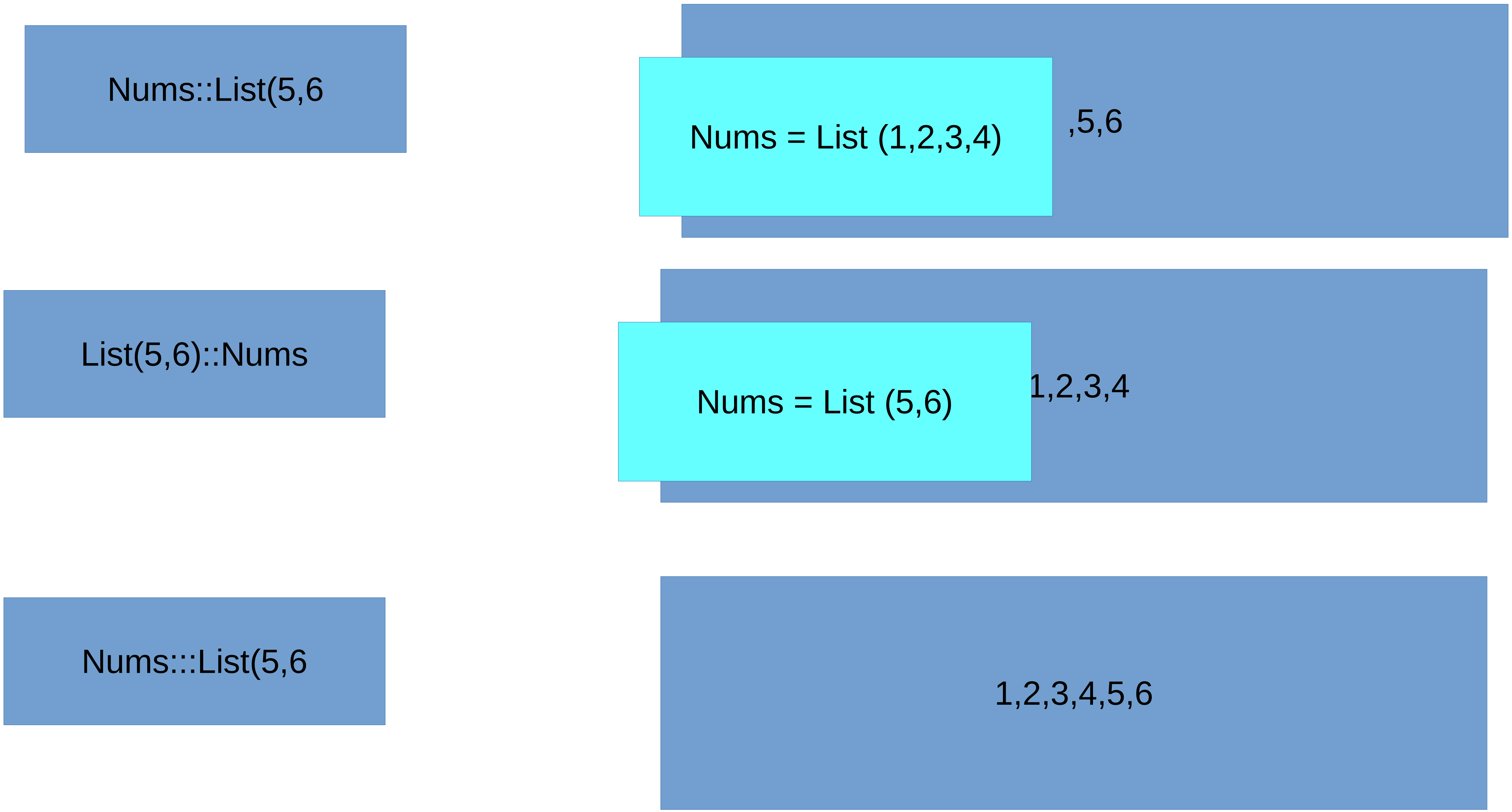
```
intList.partition (_ > 2)
```

big1

small1

List(7,9,6,5...)

List(1,2



Nums::List(5,6

Nums = List (1,2,3,4)

,5,6

List(5,6)::Nums

Nums = List (5,6)

1,2,3,4

Nums:::List(5,6

1,2,3,4,5,6

List of 100 Claims
List[Claims]

Get all claim amt > $100

Highclaimamts = For loop{
Claim.amt > $100 } yield

Class

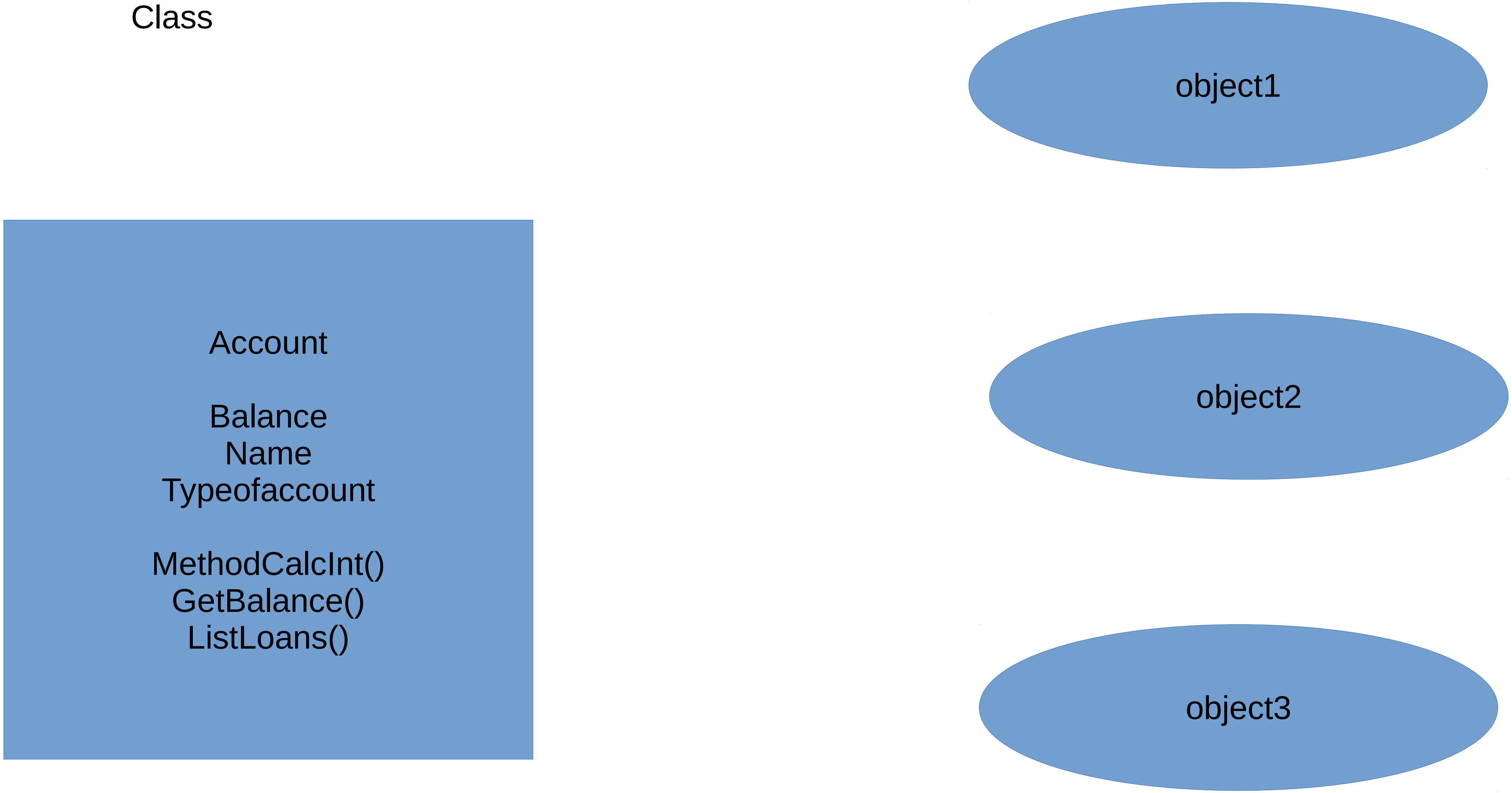
Class

object1

object2

object3

Class



Account

Balance
Name
Typeofaccount

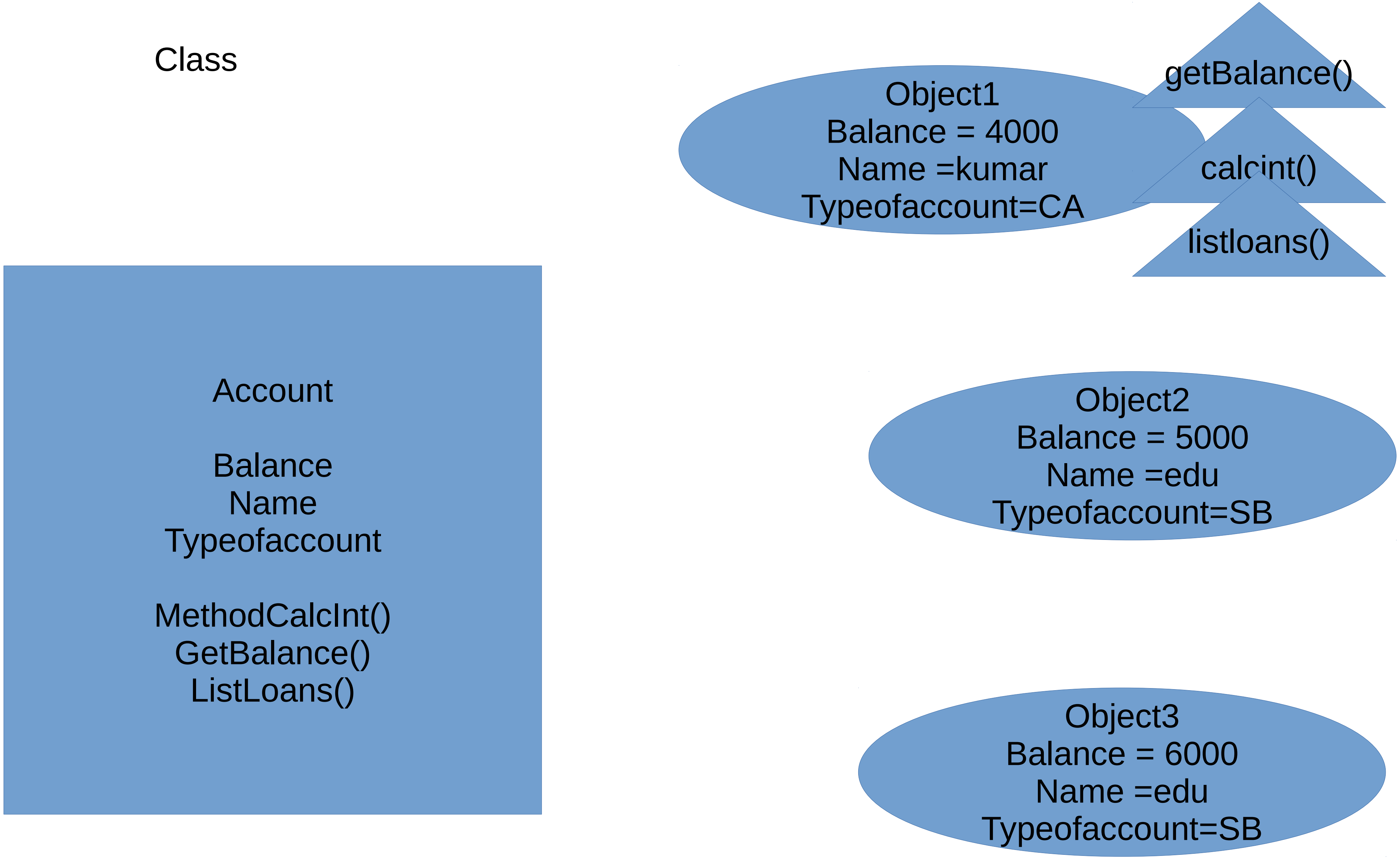
MethodCalcInt()
GetBalance()
ListLoans()

object1

object2

object3

Class



Account

Balance
Name
Typeofaccount

MethodCalcInt()
GetBalance()
ListLoans()

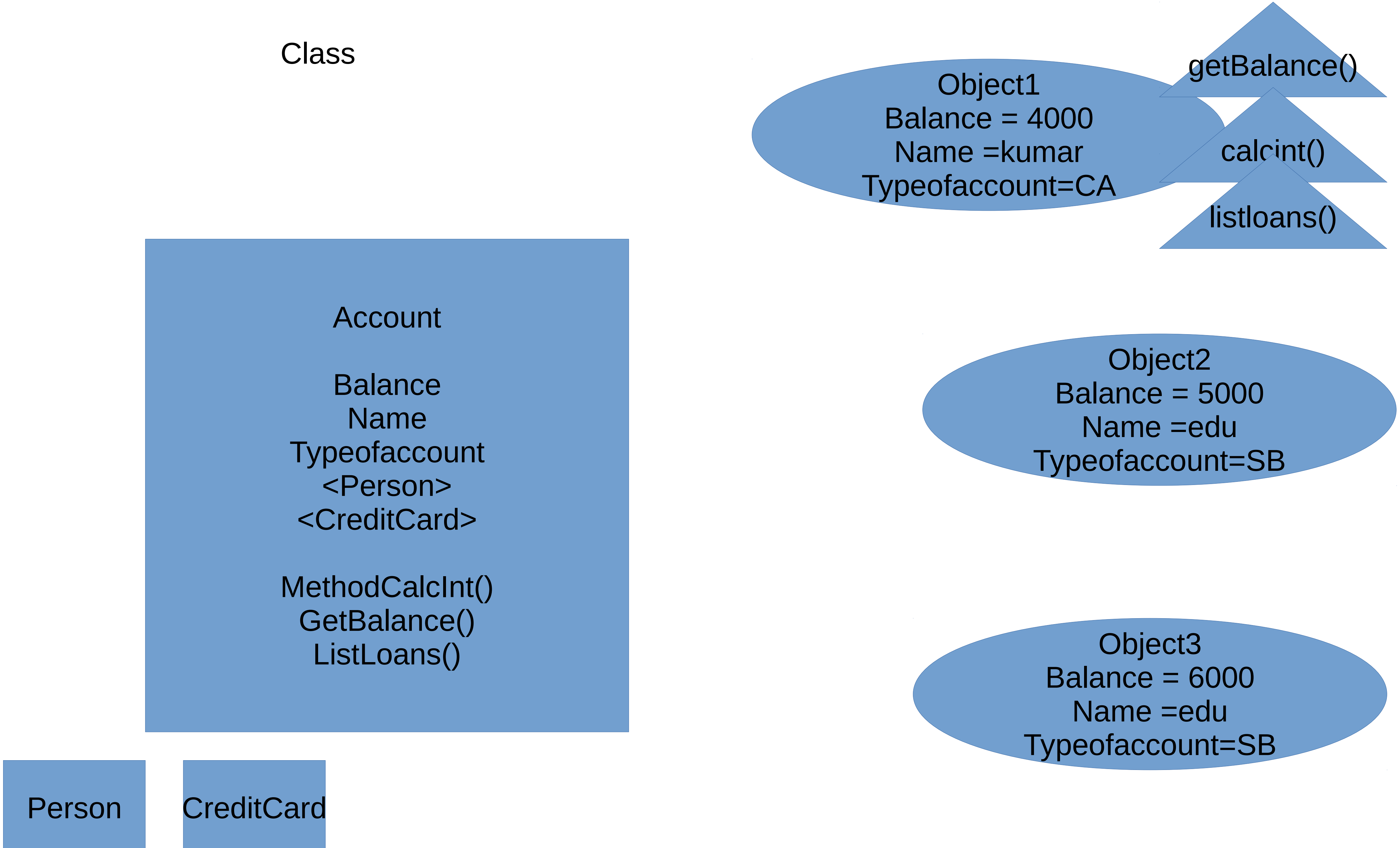
Object1
Balance = 4000
Name =kumar
Typeofaccount=CA

getBalance()
calcint()
listloans()

Object2
Balance = 5000
Name =edu
Typeofaccount=SB

Object3
Balance = 6000
Name =edu
Typeofaccount=SB

Class



Account

Balance
Name
Typeofaccount
<Person>
<CreditCard>

MethodCalcInt()
GetBalance()
ListLoans()

Person

CreditCard

Object1
Balance = 4000
Name =kumar
Typeofaccount=CA

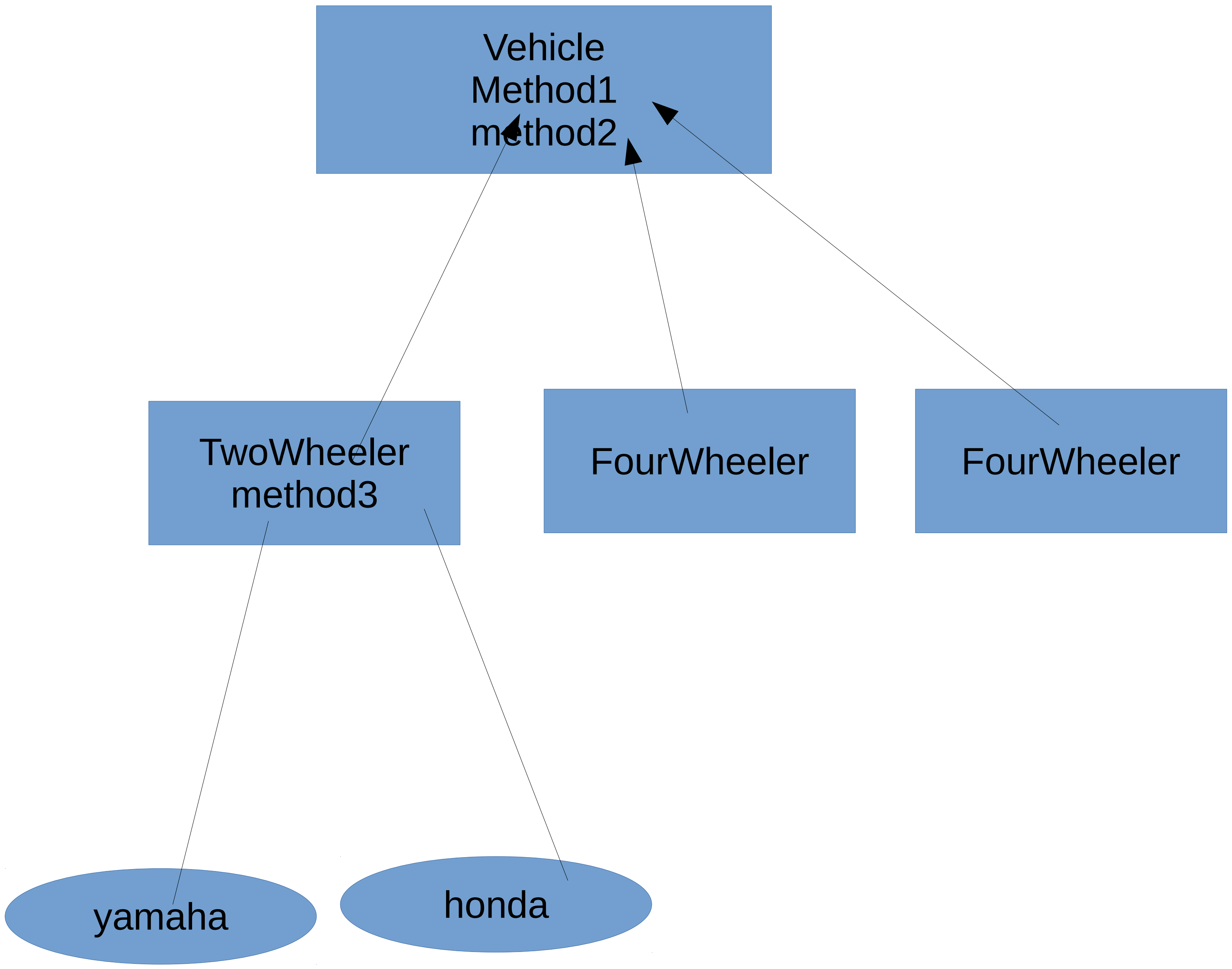
getBalance()

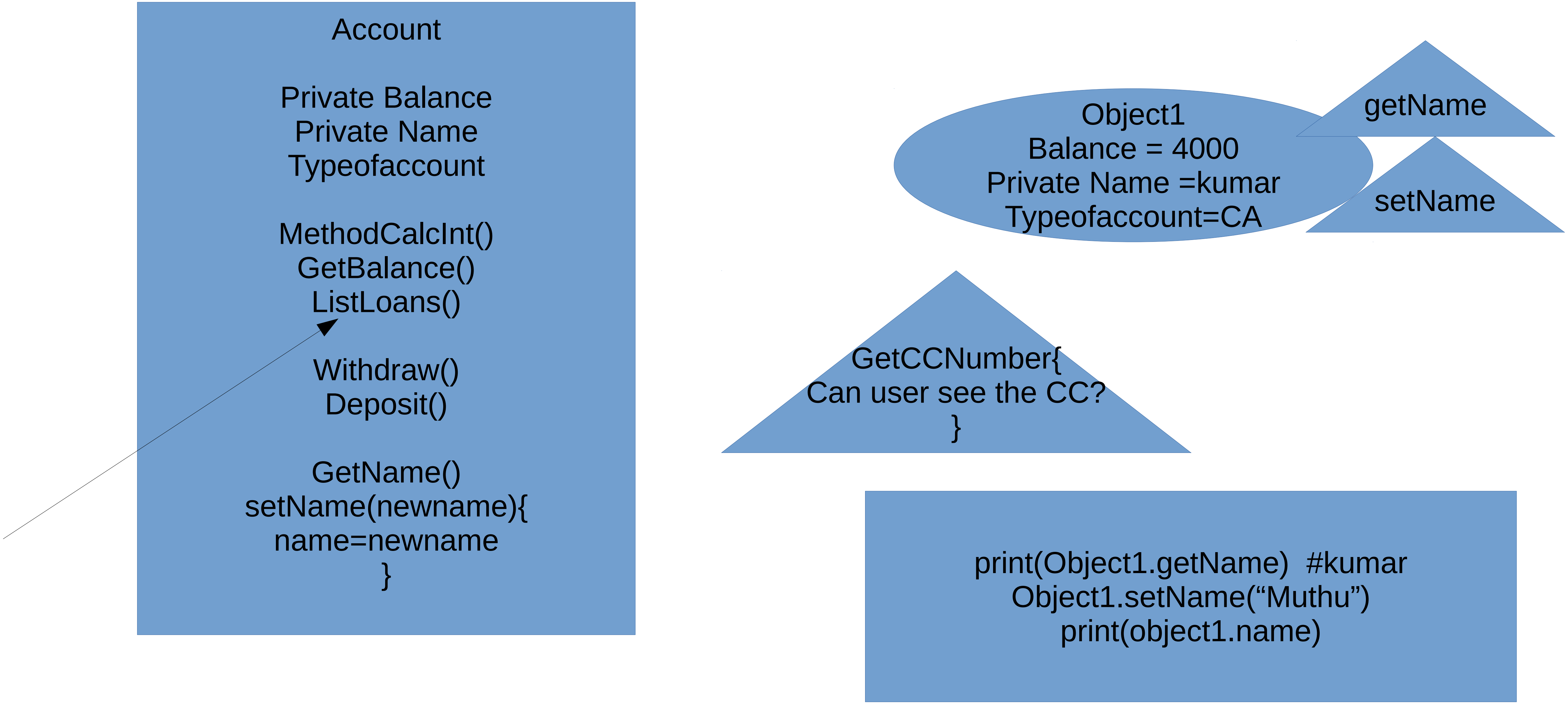
calcint()

listloans()

Object2
Balance = 5000
Name =edu
Typeofaccount=SB

Object3
Balance = 6000
Name =edu
Typeofaccount=SB


Vehicle
Method1
method2
TwoWheeler
method3
FourWheeler
FourWheeler
yamaha
honda



Account

Private Balance
Private Name
Typeofaccount

MethodCalcInt()
GetBalance()
ListLoans()

Withdraw()
Deposit()

GetName()
setName(newname){
name=newname
}

Object1
Balance = 4000
Private Name =kumar
Typeofaccount=CA

getName

setName

GetCCNumber{
Can user see the CC?
}

```
print(Object1.getName)  #kumar
Object1.setName(“Muthu”)
print(object1.name)
```

Default var - > make it as private, generate public getter and setter

Private var -> var will be private, generate private getter and setter

Type val -> make the val as private and generate only getter

Private this - > variable will be private, no getter no setter isSSNequal

Person1
SSN = 111

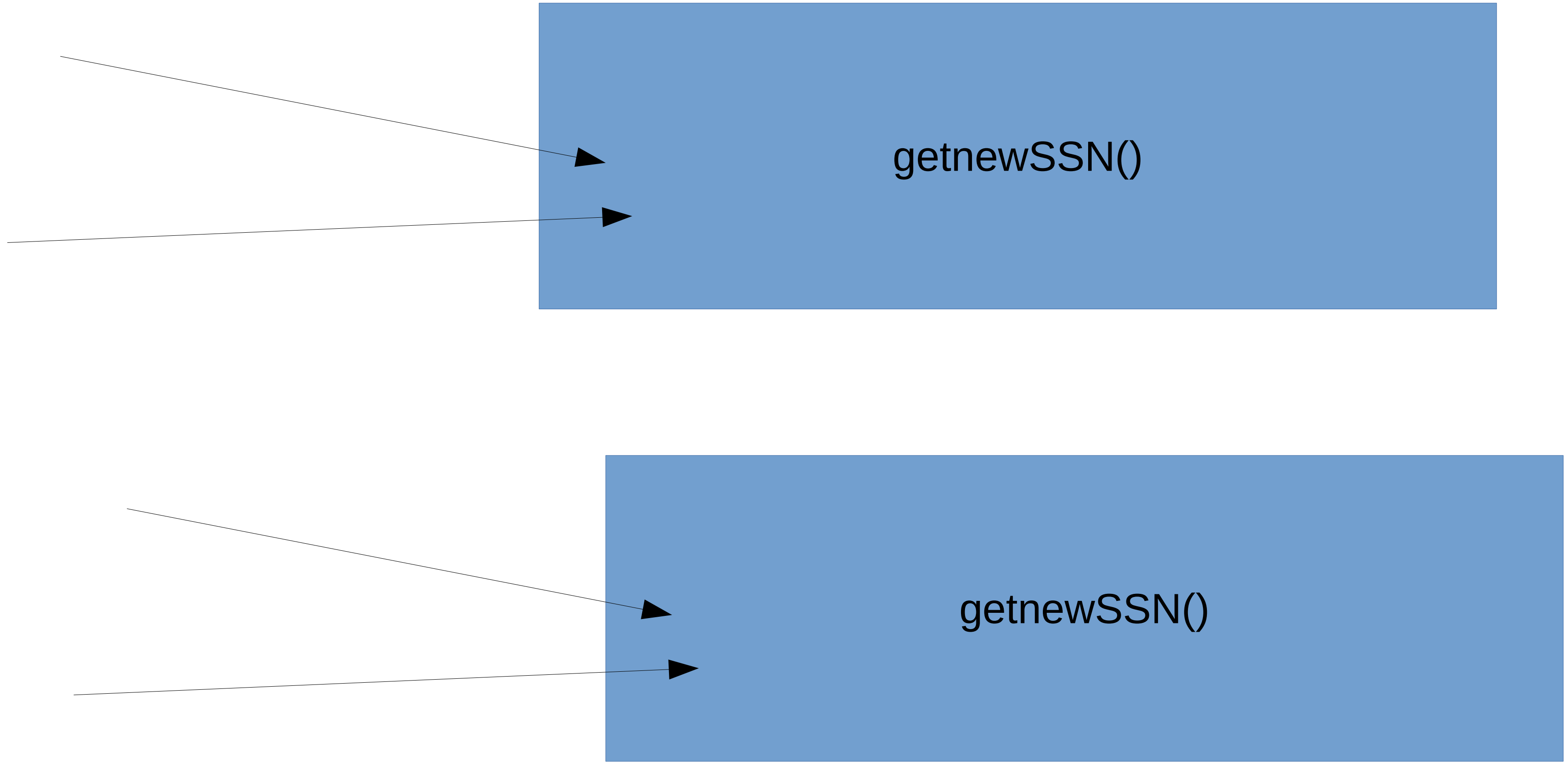
Person1.isSSNequal(person2)

Ssn == person2.ssn
Print person2.ssn

Person

```
Private SSN

isSSNequal(tocompareobj)
{
if(ssn == tocompareobj.ssn)
Print tocompareobj.ssn
Return “YES”
Else
Return “NO”

}
```

Person2
SSN = 222
incr()

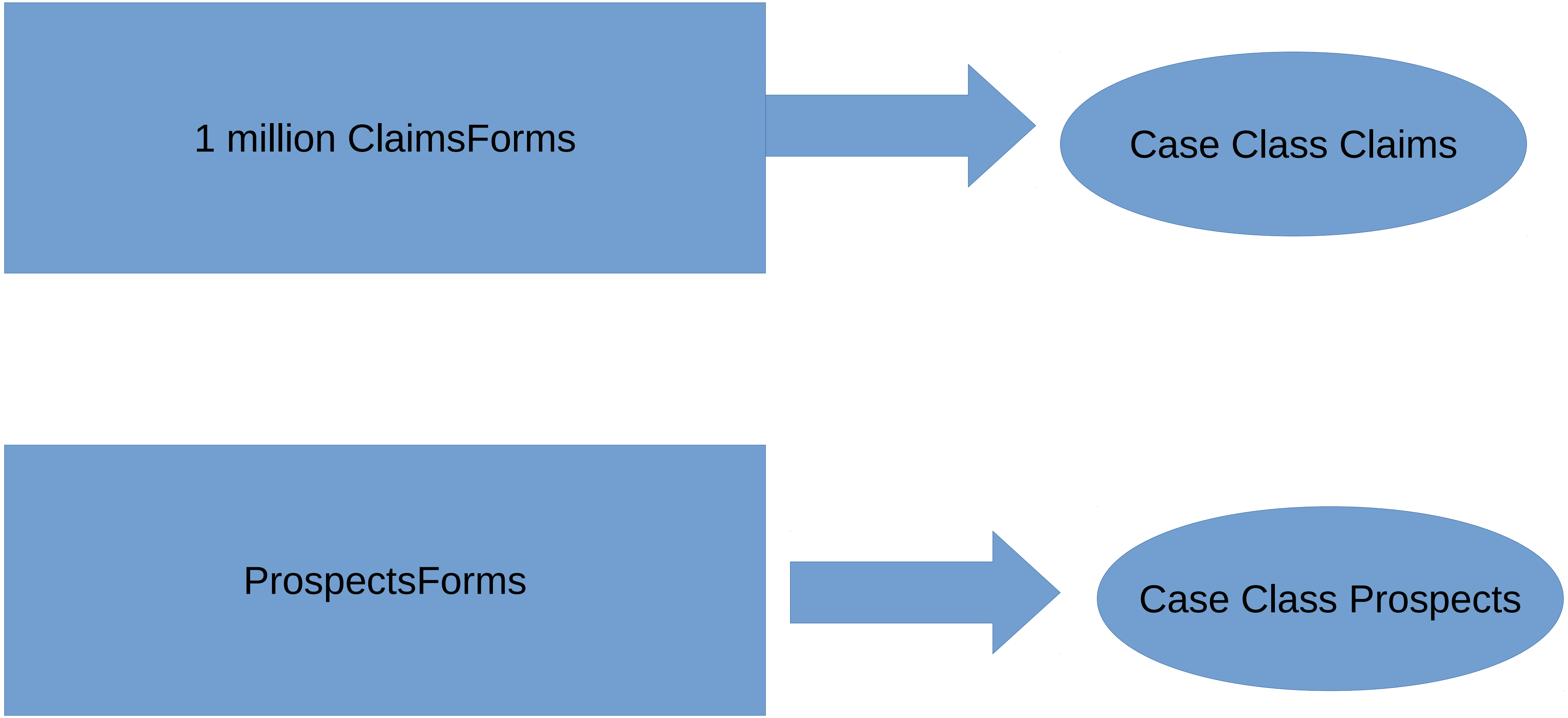


getnewSSN()

getnewSSN()

55 – Firstname
100 – Lastname
103 - Age

```
object delimiter{

val firstName = 50

}
```

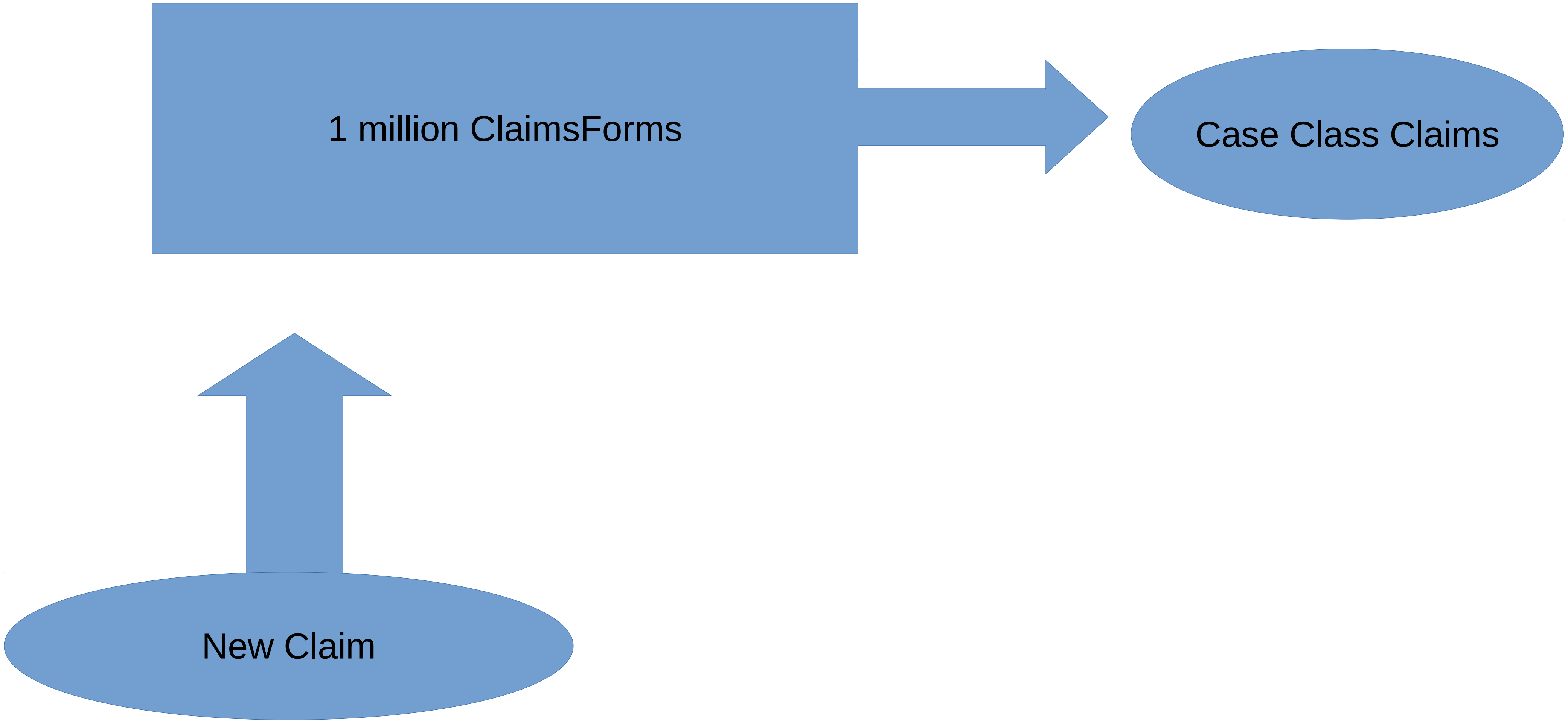


1 million ClaimsForms

Case Class Claims

ProspectsForms

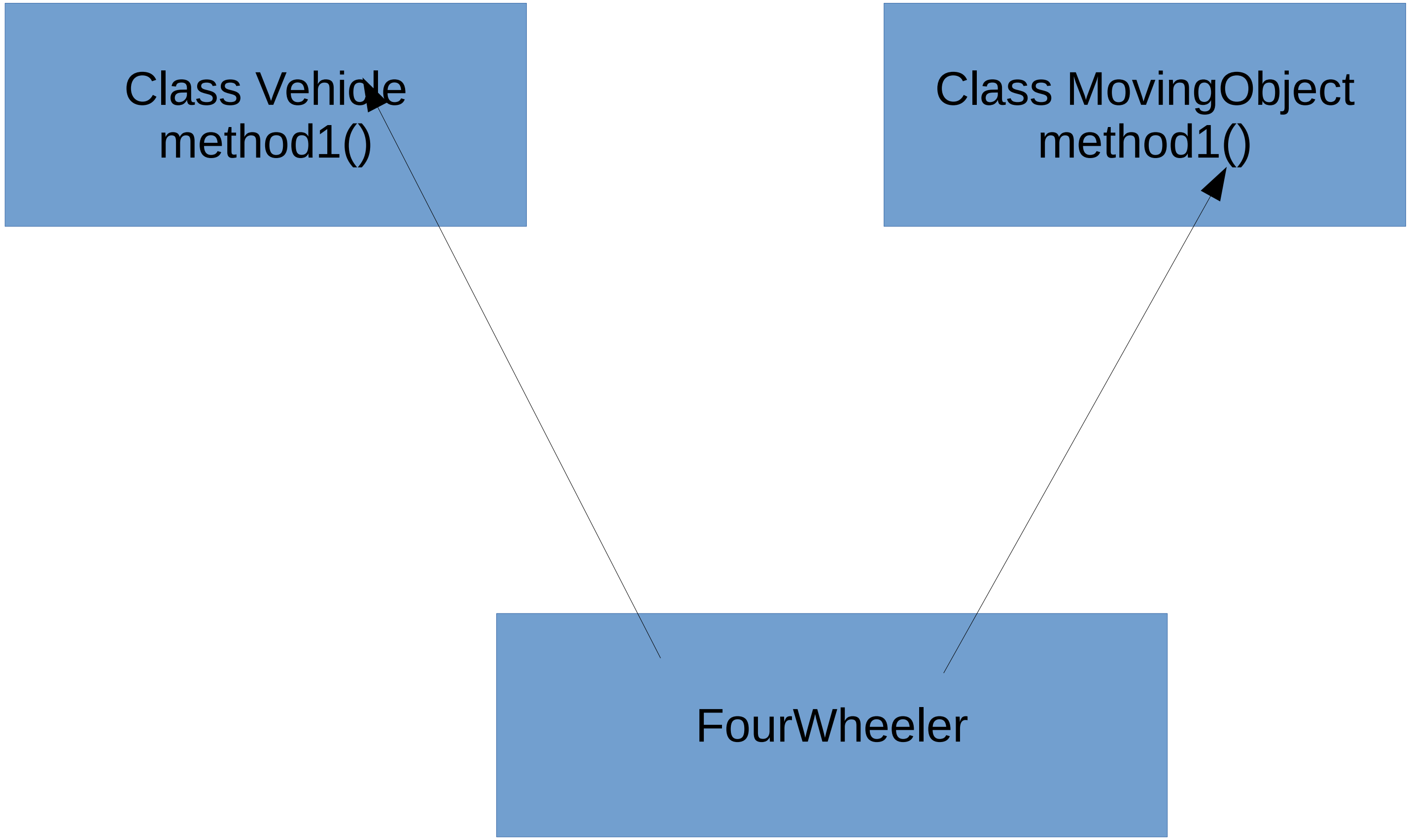
Case Class Prospects



1 million ClaimsForms

Case Class Claims

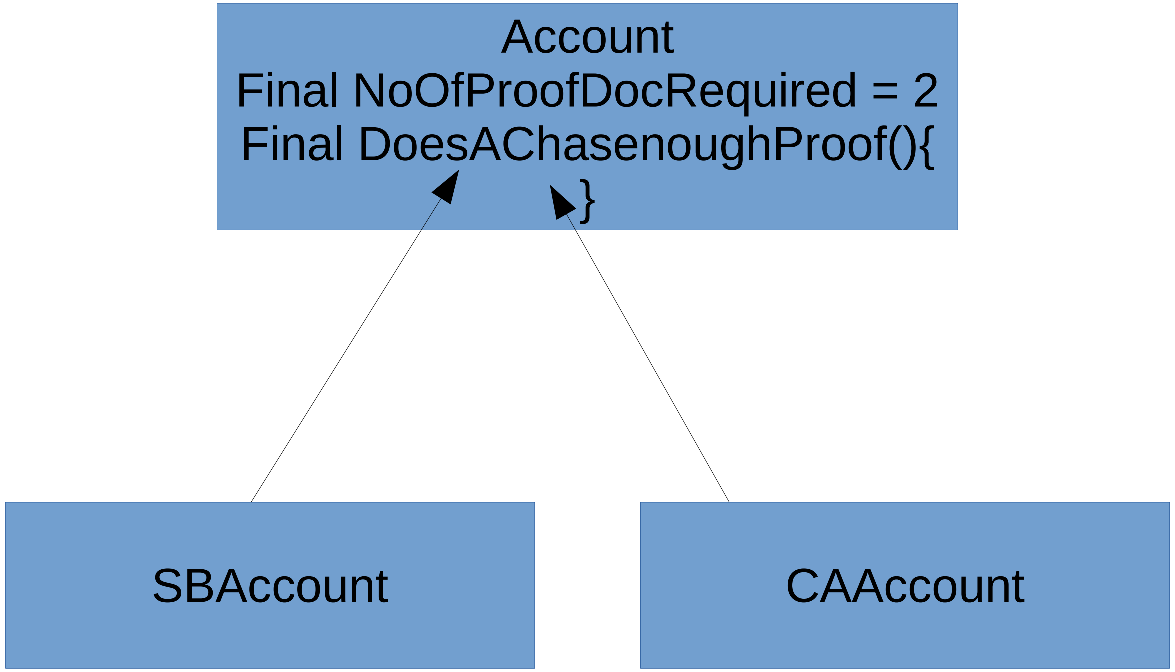
New Claim



Class Vehicle
method1()

Class MovingObject
method1()

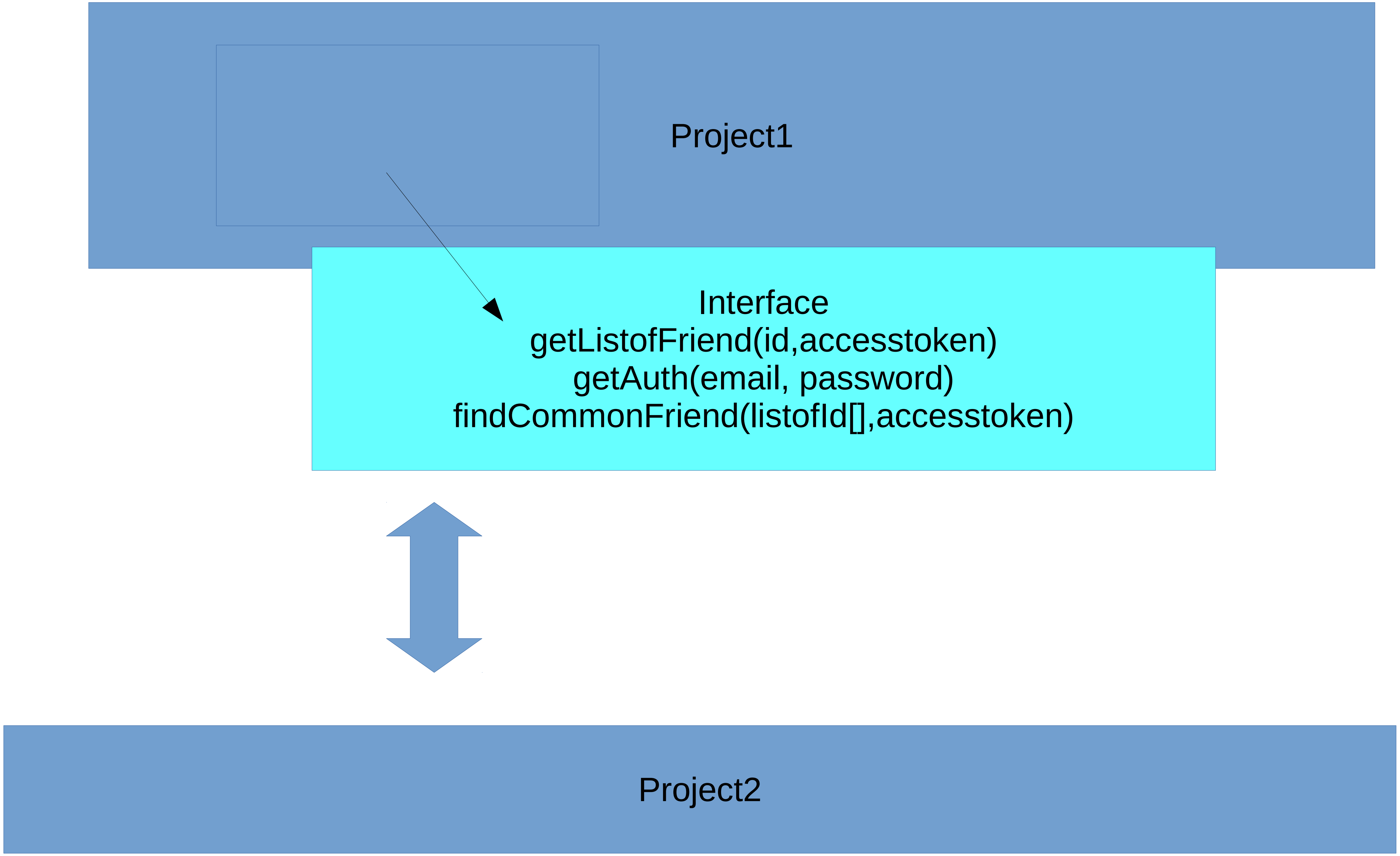
FourWheeler



Account
Final NoOfProofDocRequired = 2
Final DoesAChasenoughProof(){
}

SBAccount

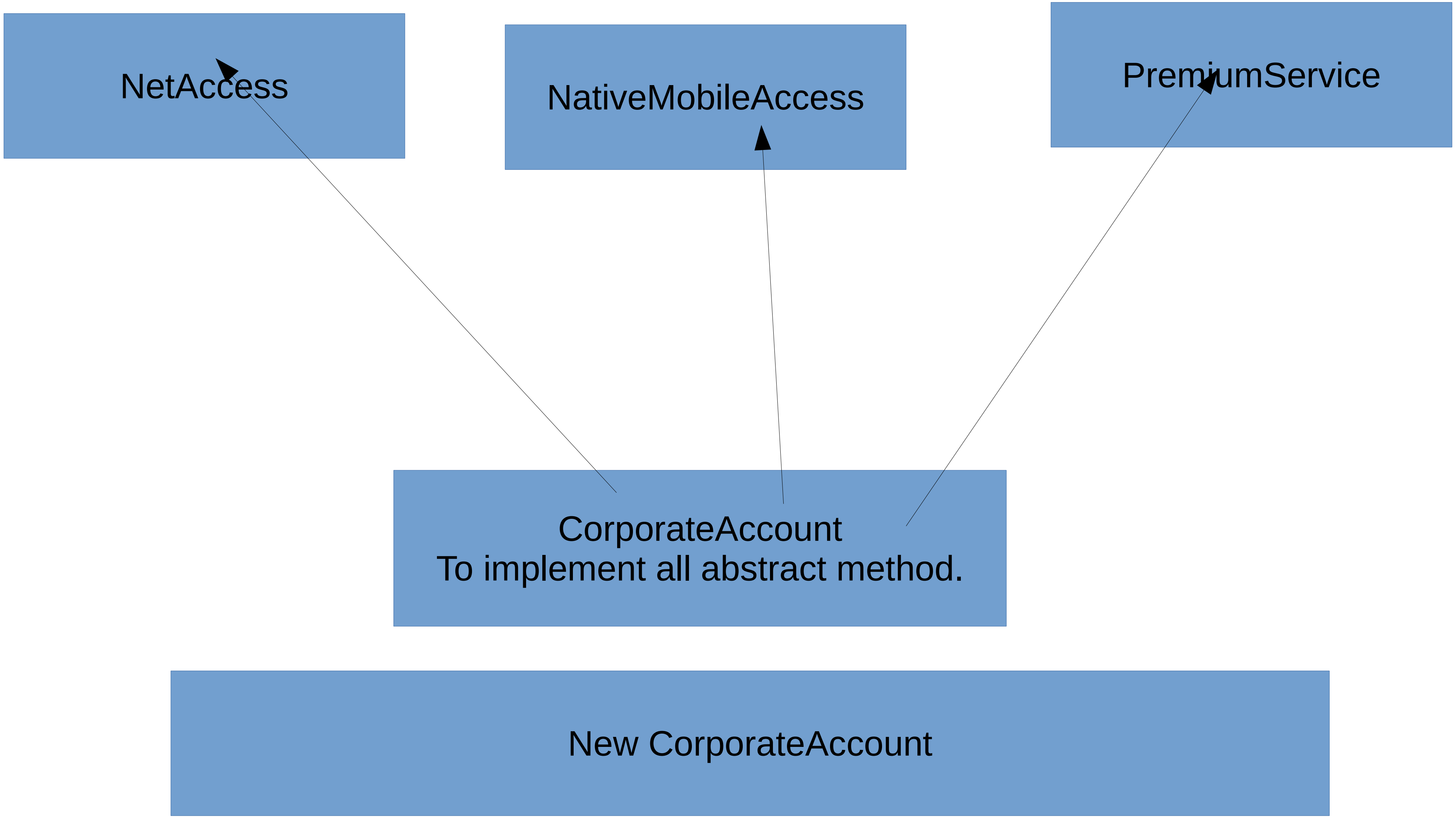
CAAccount



Project1

Interface
getListofFriend(id,accesstoken)
getAuth(email, password)
findCommonFriend(listofId[],accesstoken)

Project2



NetAccess

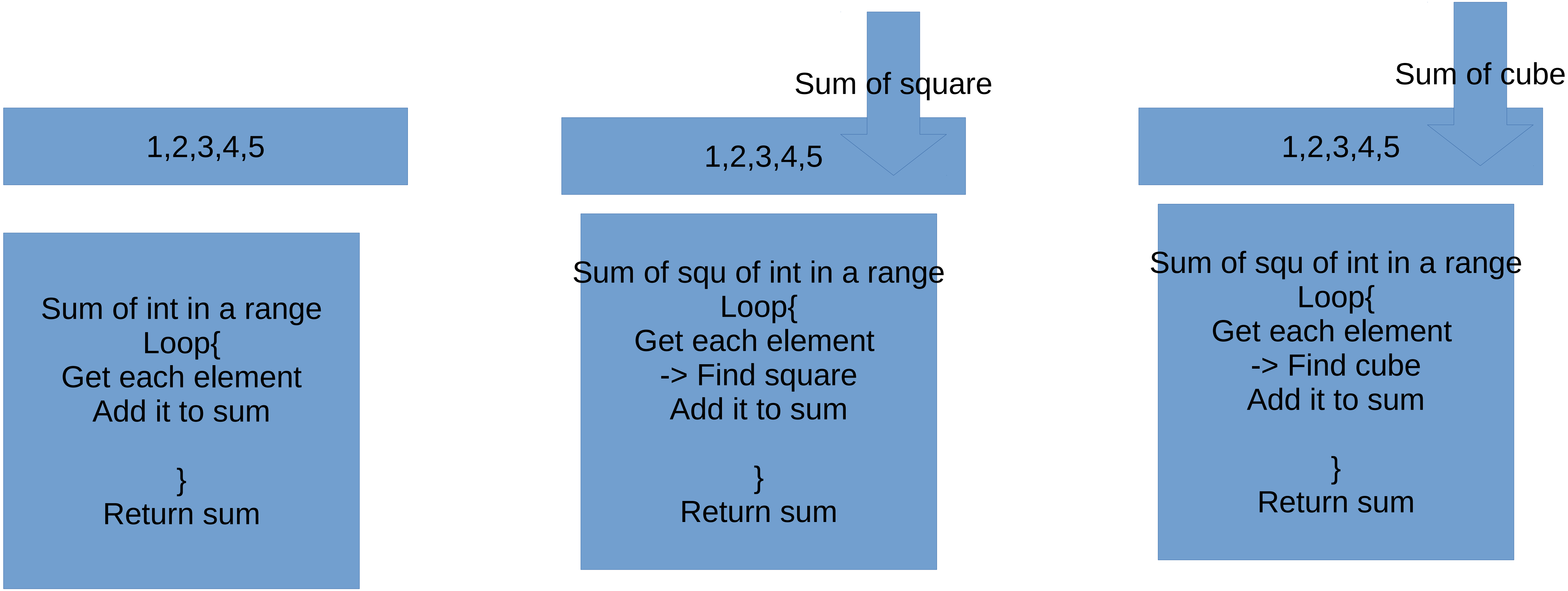
NativeMobileAccess

PremiumService

CorporateAccount
To implement all abstract method.

New CorporateAccount

Assign a function to a Variable
Pass a function as an argument to another function
Get function as return type from a function.



1,2,3,4,5

Sum of int in a range
Loop{
Get each element
Add it to sum

}
Return sum

Sum of square

1,2,3,4,5

Sum of squ of int in a range
Loop{
Get each element
-> Find square
Add it to sum

}
Return sum

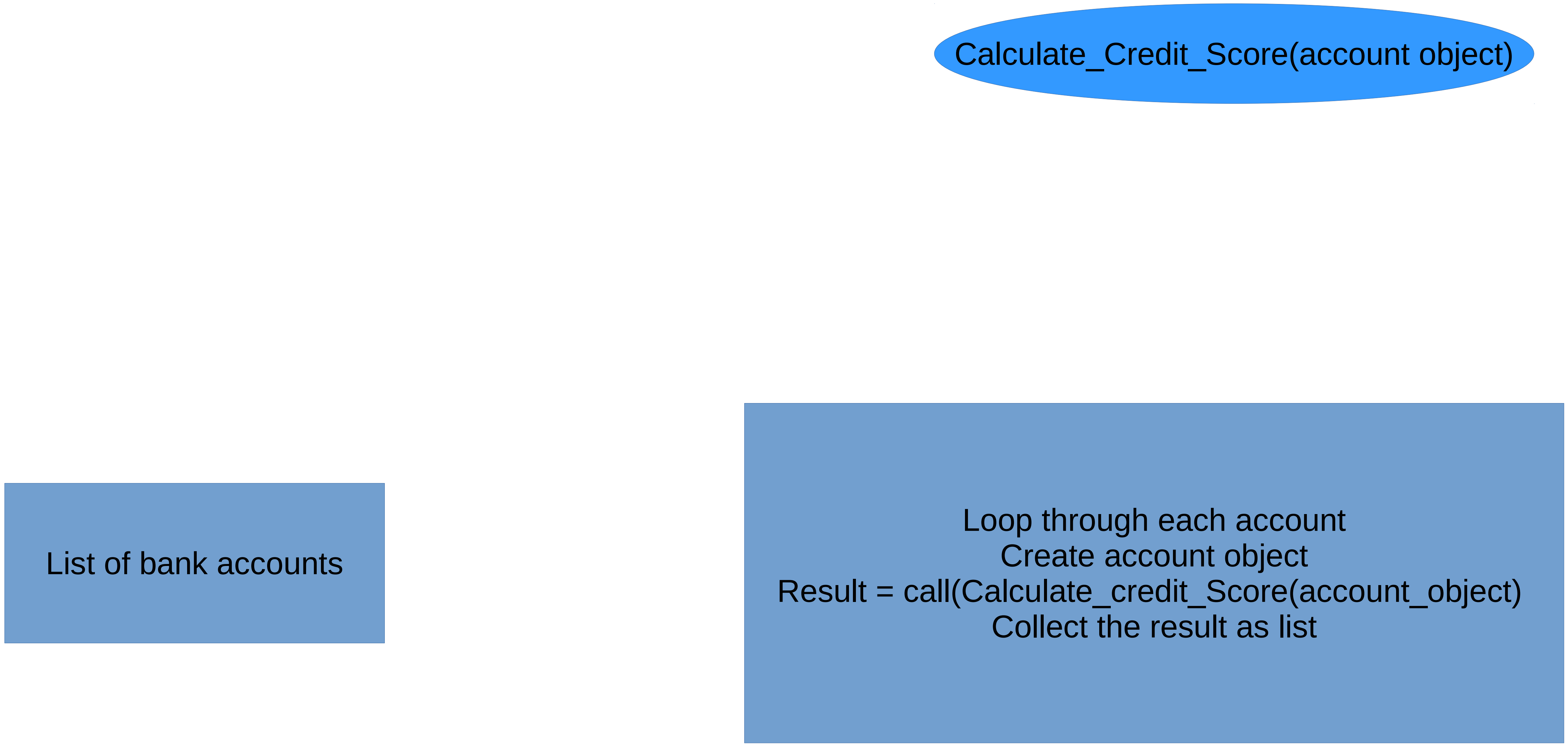
Sum of cube

1,2,3,4,5

Sum of squ of int in a range
Loop{
Get each element
-> Find cube
Add it to sum

}
Return sum

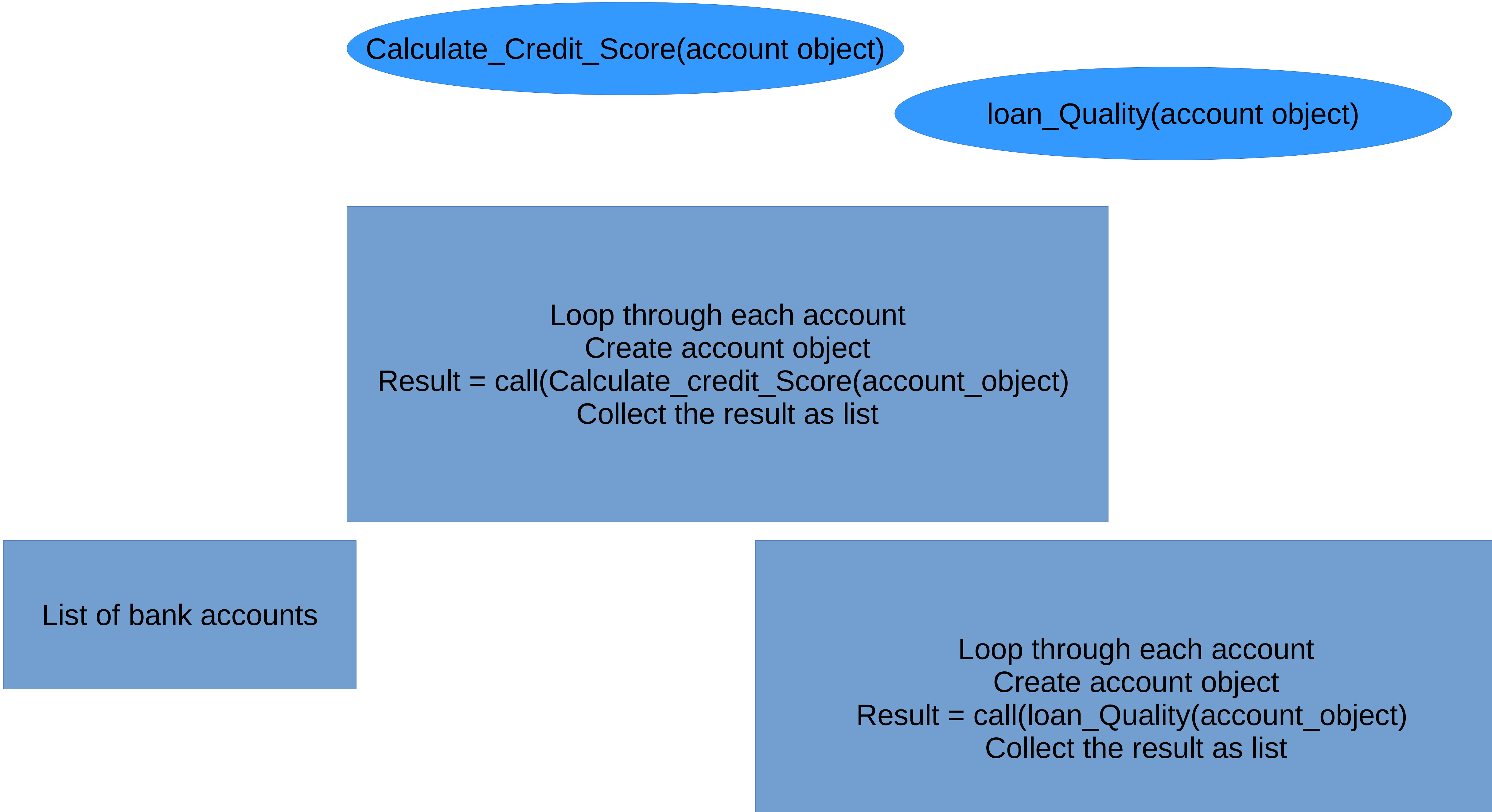
```
Function (cube, myNum[])
{
    Loop each number
    {
        cube(eachnumber)

        Sum = sum + each number
    }

}
```



Calculate_Credit_Score(account object)

List of bank accounts

Loop through each account
Create account object
Result = call(Calculate_credit_Score(account_object)
Collect the result as list



Calculate_Credit_Score(account object)

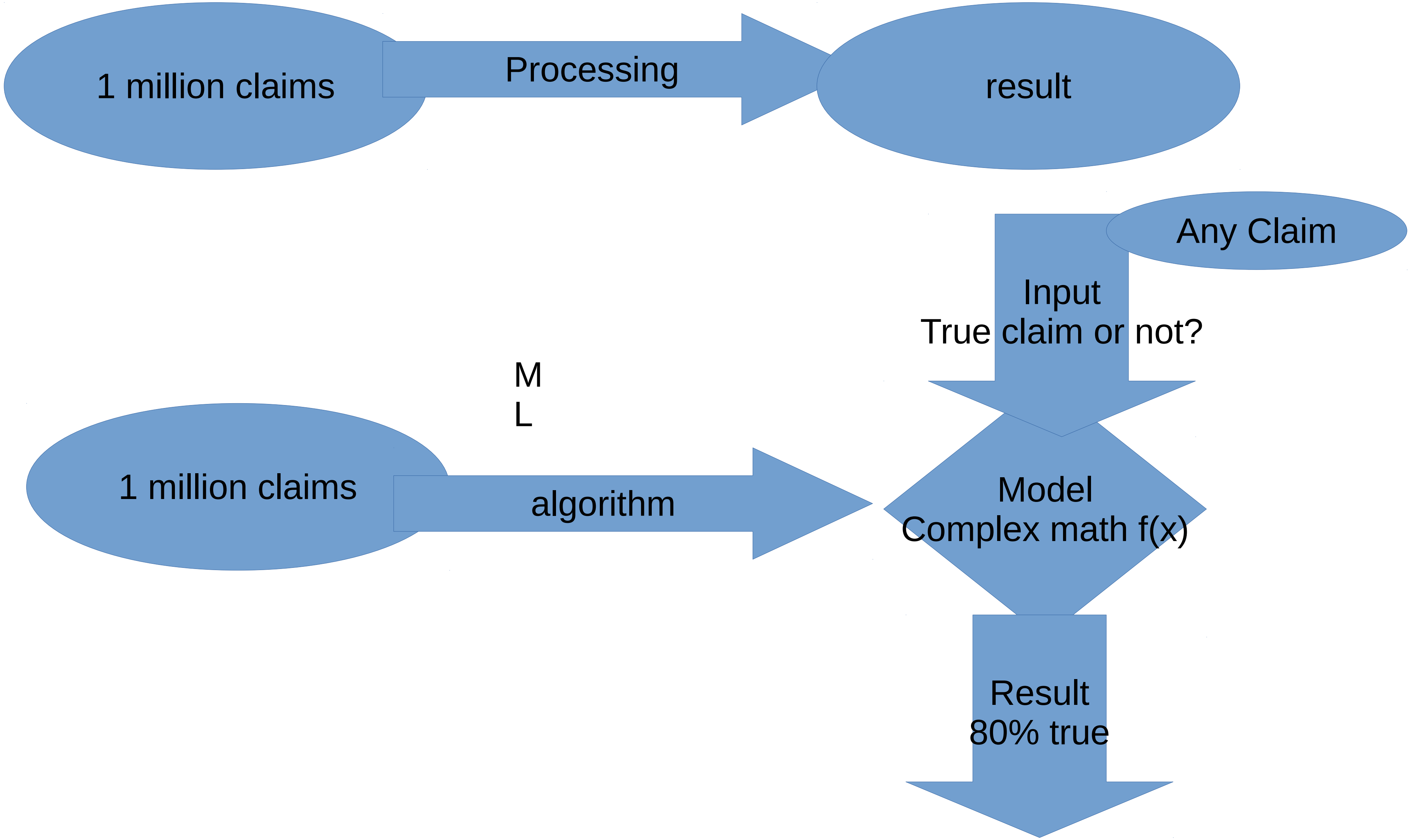
loan_Quality(account object)

```
Loop through each account
Create account object
Result = call(Calculate_credit_Score(account_object)
Collect the result as list
```

List of bank accounts

```
Loop through each account
Create account object
Result = call(loan_Quality(account_object)
Collect the result as list
```

Traditional



1 million claims

Processing

result

Any Claim

Input
True claim or not?

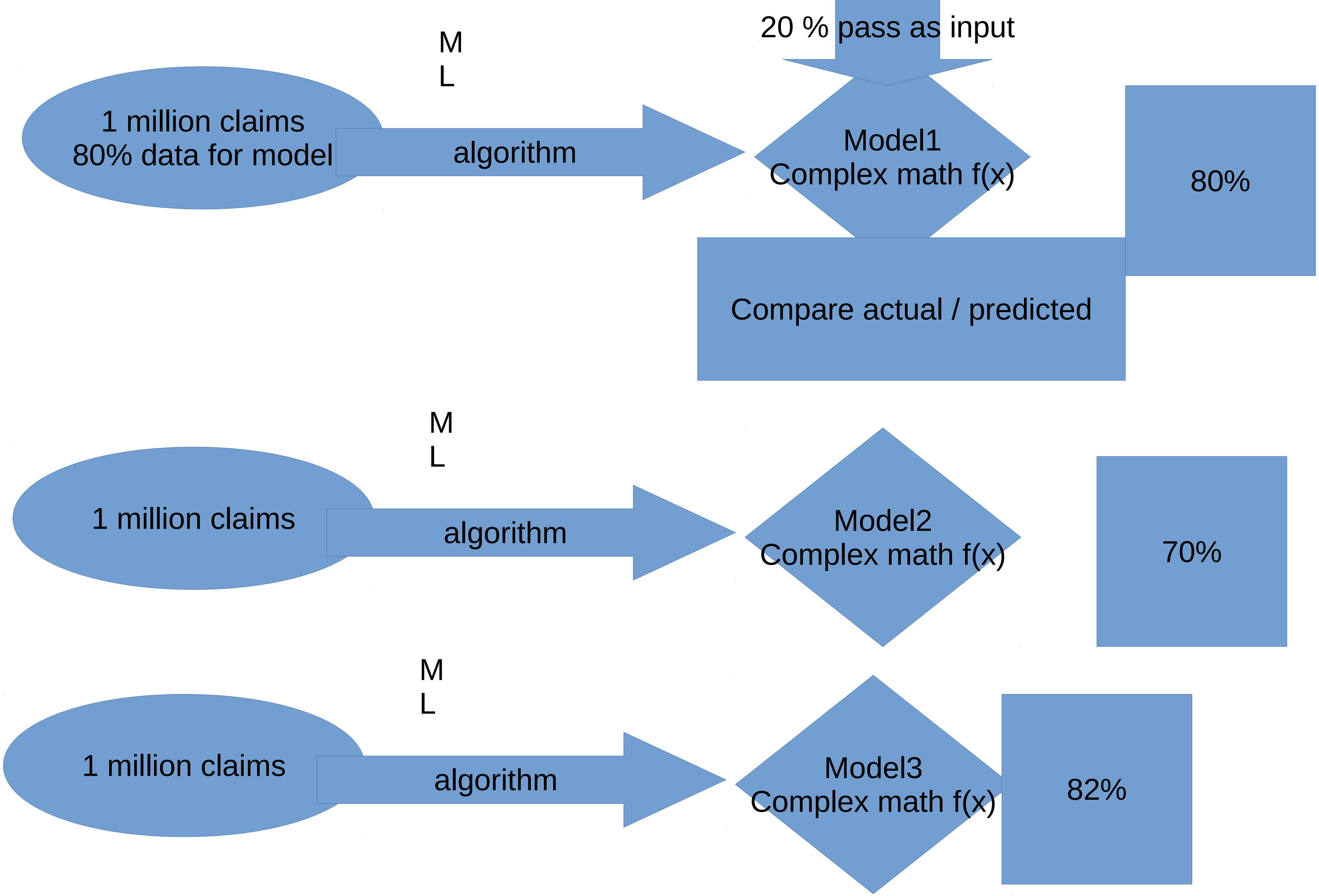
M
L

1 million claims

algorithm

Model
Complex math f(x)

Result
80% true



20 % pass as input

M
L

1 million claims
80% data for model

algorithm

Model1
Complex math f(x)

80%

Compare actual / predicted

M
L

1 million claims

algorithm

Model2
Complex math f(x)

70%

M
L

1 million claims

algorithm

Model3
Complex math f(x)

82%

ModelToUse = Model5

PredictIsITTrueClaim(claimObject, ModelToUse)

Result = modelToUse(co.name, co.accout#,co.ssn)

Production

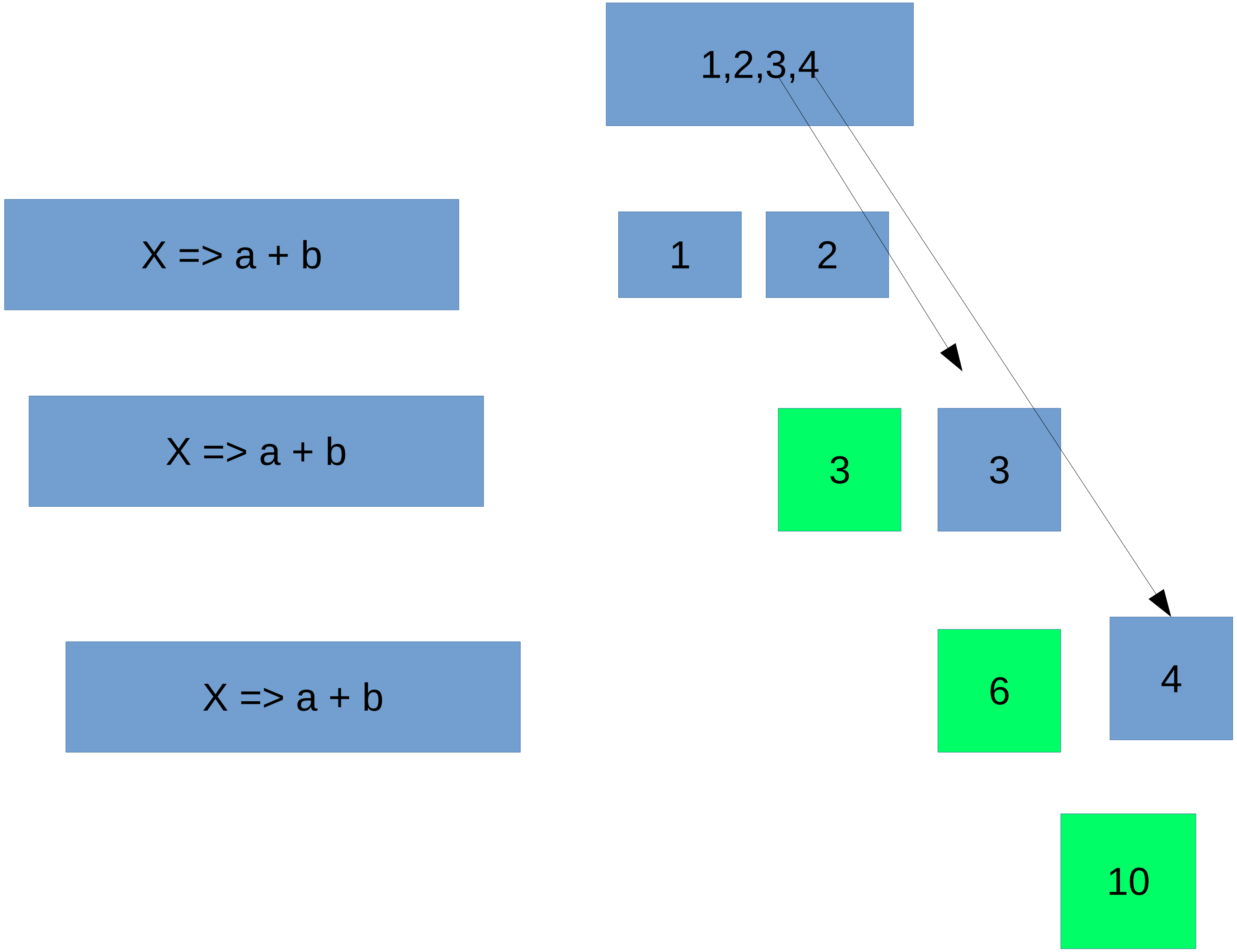
```
x  =>  X+2

_  *  2
```

PredictTrueClaims

100 million Claims.map(PredictTrueClaims)

(True, True, False,False,False,True........)

Val myStr = “Edureka”
println(myStr) or println(“Edureka”)



1,2,3,4

X => a + b

1

2

X => a + b

3

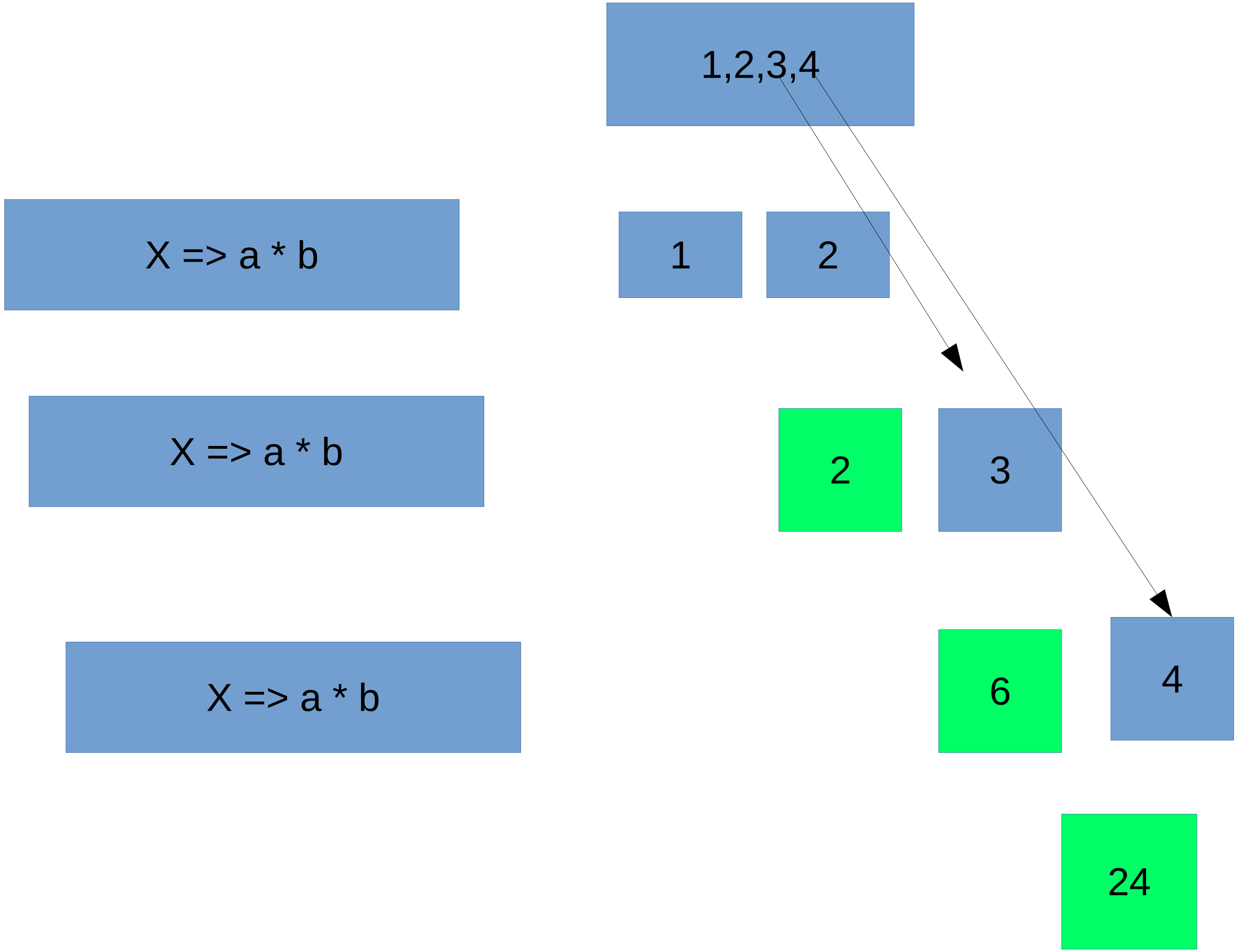
3

X => a + b

6

4

10



1,2,3,4

X => a * b

1

2

X => a * b

2

3

X => a * b

6

4

24

Mary had a little lamb Mary

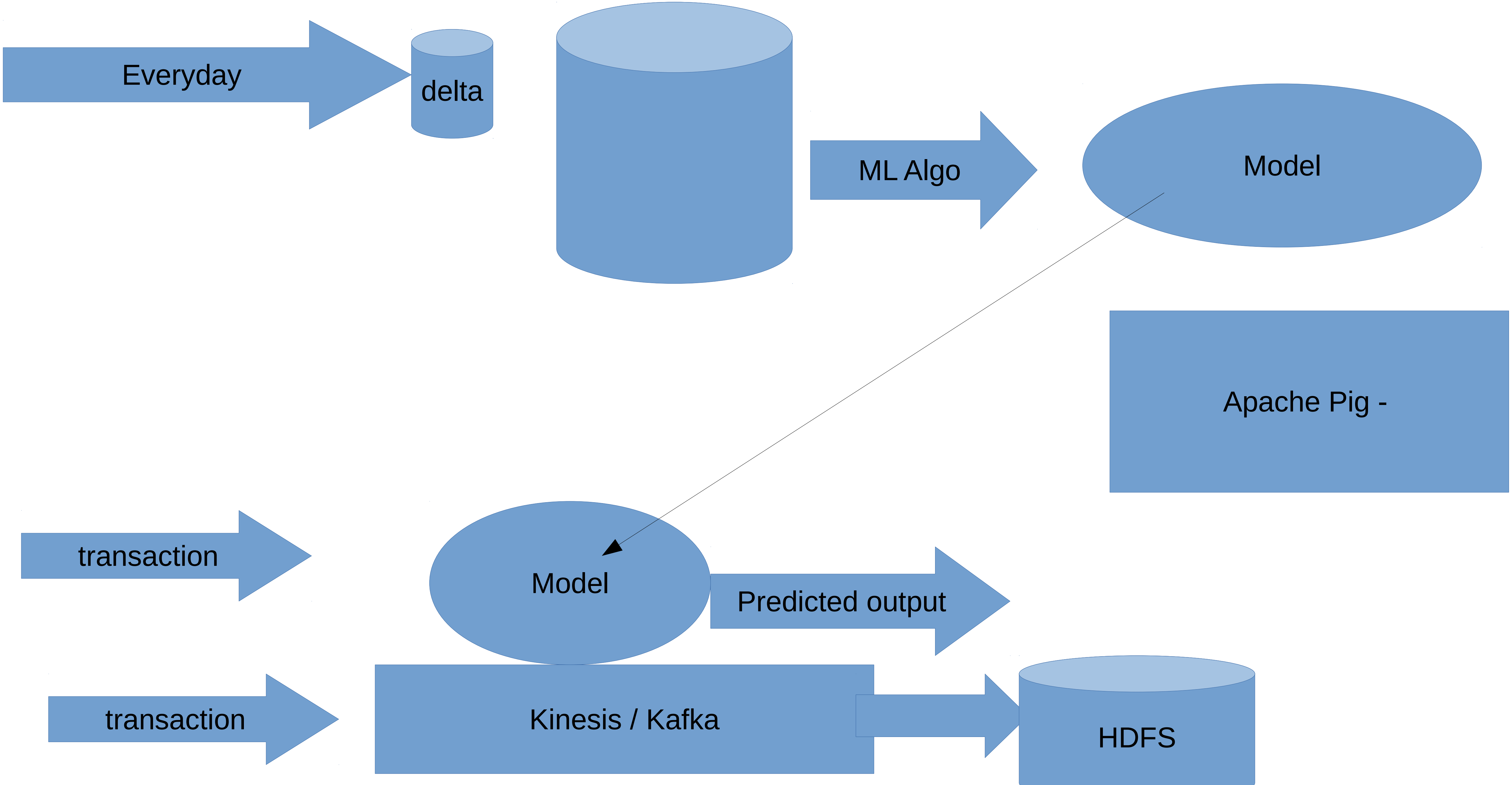
ary”.length < “had”.length

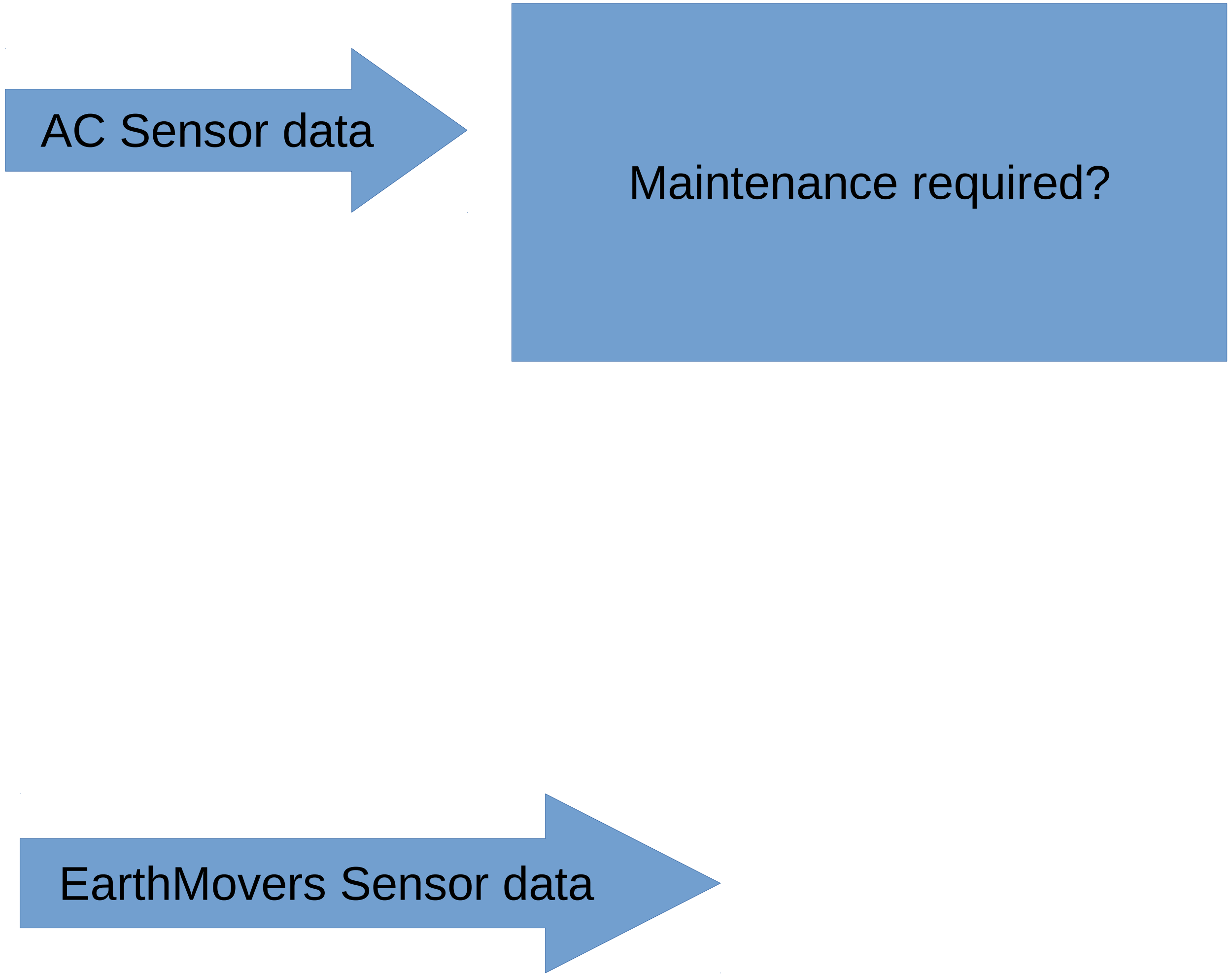
Mary, had

4 < 3

ModelForHighSalary

ModelForHighSavings

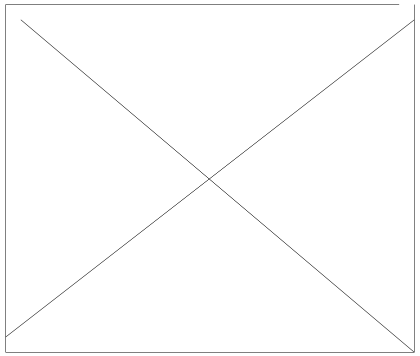

Everyday
delta
ML Algo
Model
Apache Pig -
transaction
Model
Predicted output
transaction
Kinesis / Kafka
HDFS

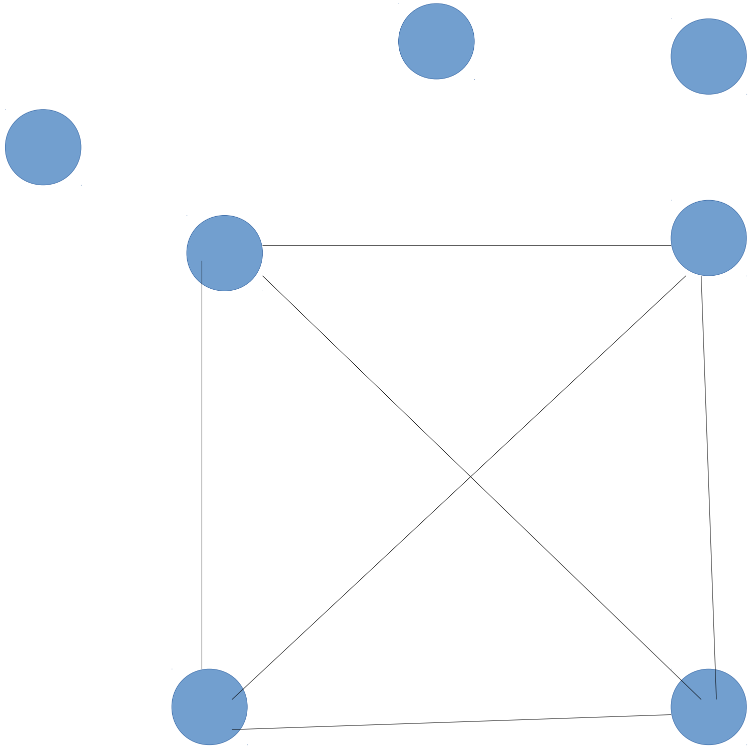


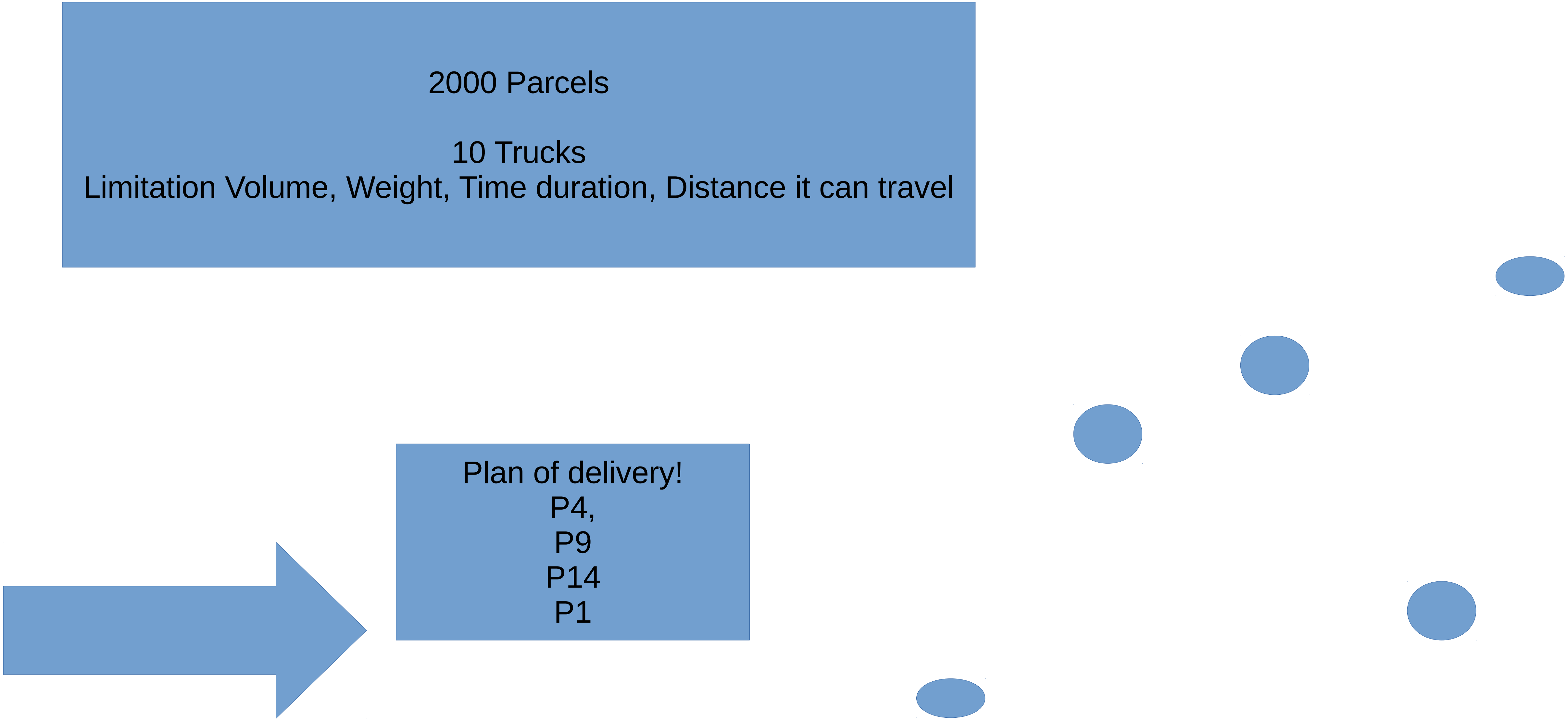
AC Sensor data

Maintenance required?

EarthMovers Sensor data



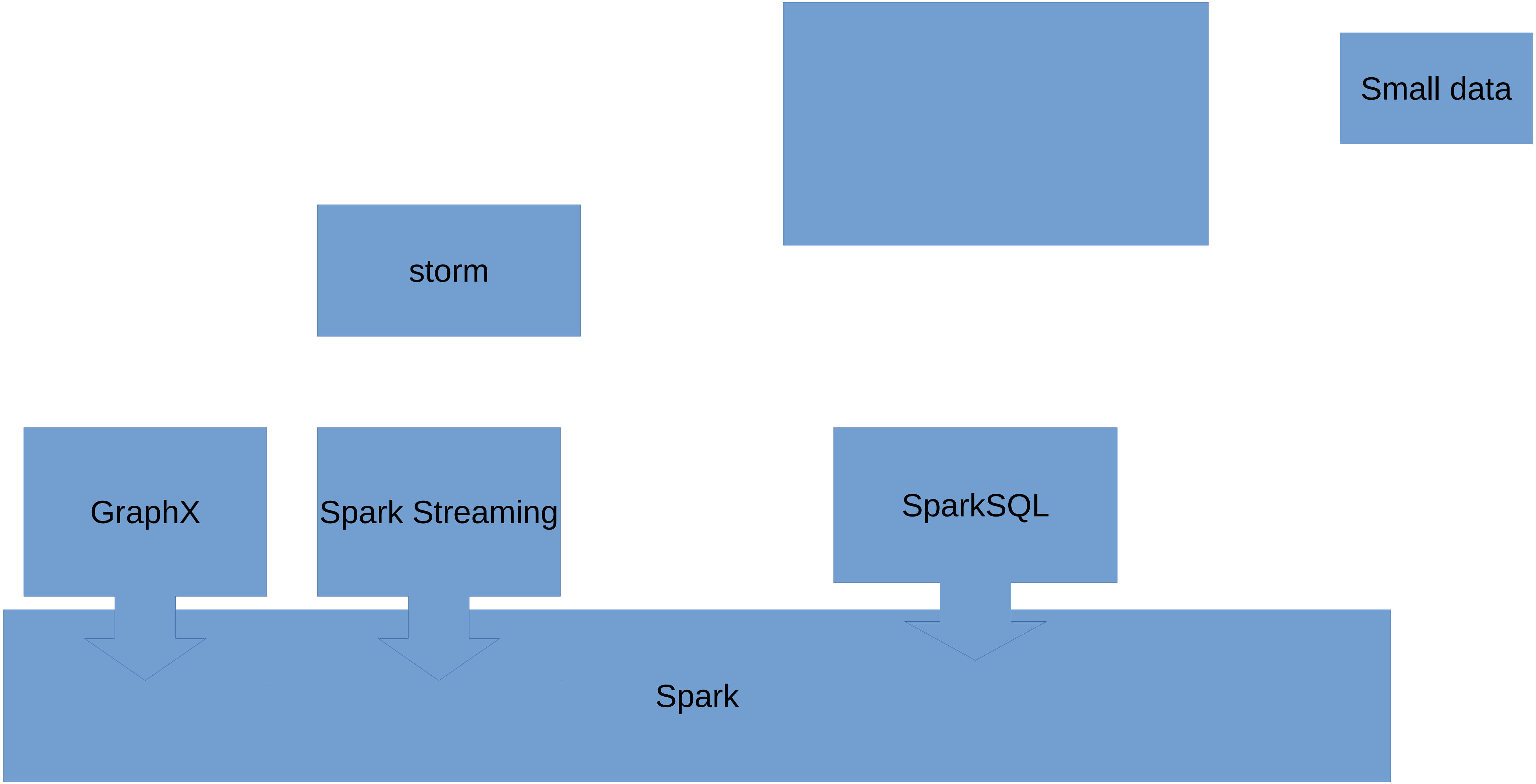




2000 Parcels

10 Trucks
Limitation Volume, Weight, Time duration, Distance it can travel

Plan of delivery!
P4,
P9
P14
P1



Small data

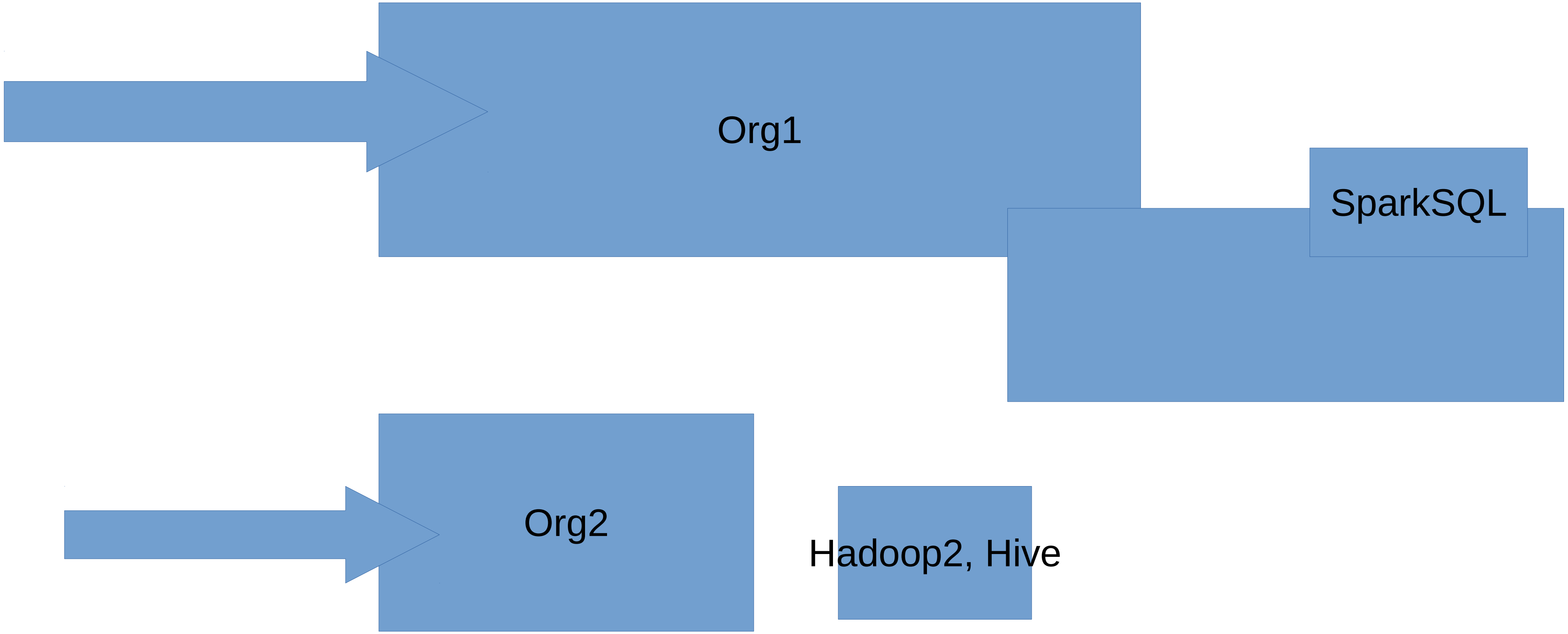
storm

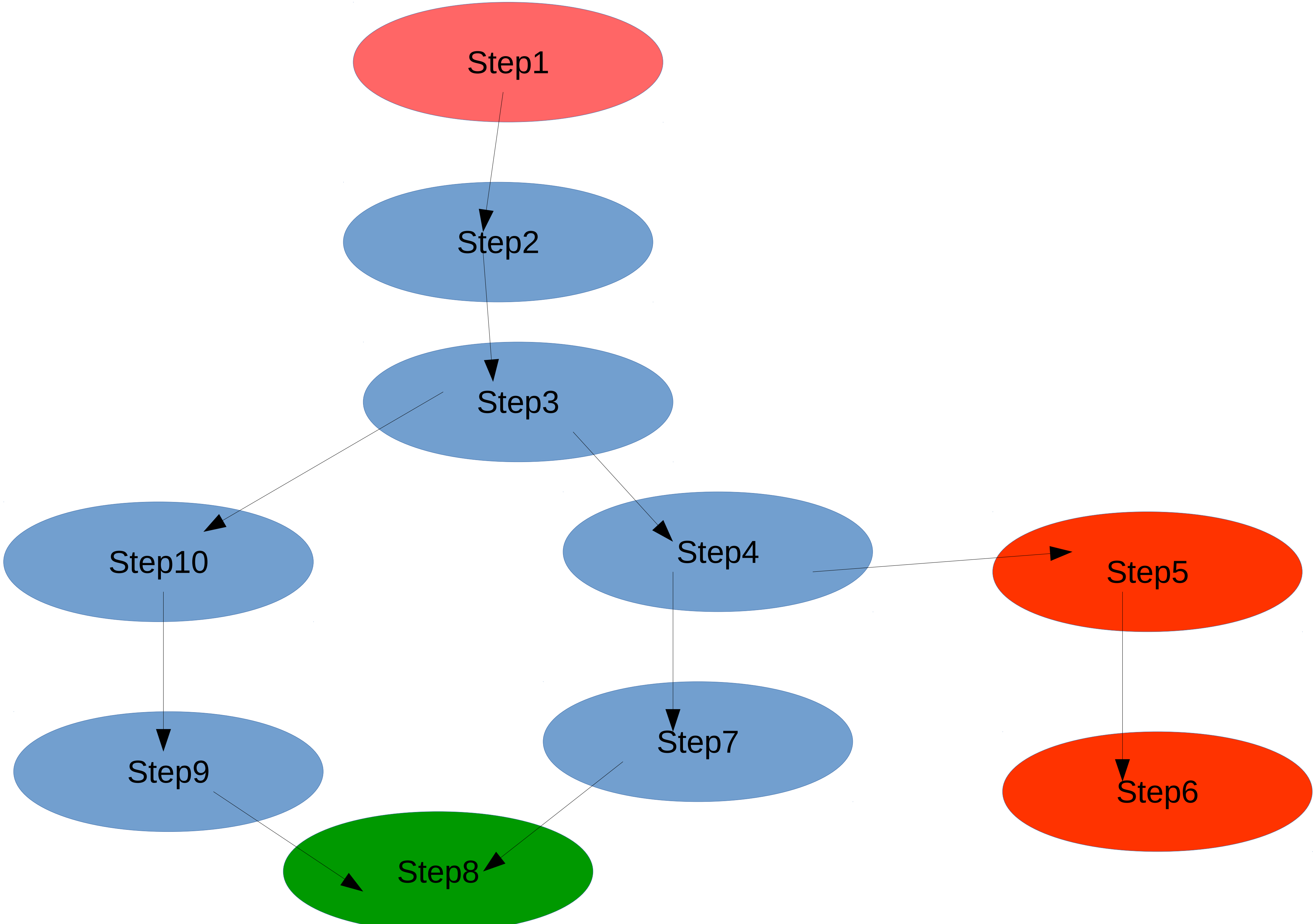
GraphX

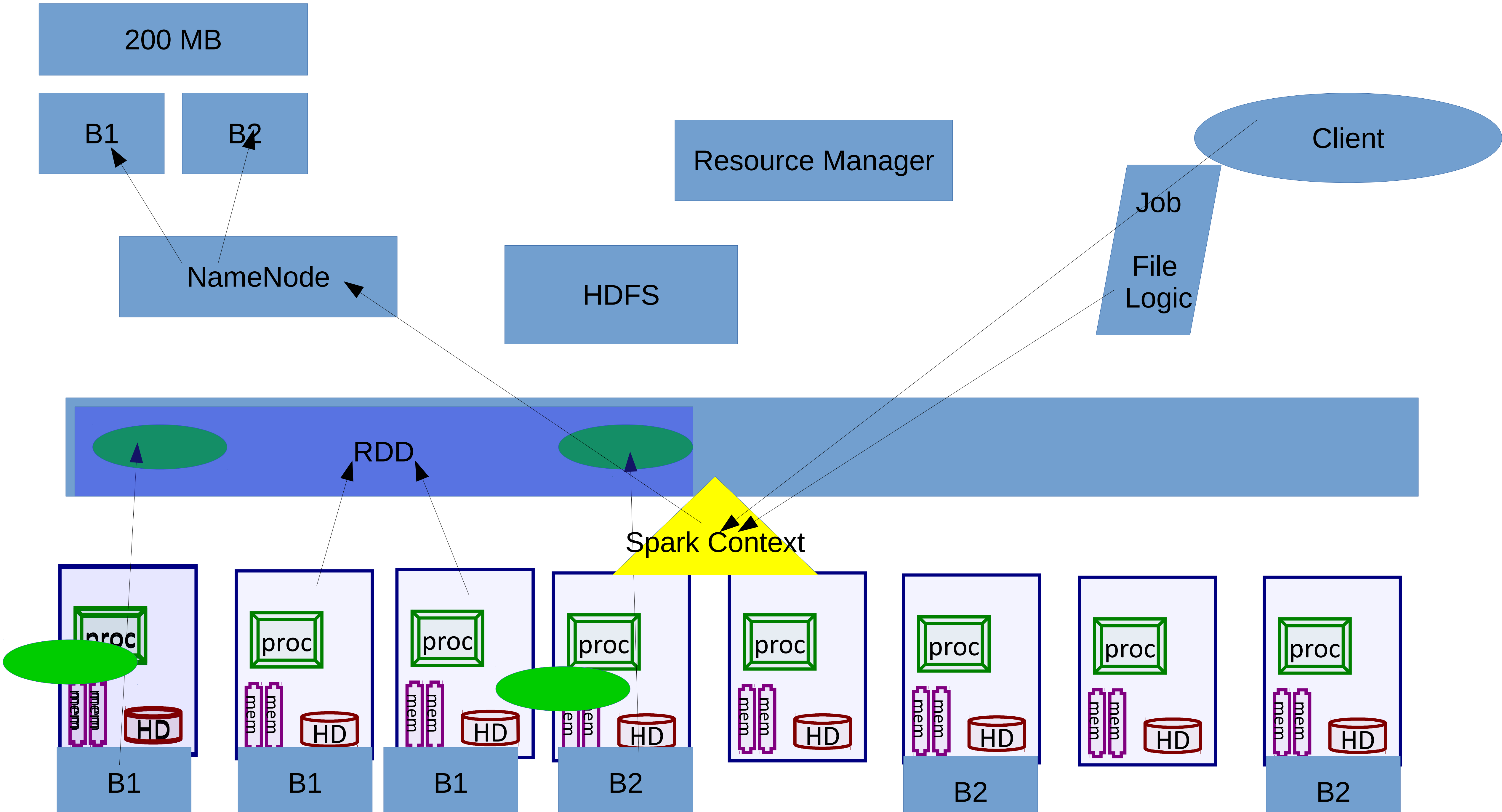
Spark Streaming

SparkSQL

Spark


Org1
SparkSQL
Org2
Hadoop2, Hive


Step1
Step2
Step3
Step10
Step4
Step5
Step9
Step7
Step6
Step8



200 MB

B1

B2

Resource Manager

Client

Job

File Logic

NameNode

HDFS

RDD

Spark Context

proc

mem mem

HD

B1

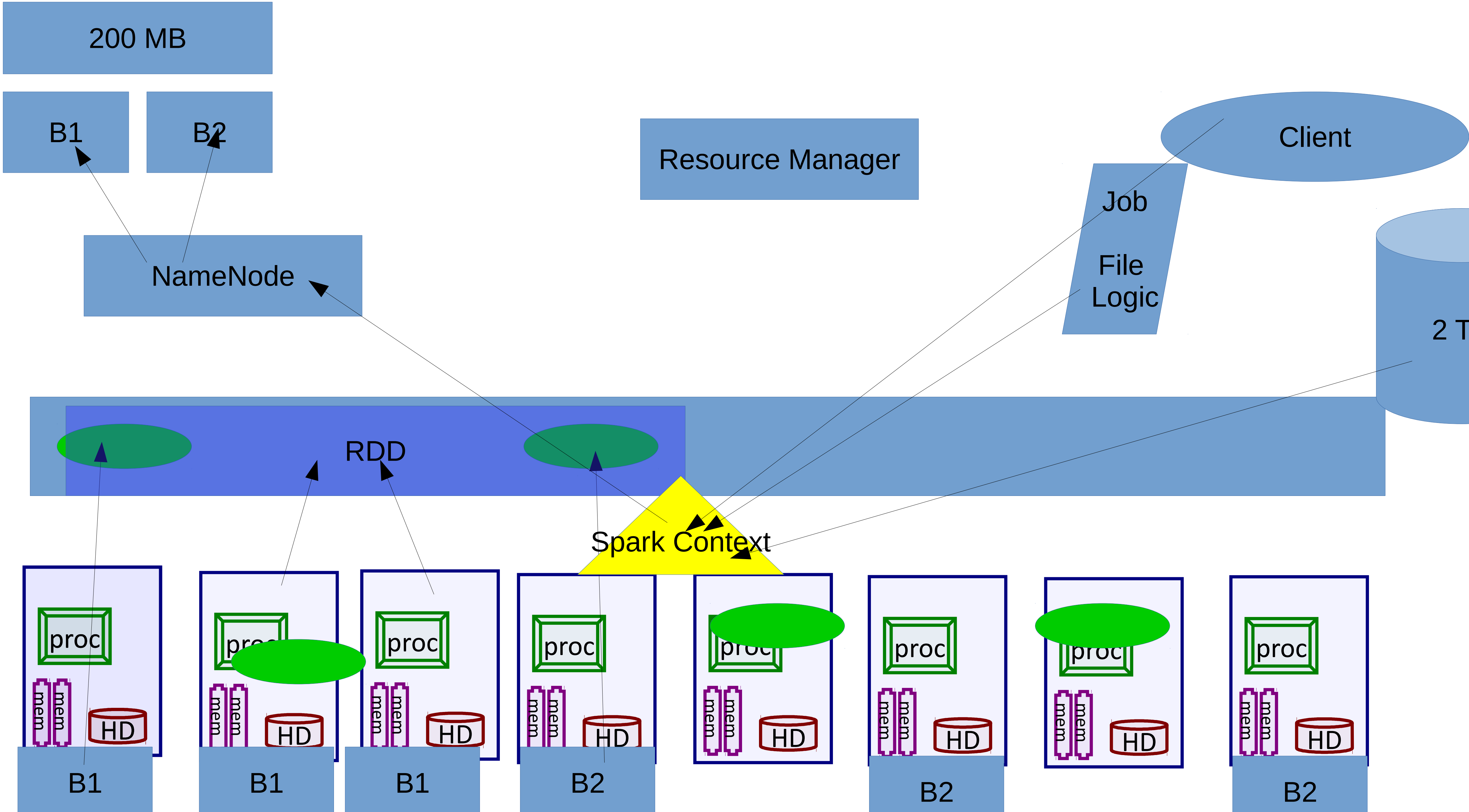
B1

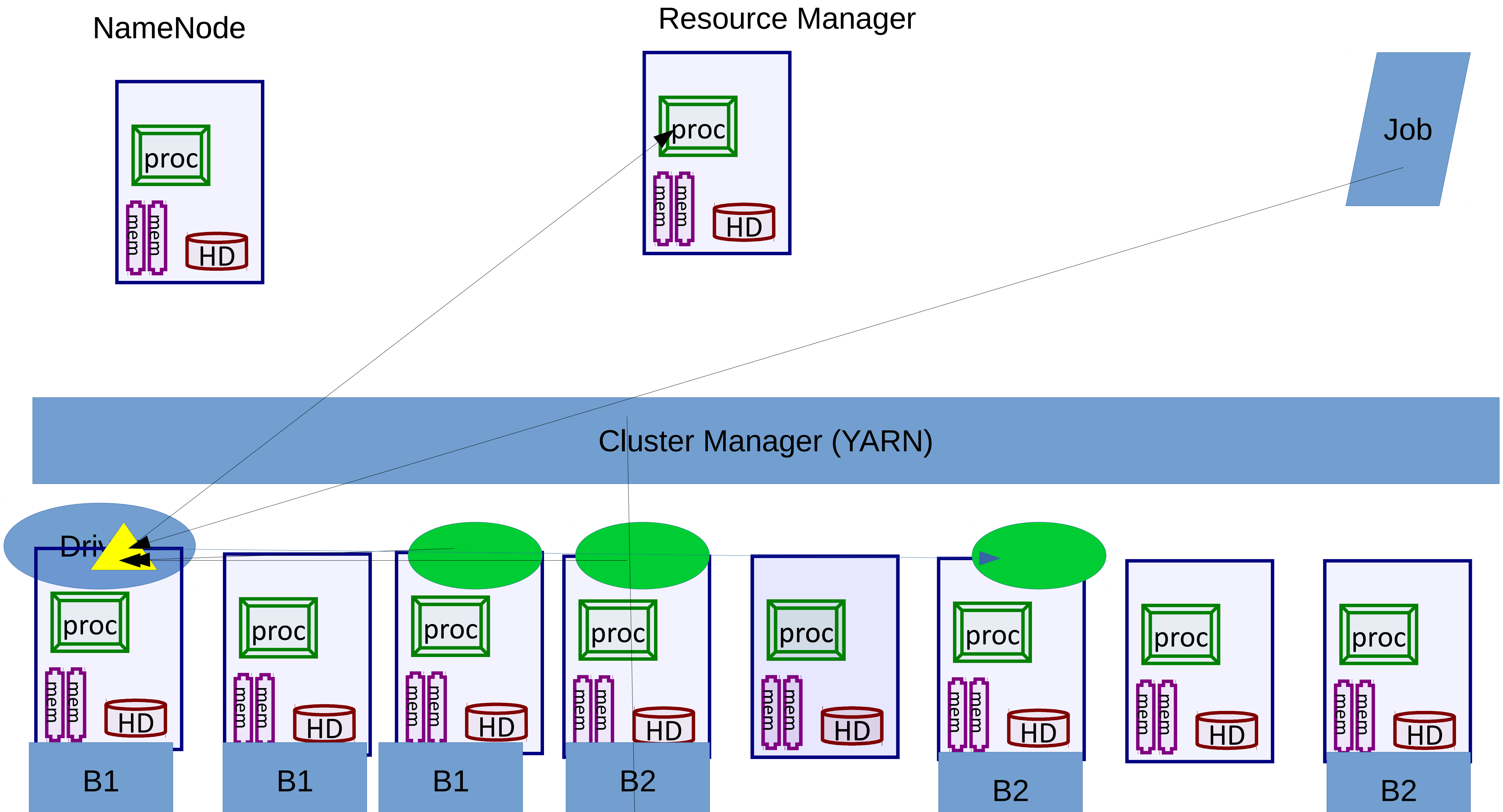
B1

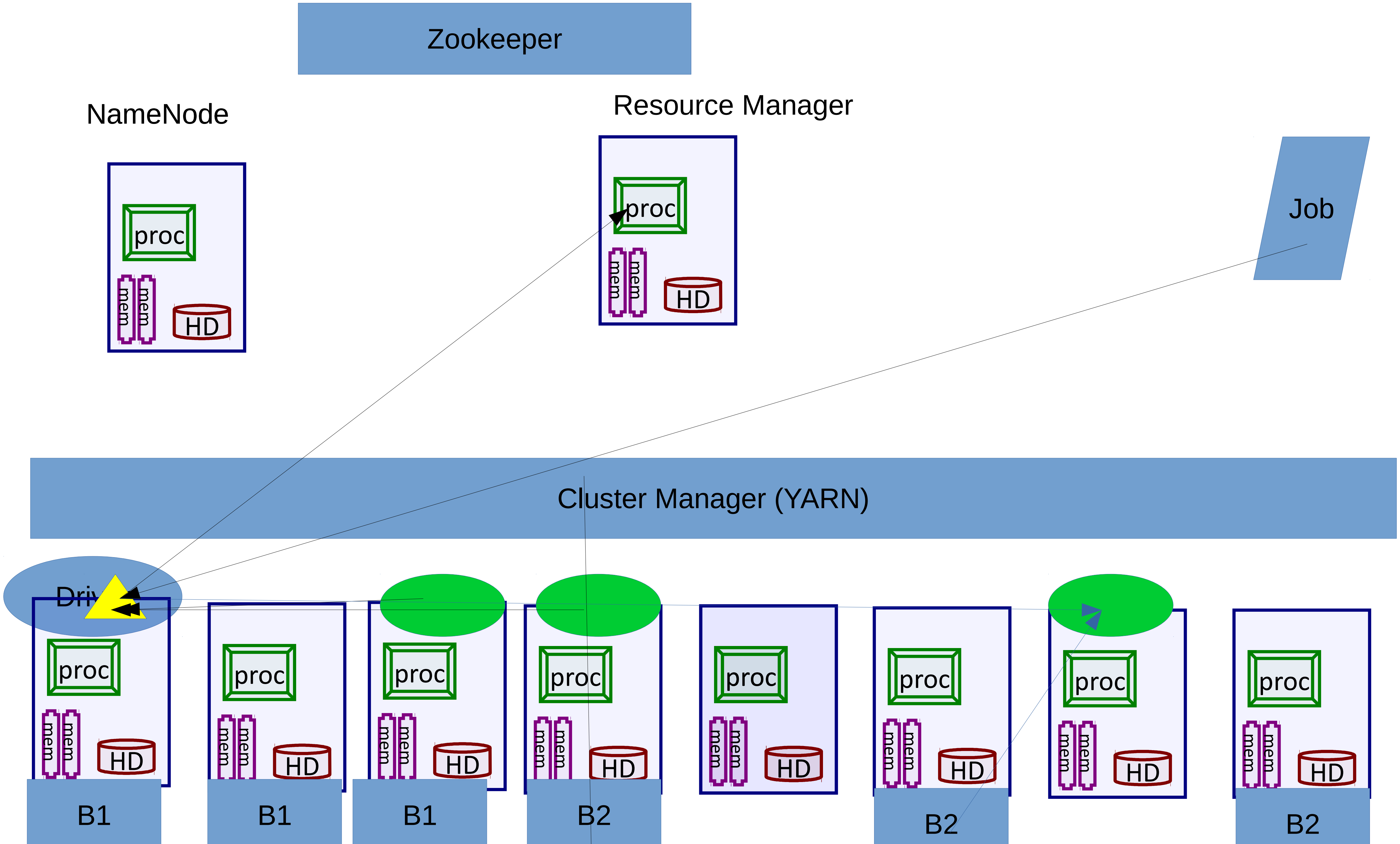
B2

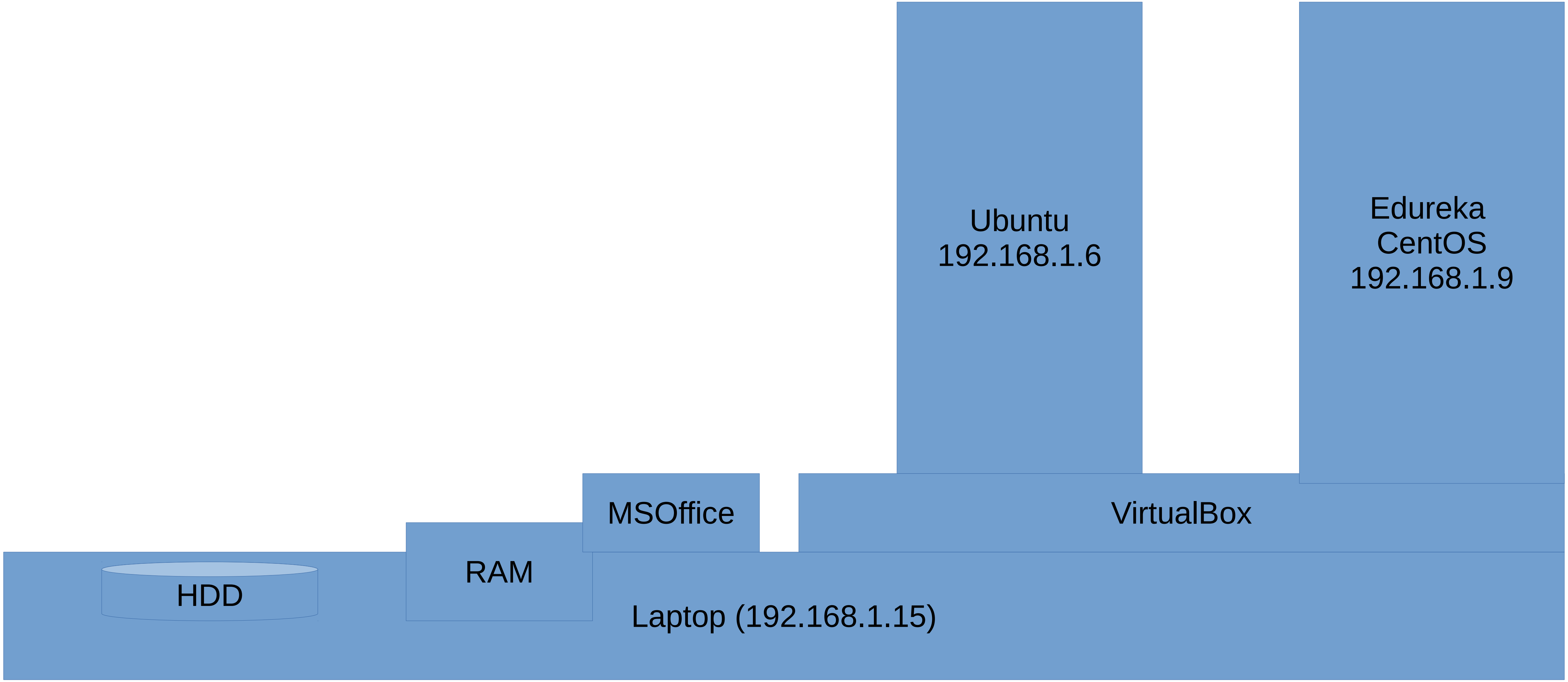
B2

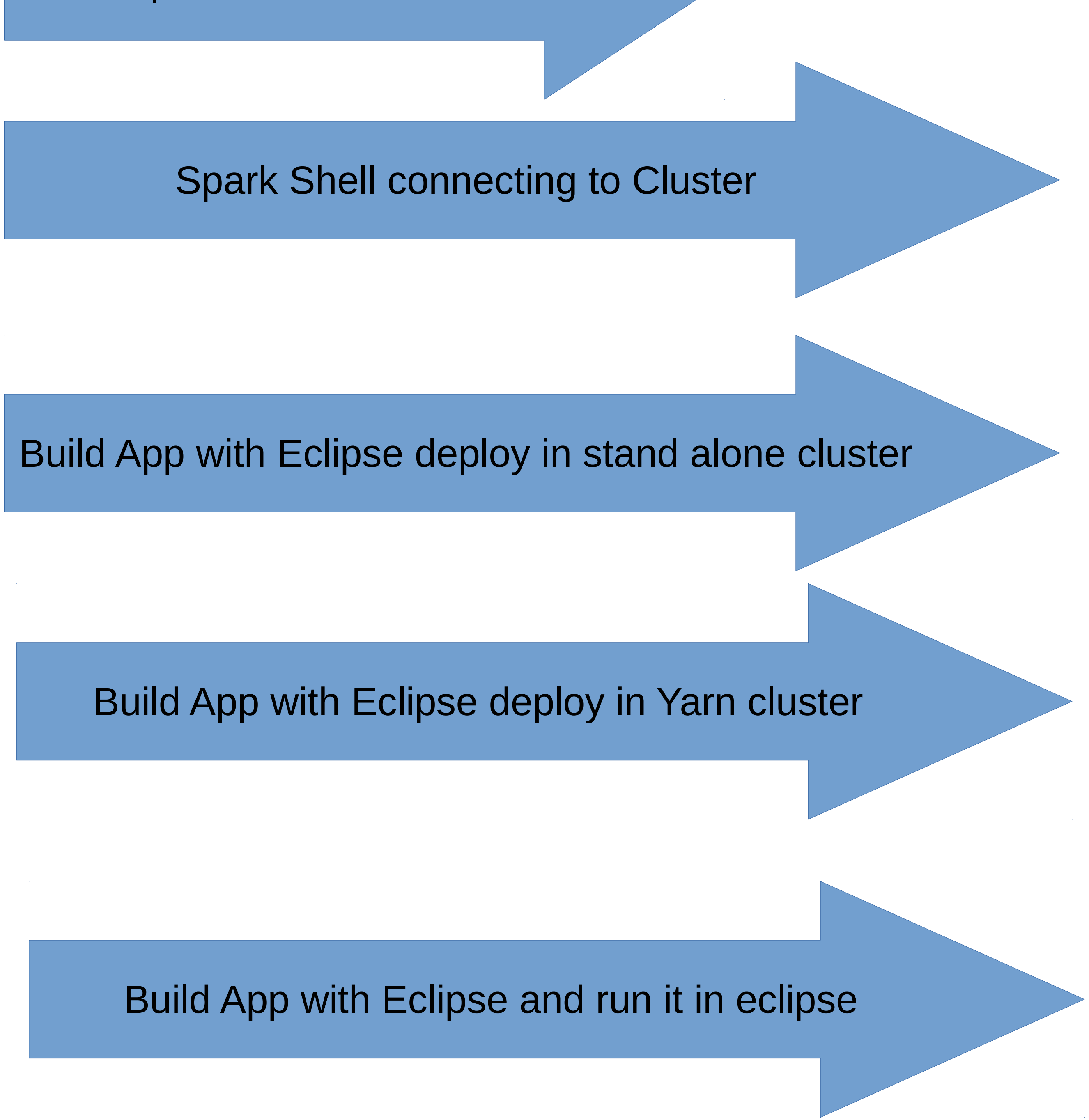
B2


200 MB
B1
B2
NameNode
Resource Manager
Client
Job
File
Logic
2 T
RDD
Spark Context
proc
mem
mem
HD
B1
B1
B1
B2
B2
B2


NameNode
proc
mem
mem
HD
Resource Manager
proc
mem
mem
HD
Job
Cluster Manager (YARN)
Driv
proc
mem
mem
HD
B1
proc
mem
mem
HD
B1
proc
mem
mem
HD
B1
proc
mem
mem
HD
B2
proc
mem
mem
HD
proc
mem
mem
HD
B2
proc
mem
mem
HD
proc
mem
mem
HD
B2


Zookeeper
NameNode
proc
mem
mem
HD
Resource Manager
proc
mem
mem
HD
Job
Cluster Manager (YARN)
Driv
proc
mem
mem
HD
B1
proc
mem
mem
HD
B1
proc
mem
mem
HD
B1
proc
mem
mem
HD
B2
proc
mem
mem
HD
proc
mem
mem
HD
B2
proc
mem
mem
HD
proc
mem
mem
HD
B2


Ubuntu
192.168.1.6
Edureka
CentOS
192.168.1.9
MSOffice
VirtualBox
RAM
HDD
Laptop (192.168.1.15)

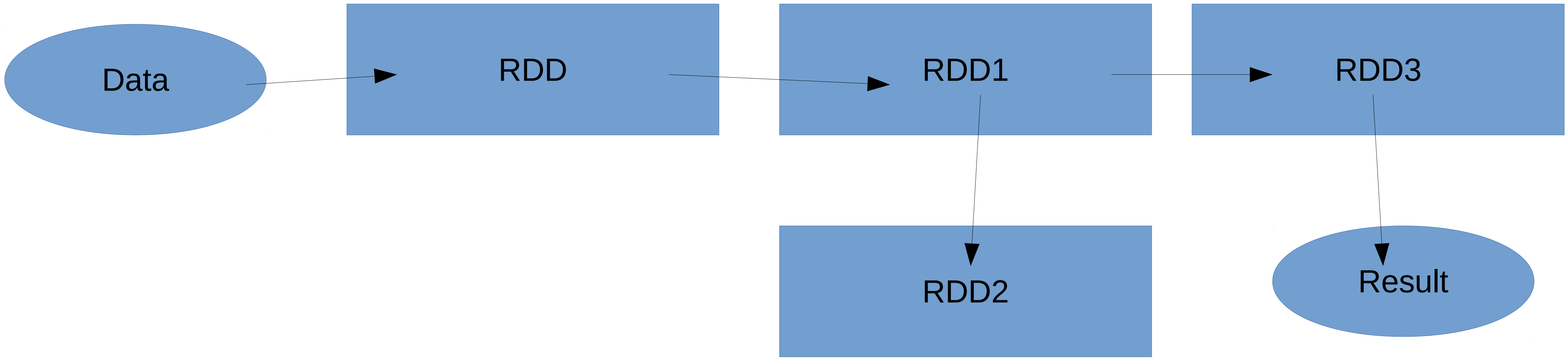


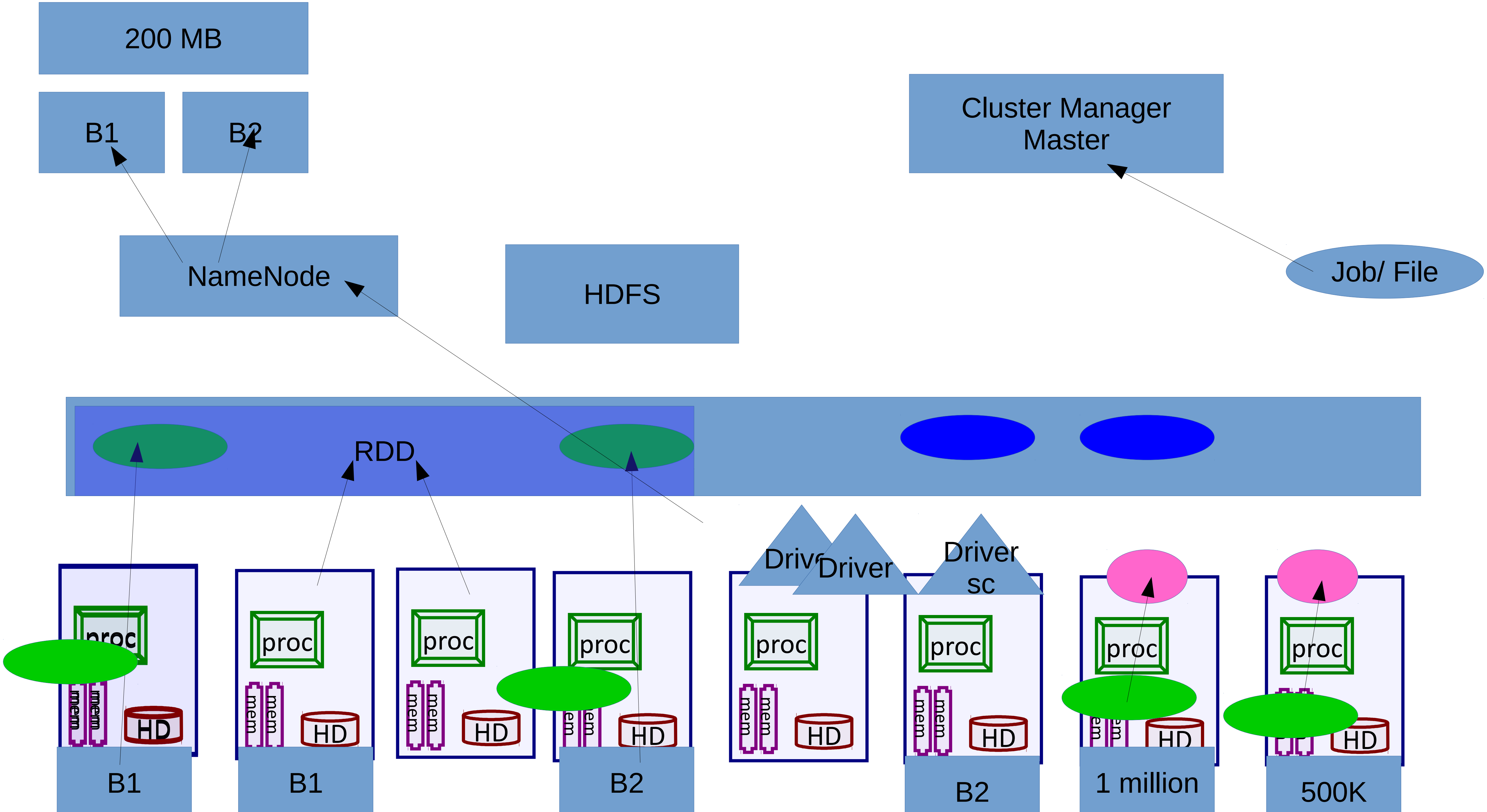
Spark Shell connecting to Cluster

Build App with Eclipse deploy in stand alone cluster

Build App with Eclipse deploy in Yarn cluster

Build App with Eclipse and run it in eclipse


Data
RDD
RDD1
RDD3
RDD2
Result



200 MB

B1

B2

NameNode

HDFS

Cluster Manager
Master

Job/ File

RDD

Driv

Driver

Driver
sc

proc

mem

mem

HD

B1

B1

B2

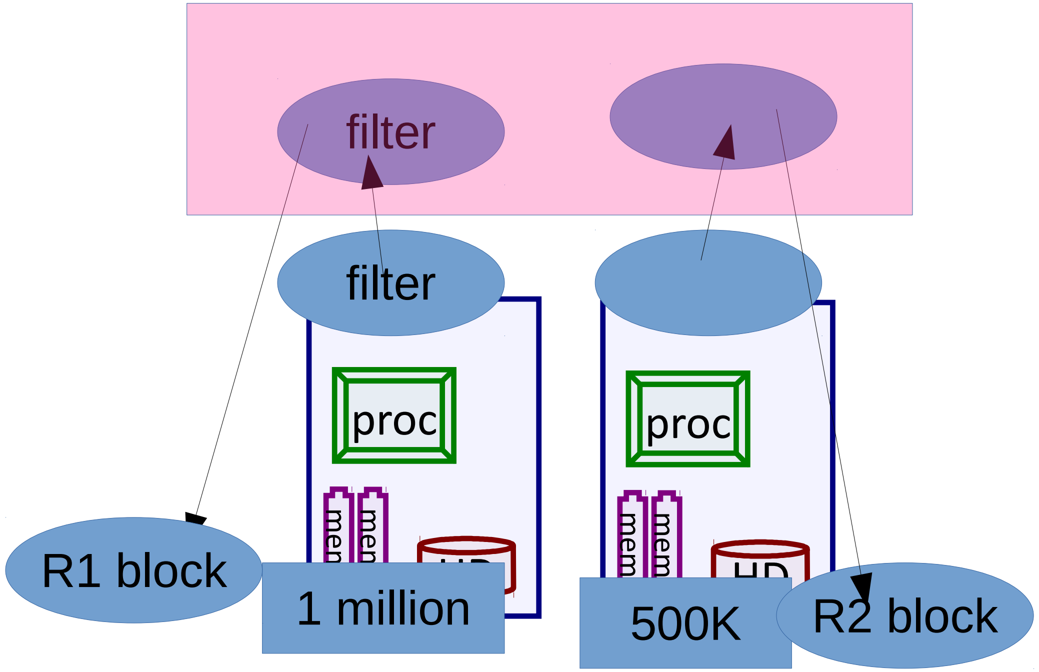
B2

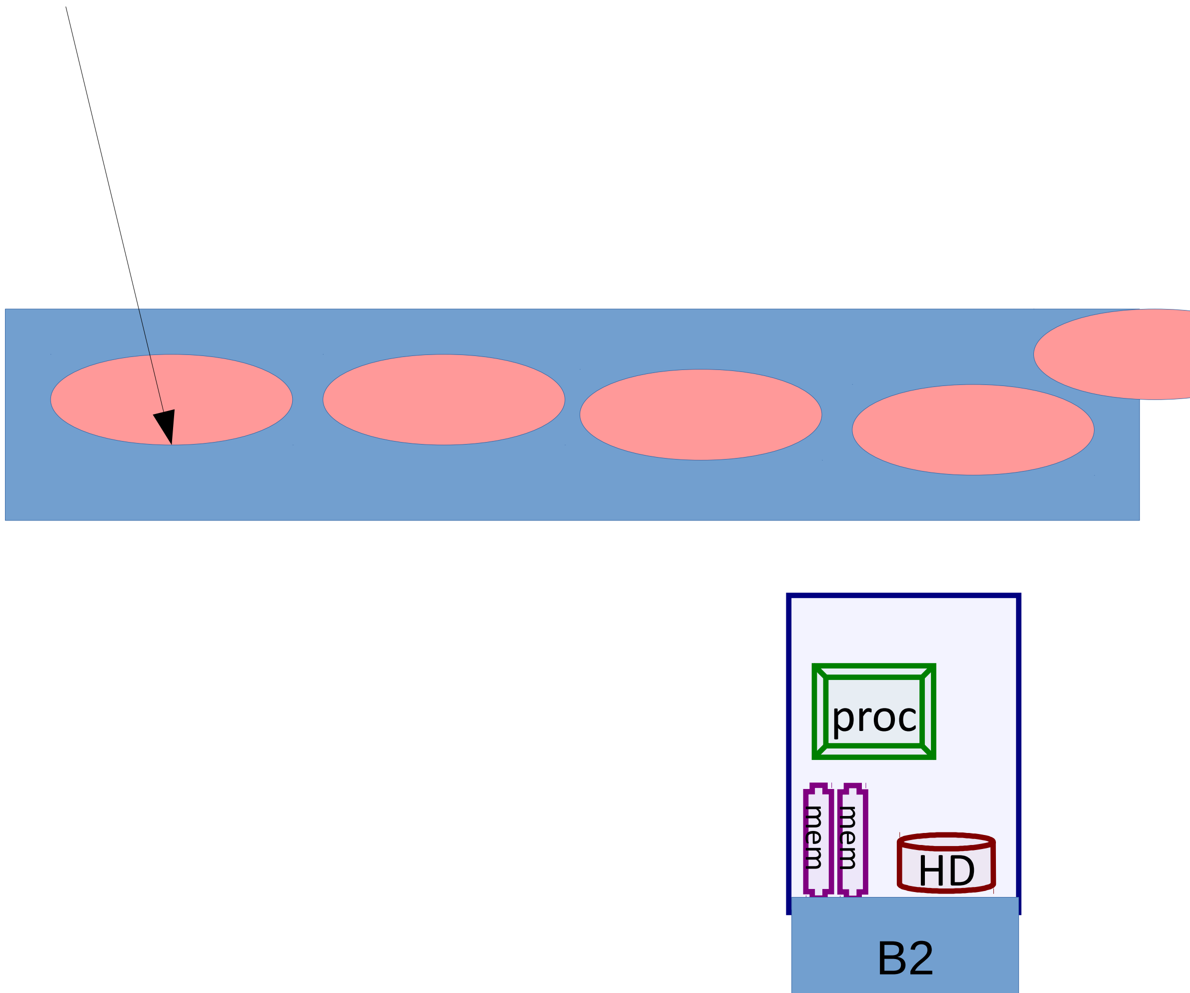
1 million

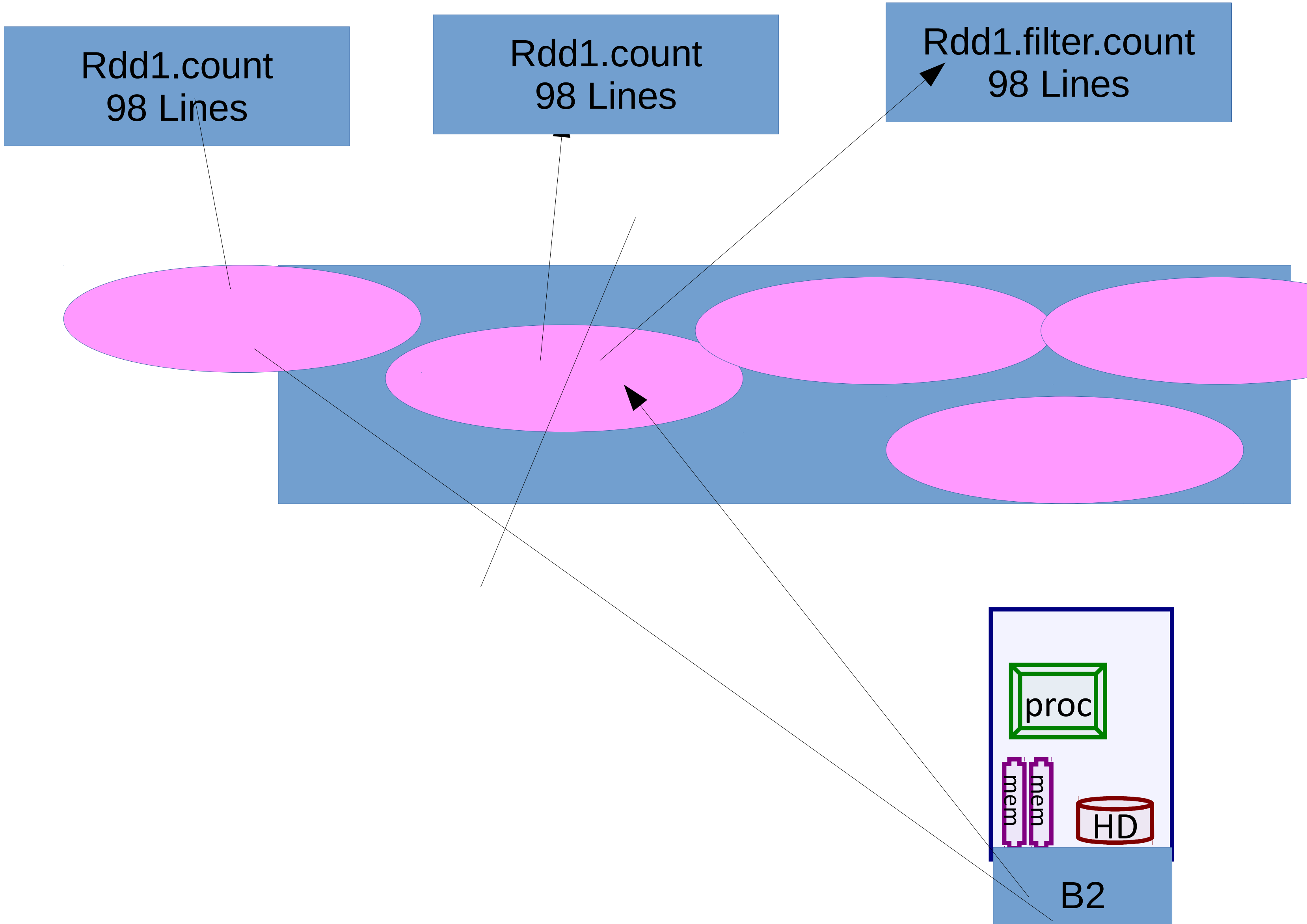
500K

Read fileA
FilteredValue = Filter FileA only errors
Write Filtered Value as restul.txt

Result.txt - > R1 and R2
R1 – N1, N2, N4
R2 – N4, N3, N1




proc
mem
mem
HD
B2



Rdd1.count
98 Lines

Rdd1.count
98 Lines

Rdd1.filter.count
98 Lines

100
100
100
10
10

100
100
100

10
10

64

100 - 3

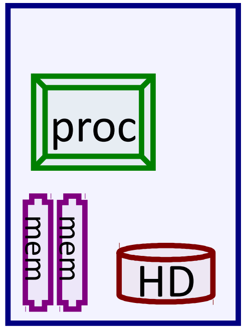
10 -2

100
10

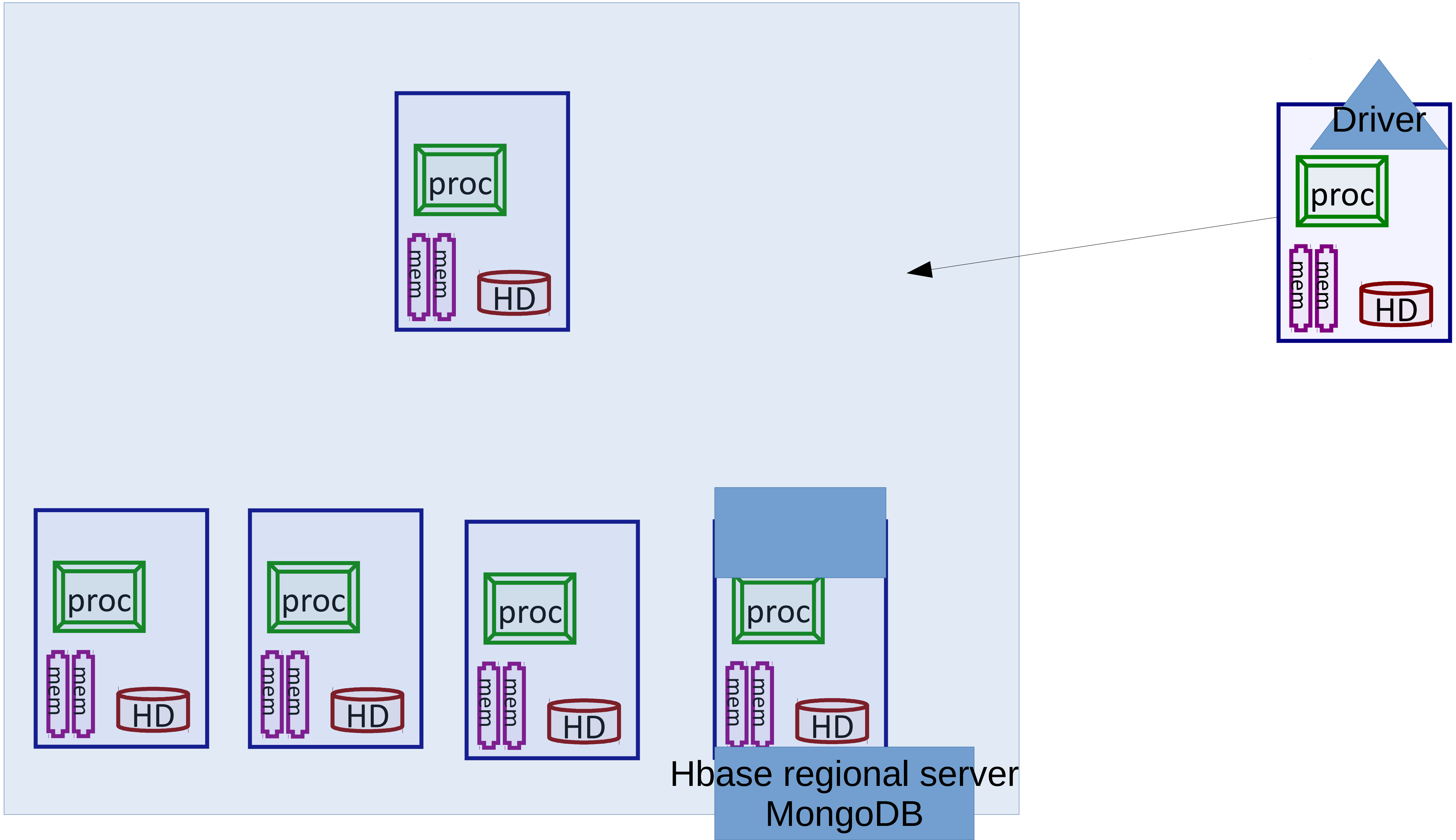
(100 *3)+
(10 * 2)

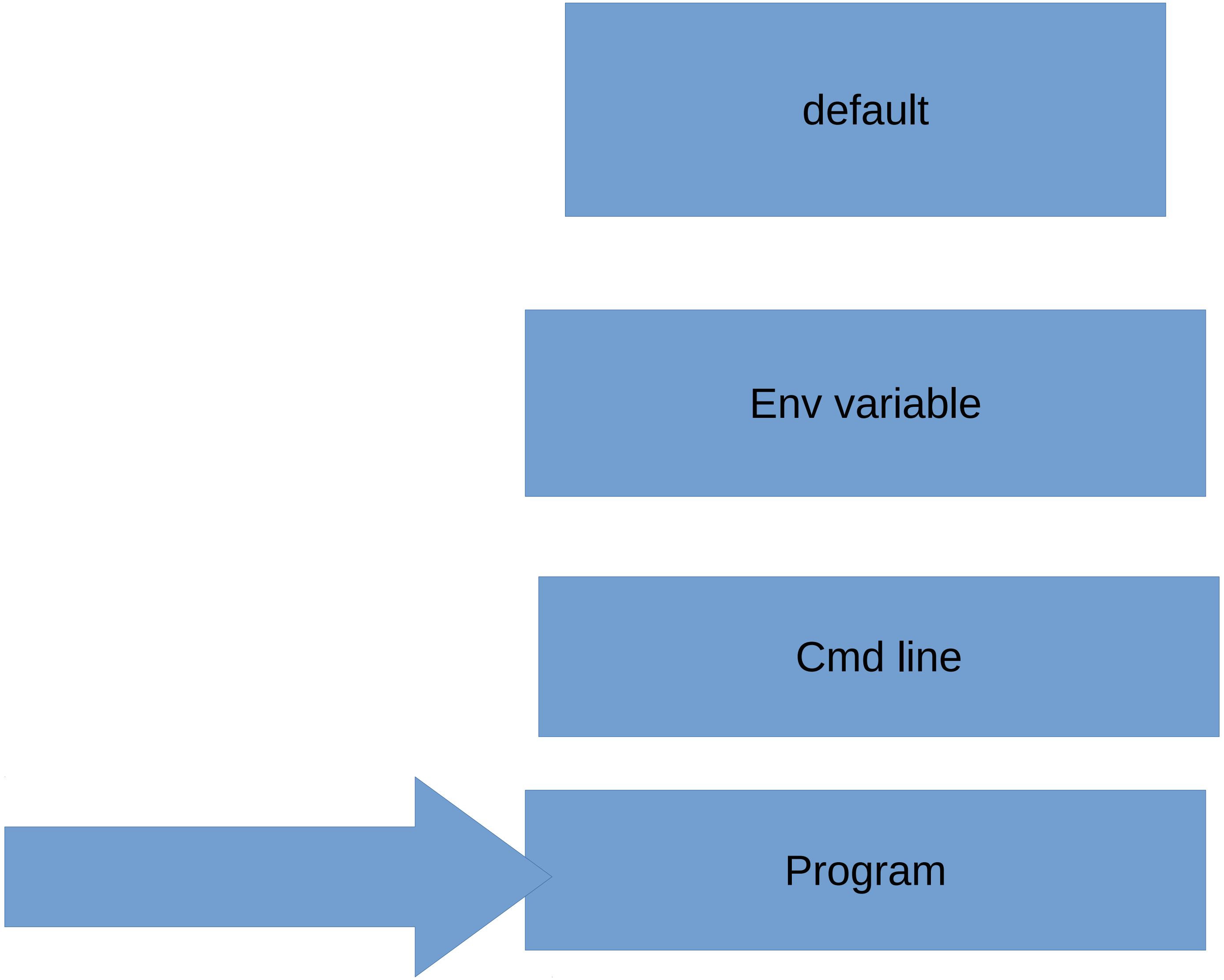
320/5

55


proc
mem
mem
HD

Mesos / Yarn / Spark Cluster



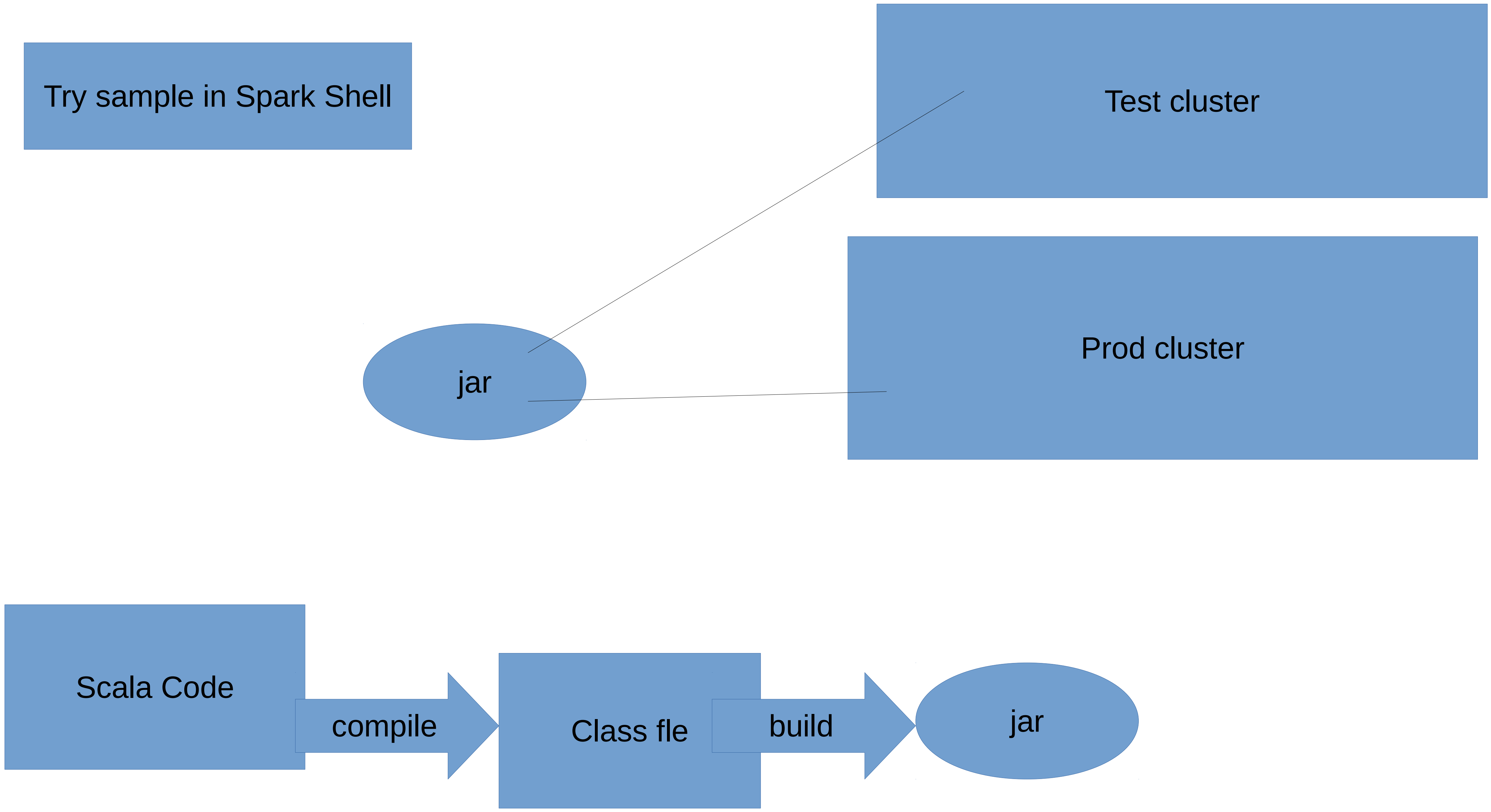


default

Env variable

Cmd line

Program



Try sample in Spark Shell

Test cluster

jar

Prod cluster

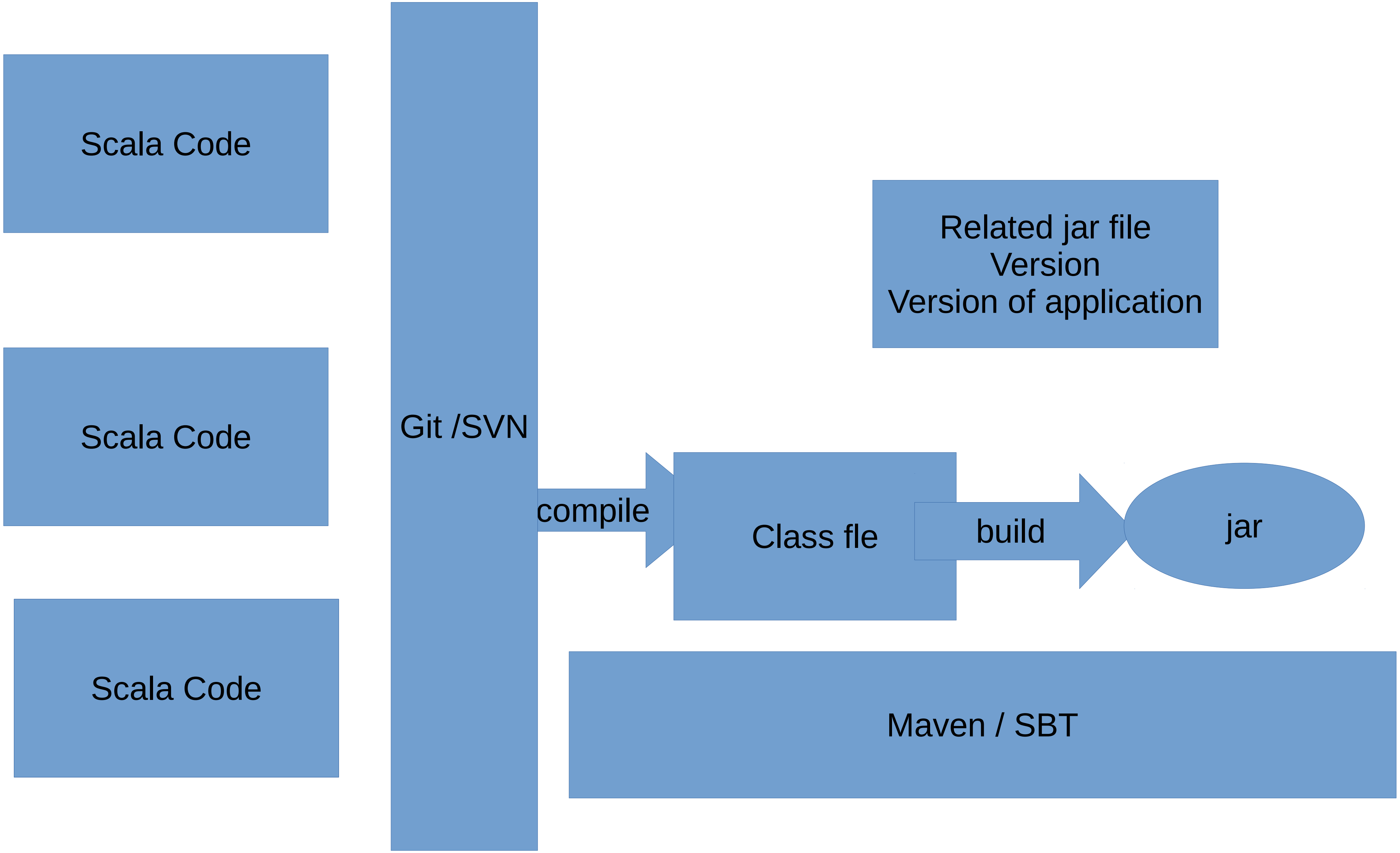
Scala Code

compile

Class fle

build

jar



Scala Code

Scala Code

Scala Code

Git /SVN

compile

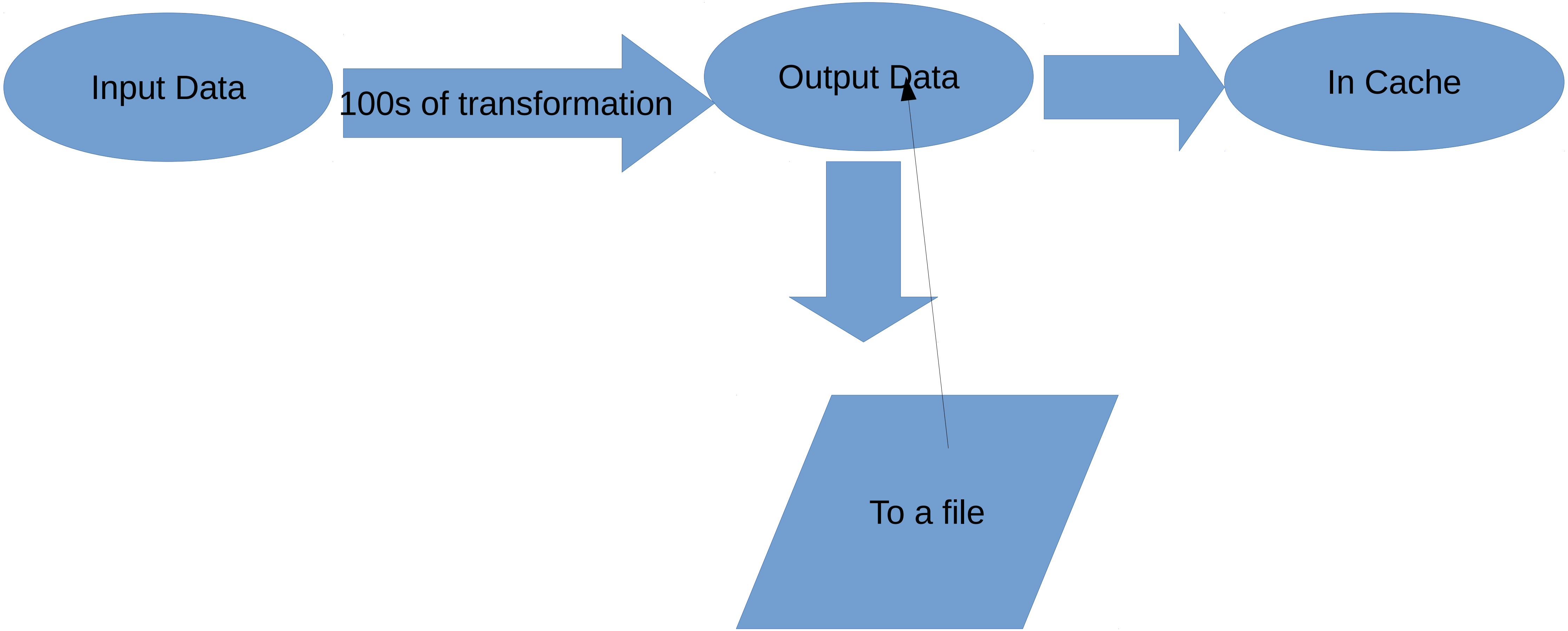
Class fle

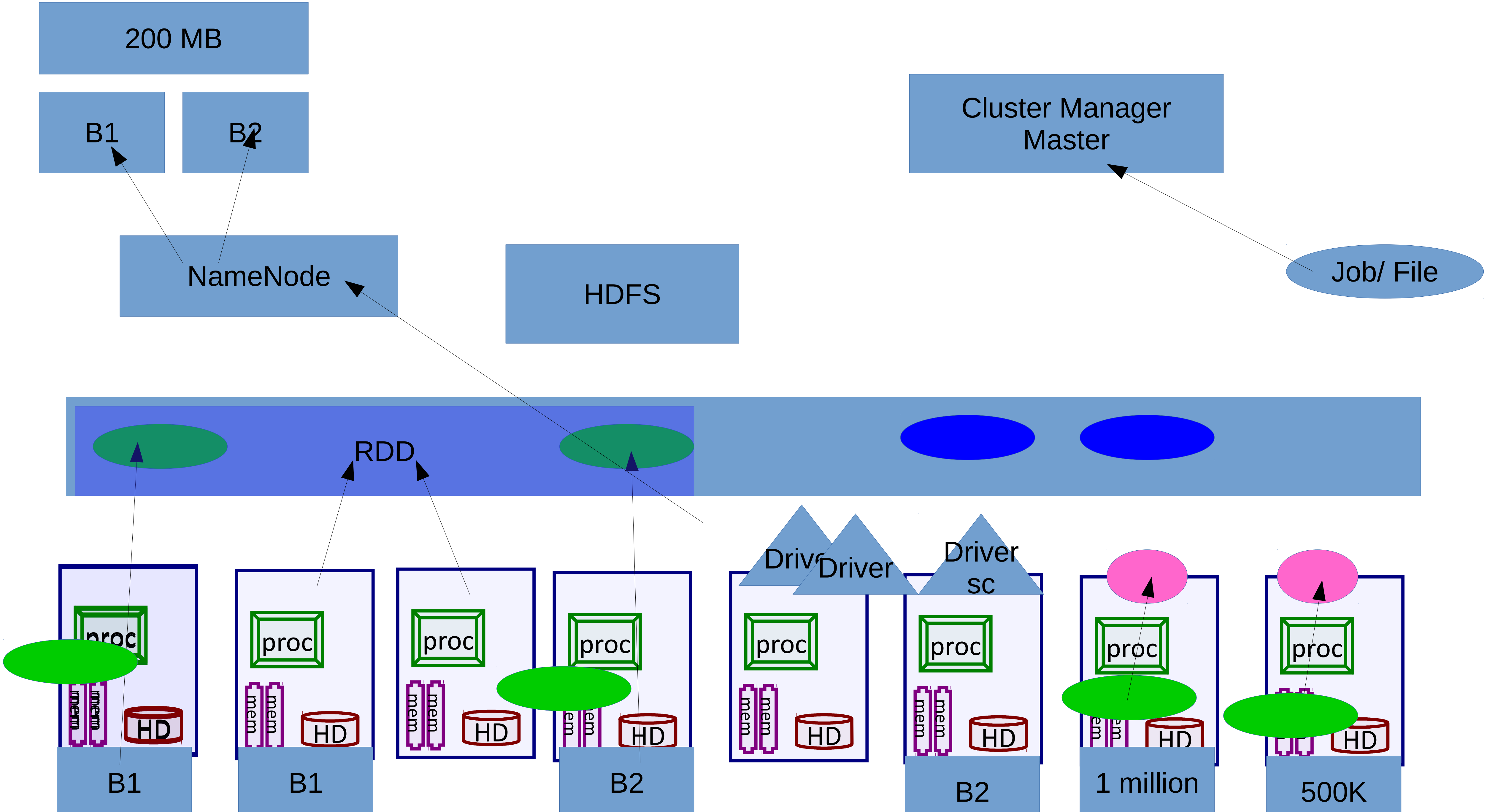
build

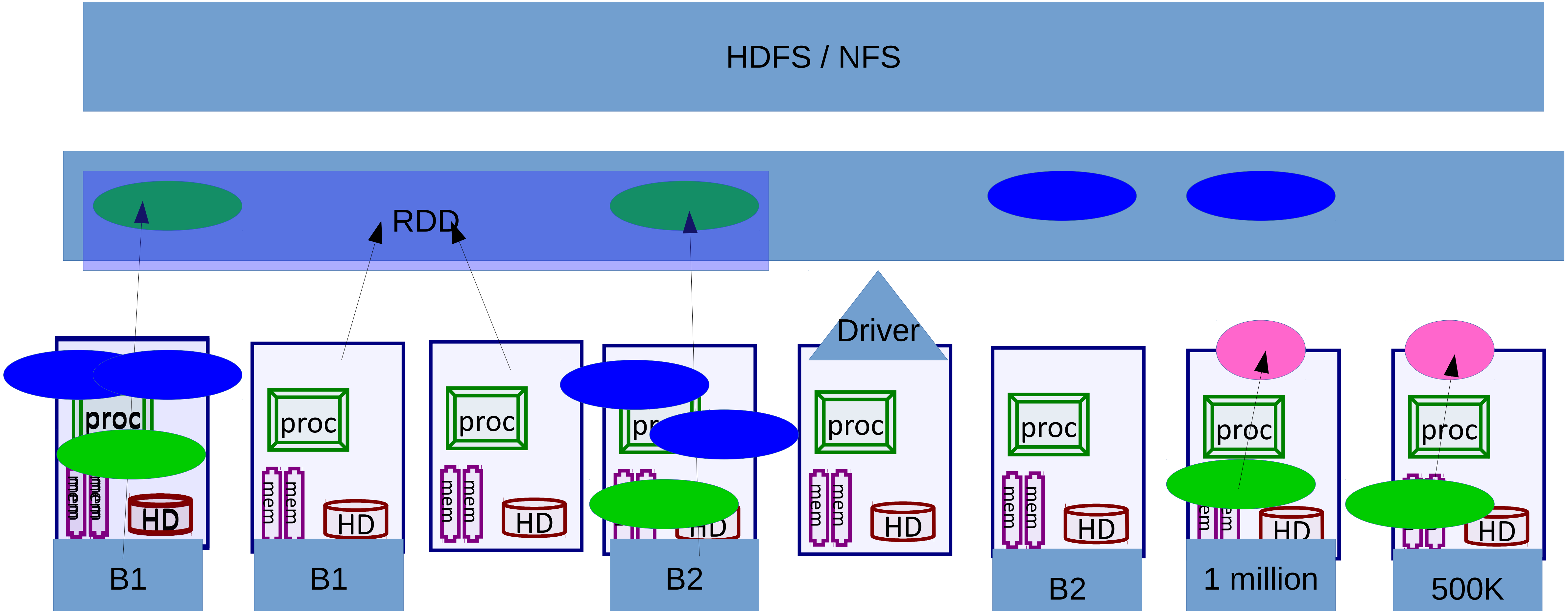
jar

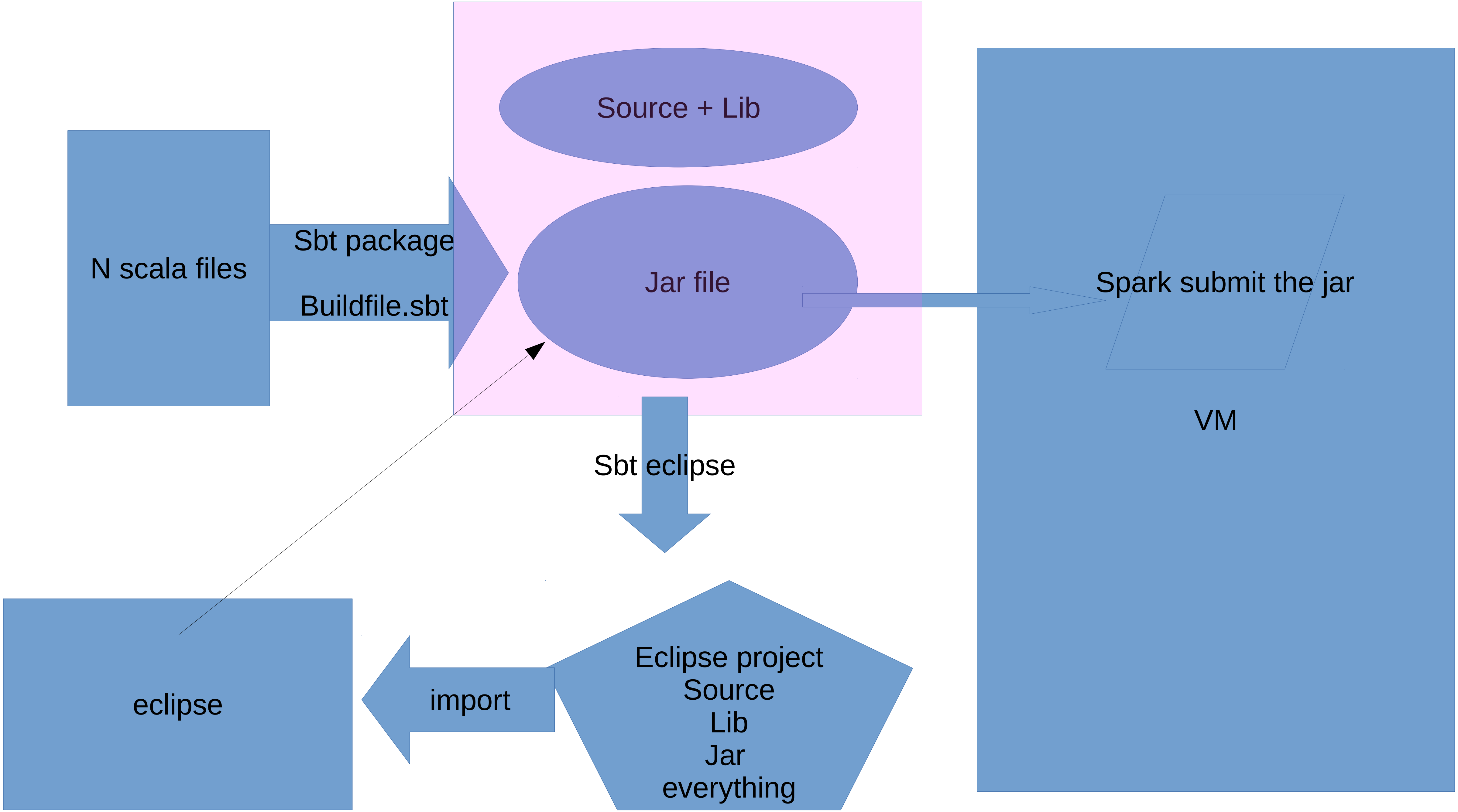
Related jar file
Version
Version of application

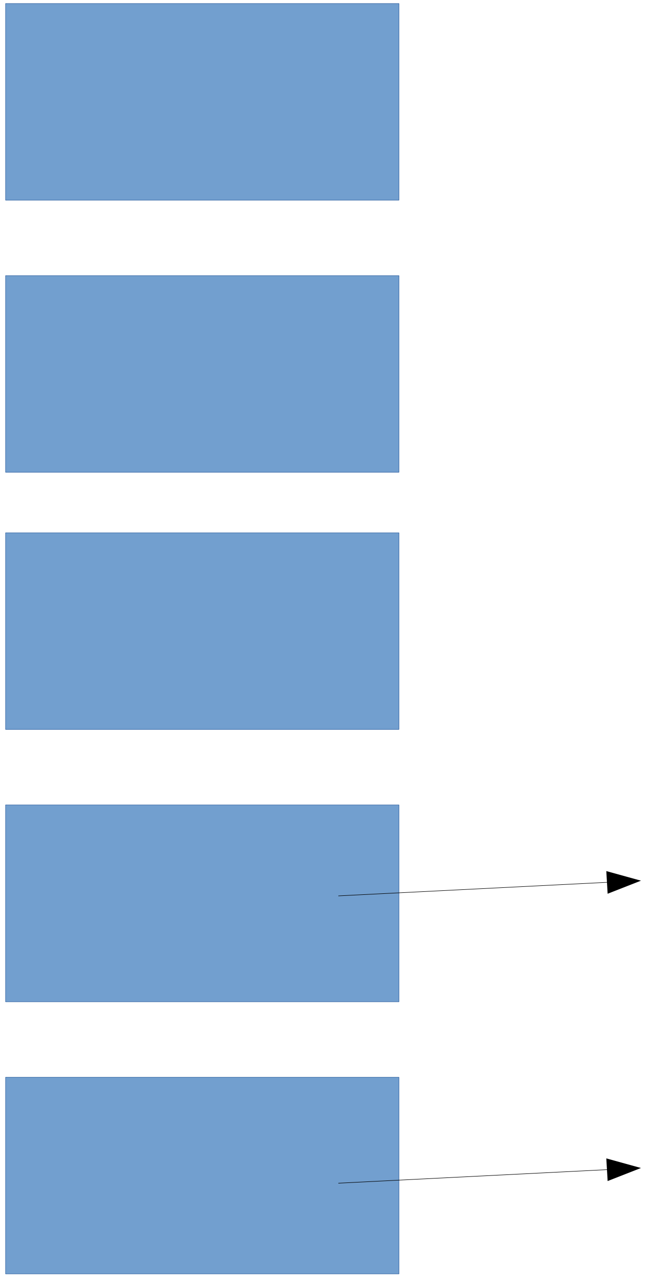
Maven / SBT


Input Data
100s of transformation
Output Data
In Cache
To a file


200 MB
B1
B2
Cluster Manager Master
NameNode
HDFS
Job/ File
RDD
Driv
Driver
Driver sc
proc
proc
proc
proc
proc
proc
proc
proc
mem
mem
HD
HD
HD
HD
HD
HD
HD
HD
B1
B1
B2
B2
1 million
500K

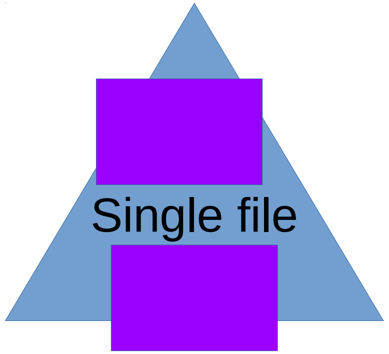

HDFS / NFS
RDD
Driver
proc
mem
mem
HD
B1
B1
B2
B2
1 million
500K


Source + Lib
N scala files
Sbt package
Buildfile.sbt
Jar file
Spark submit the jar
VM
Sbt eclipse
Eclipse project
Source
Lib
Jar
everything
import
eclipse



Name Node
Meta data in memory

Metadata
File1 – B1, B2
B1 – S1, S4, S5
B2 – S2, S3, S4
File2 – B3
B3 – S4,S5,S6



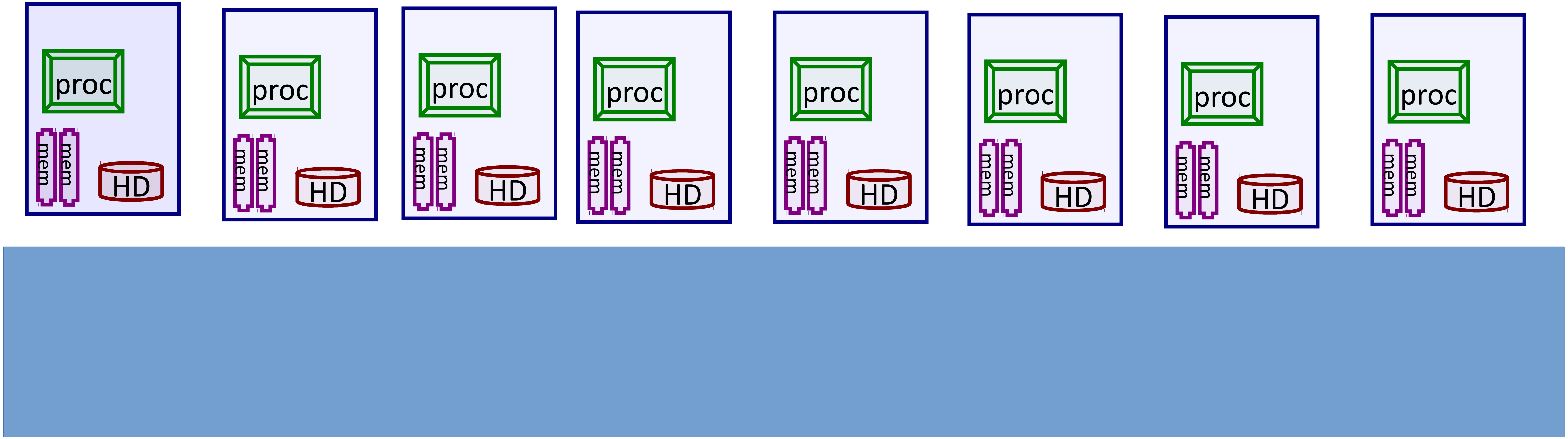
Normal Image

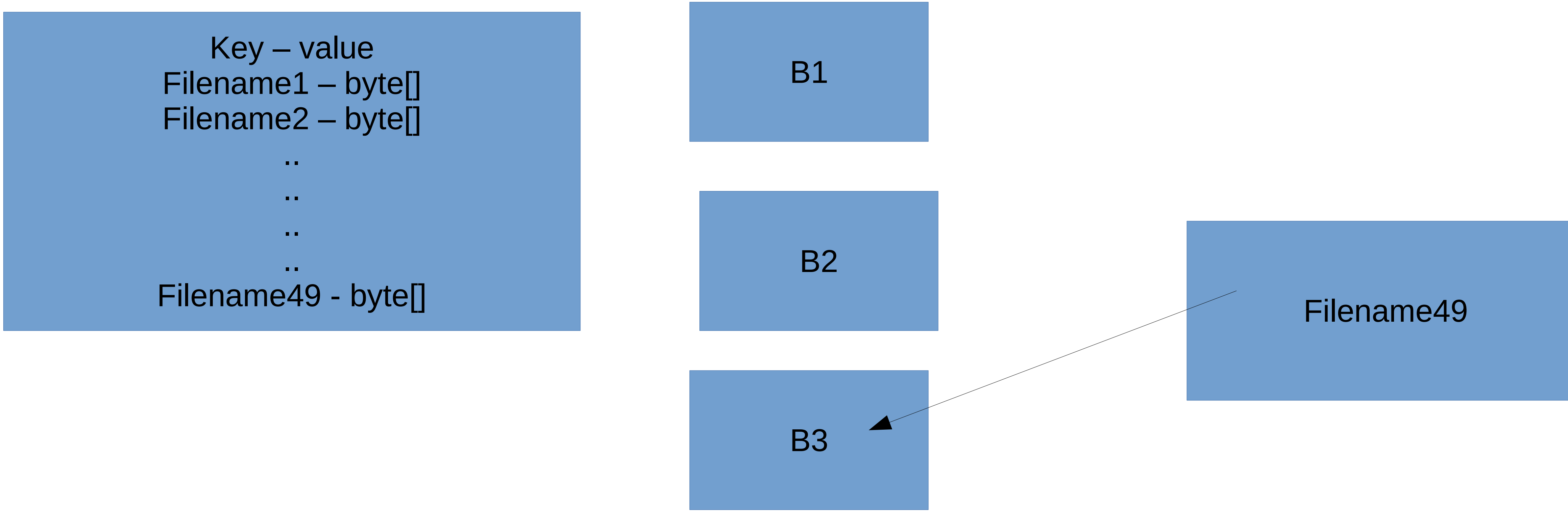
Issue Image

1 million CT images

Let meta data storage

Sequence file





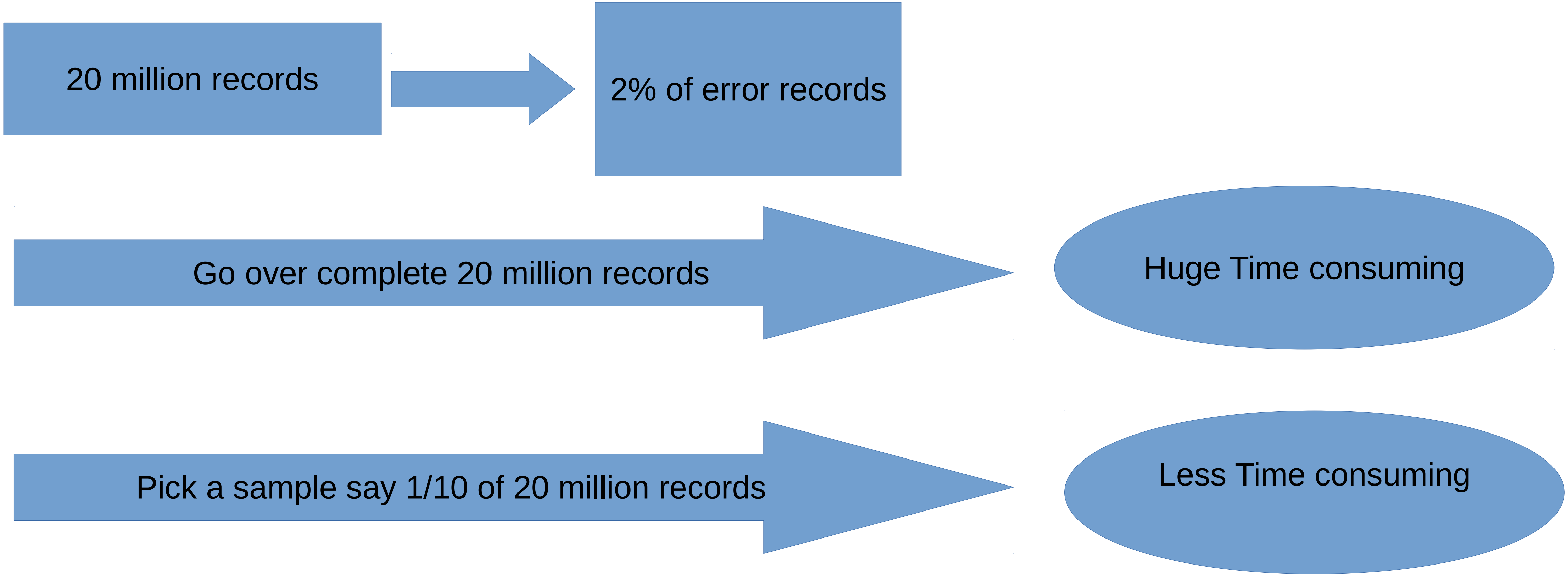
Key – value
Filename1 – byte[]
Filename2 – byte[]
..
..
..
..
Filename49 - byte[]

B1

B2

B3

Filename49



20 million records

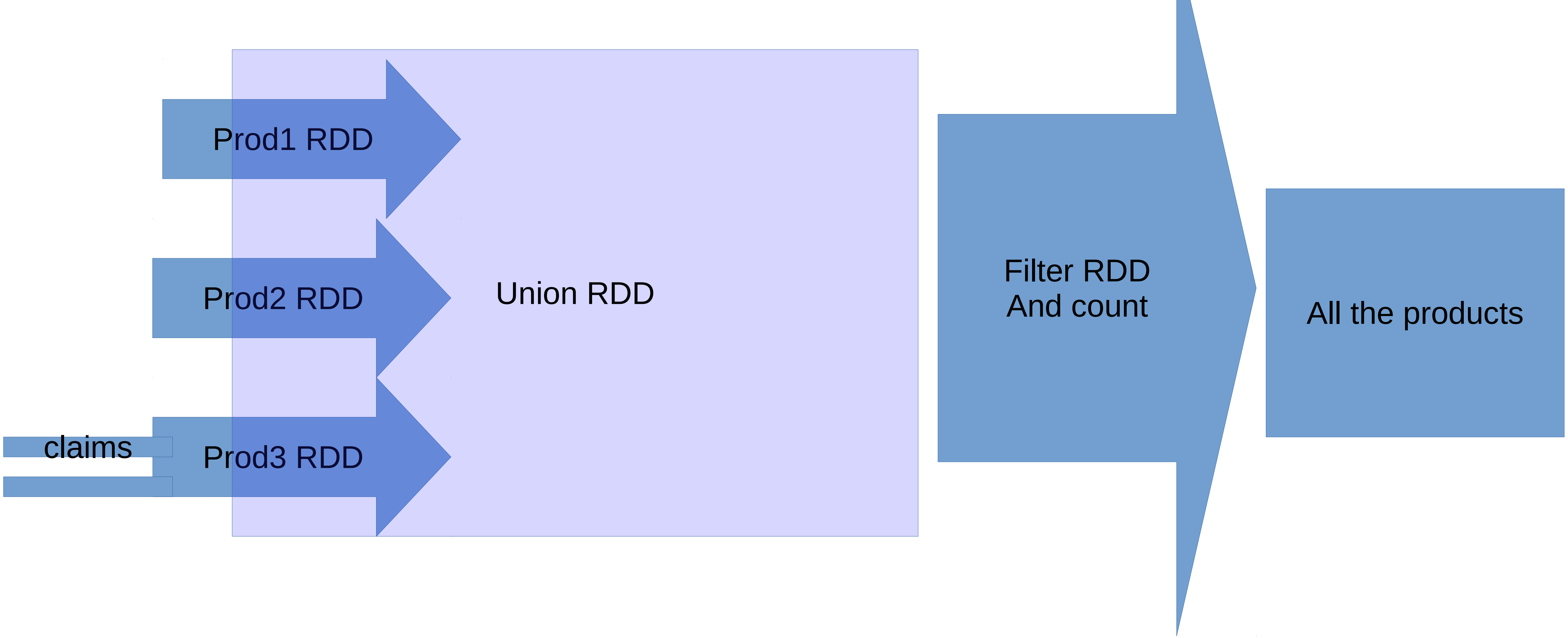
2% of error records

Go over complete 20 million records

Huge Time consuming

Pick a sample say 1/10 of 20 million records

Less Time consuming



claims

Prod1 RDD

Prod2 RDD

Prod3 RDD

Union RDD

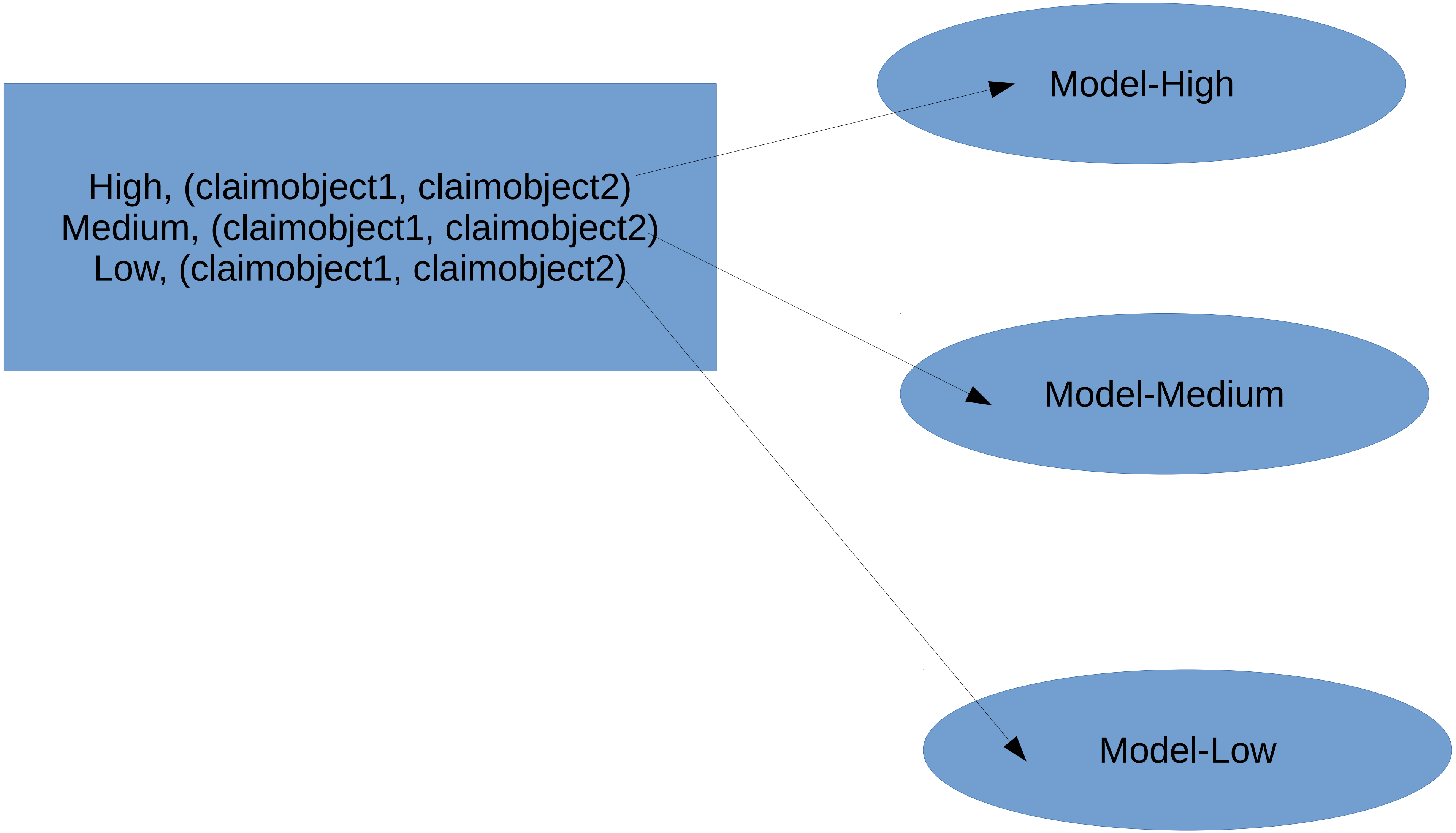
Filter RDD
And count

All the products

Calaim forms

(high, claimobject)
(high, claimobject)
(Low, claimobject)
(Low, claimobject)
(Medium, claimobject)

High, (claimobject1, claimobject2)
Medium, (claimobject1, claimobject2)
Low, (claimobject1, claimobject2)

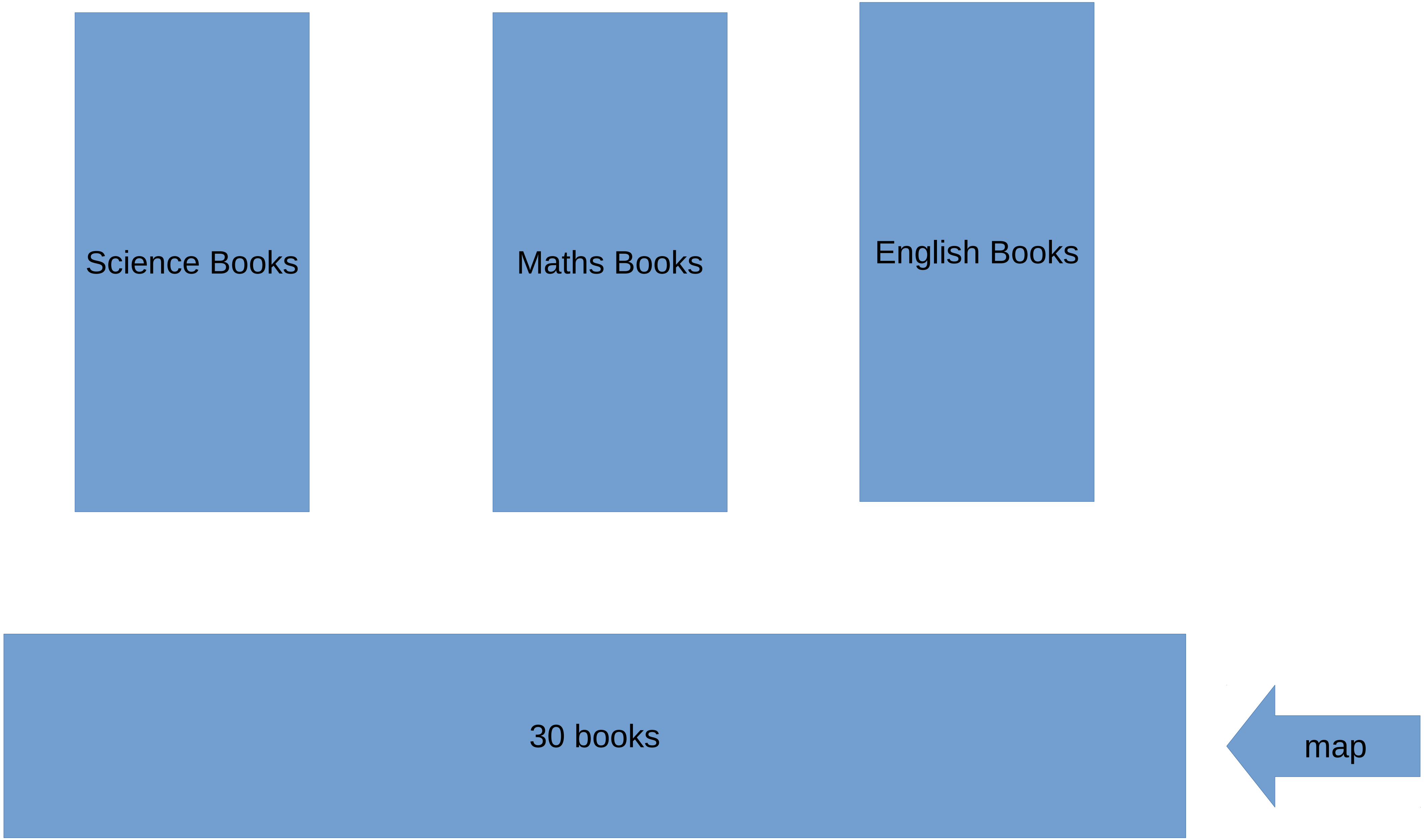


High, (claimobject1, claimobject2)
Medium, (claimobject1, claimobject2)
Low, (claimobject1, claimobject2)

Model-High

Model-Medium

Model-Low



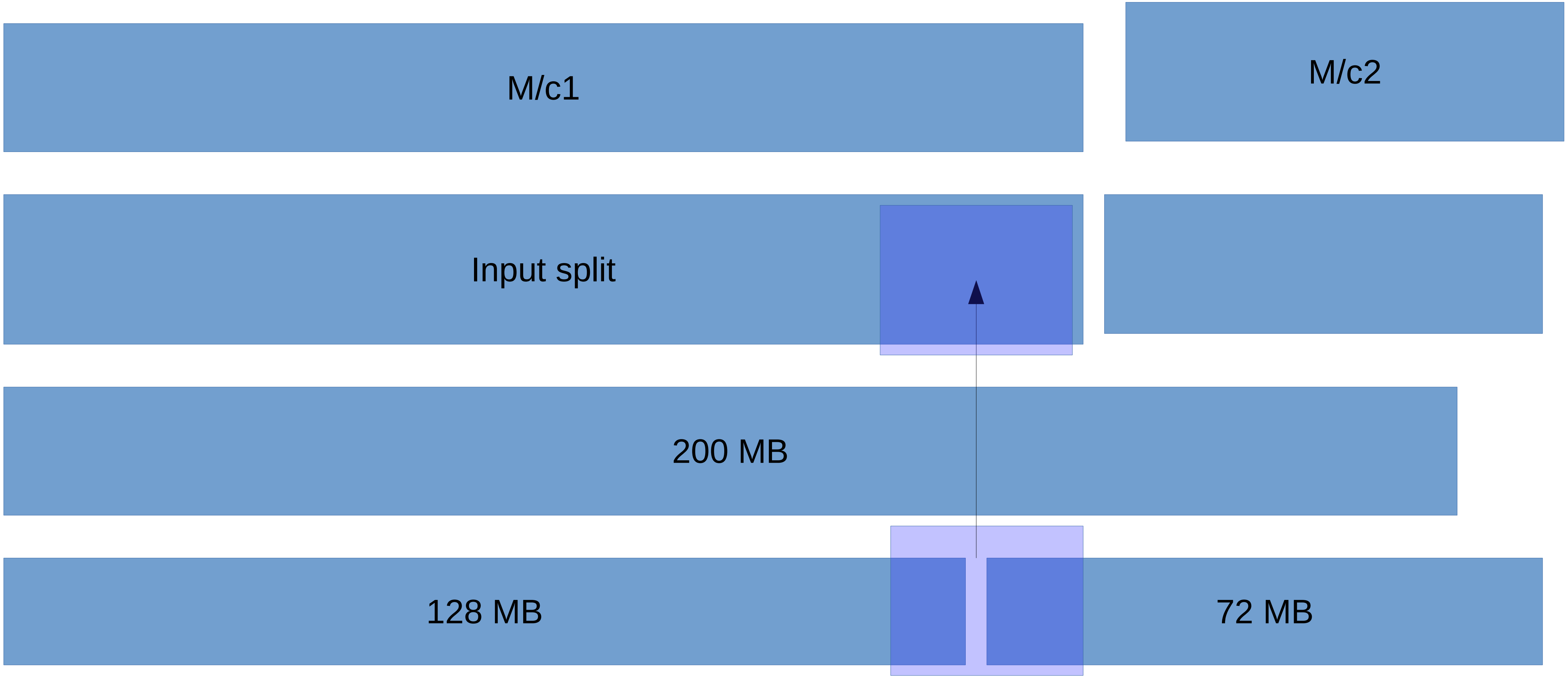
Science Books

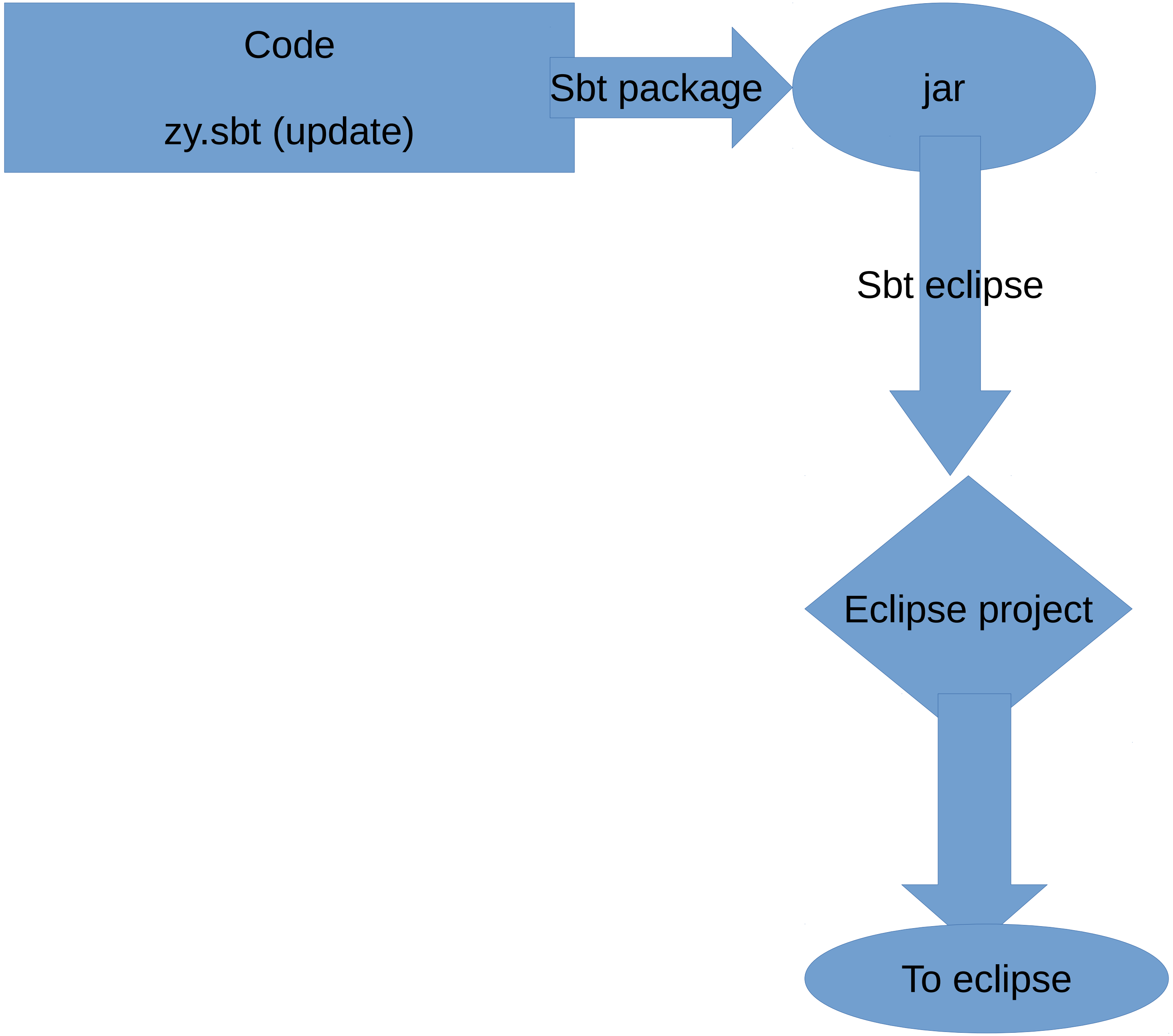
Maths Books

English Books

30 books

map


M/c1
M/c2
Input split
200 MB
128 MB
72 MB



Code

zy.sbt (update)

Sbt package

jar

Sbt eclipse

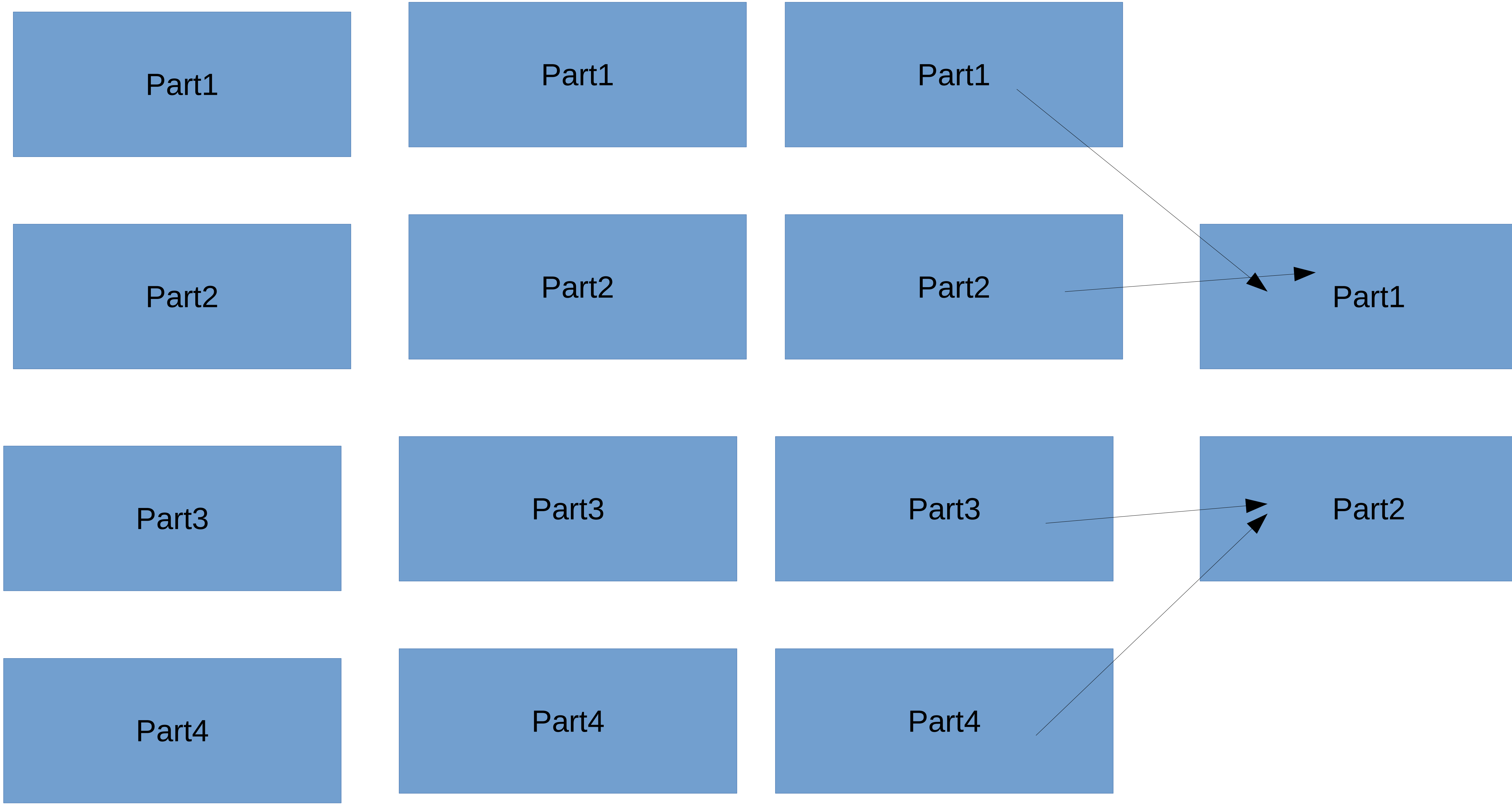
Eclipse project

To eclipse

Email as key (details about facebook wall)

Email as key (twitter comments)

Email as key (org internal performance mat)



Part1

Part2

Part3

Part4

Part1

Part2

Part3

Part4

Part1

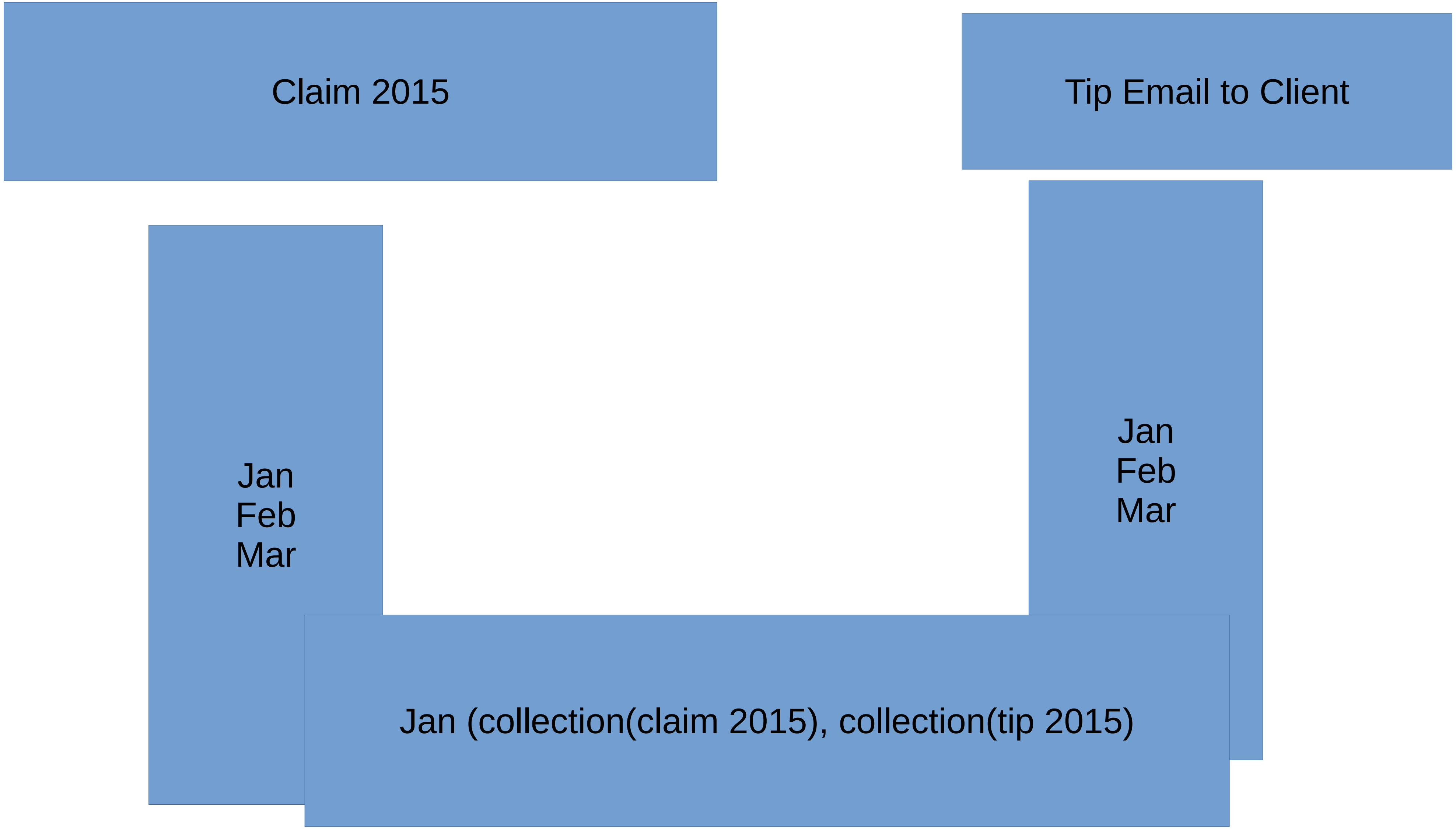
Part2

Part3

Part4

Part1

Part2



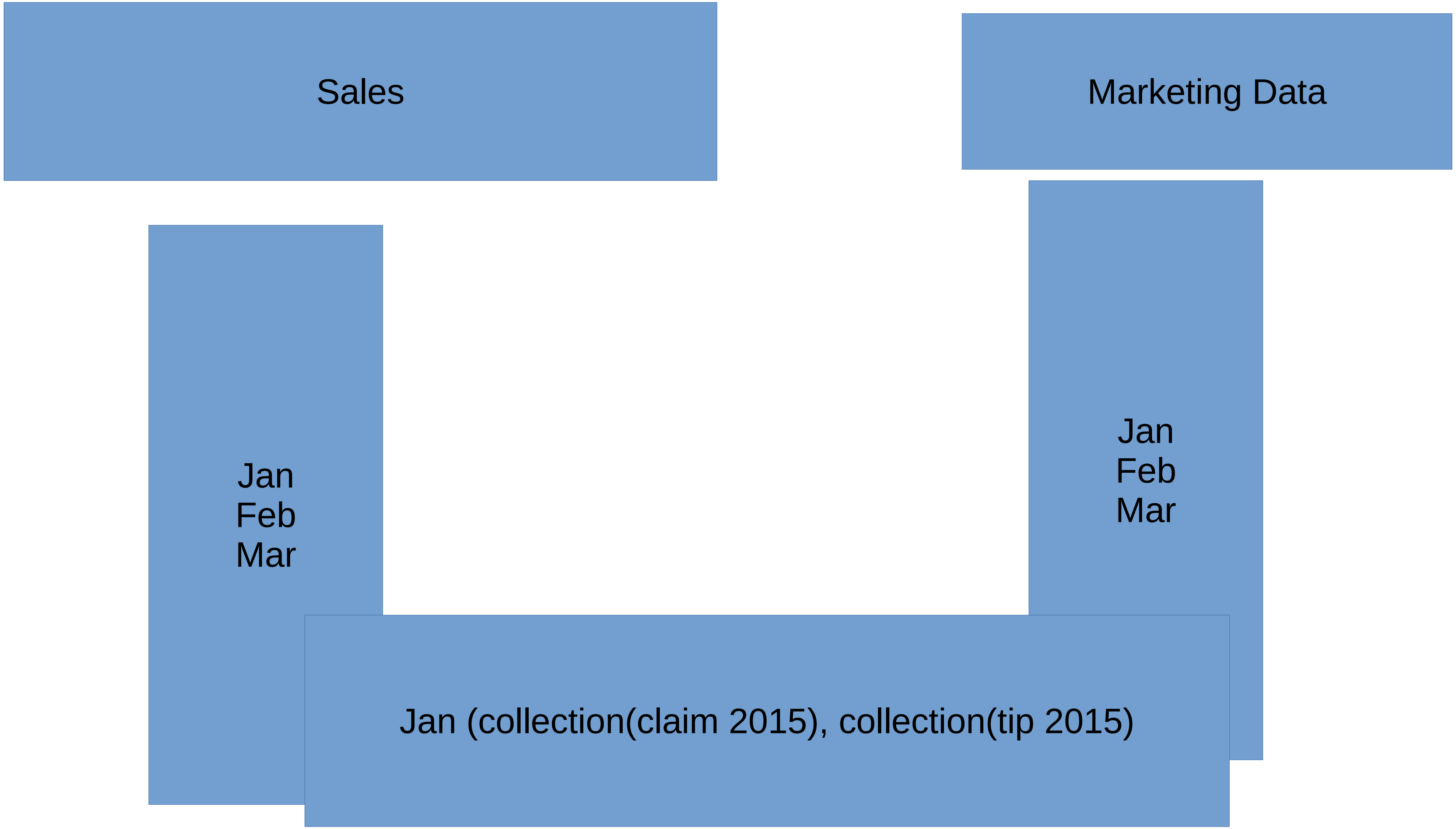
Claim 2015

Tip Email to Client

Jan
Feb
Mar

Jan
Feb
Mar

Jan (collection(claim 2015), collection(tip 2015)



Sales

Marketing Data

Jan
Feb
Mar

Jan
Feb
Mar

Jan (collection(claim 2015), collection(tip 2015)

(1, "A"), (2, "B"), (2, "D"), (3, "C")

(1, "A"), (2, "D"), (3, "C")

(3, "A"), (3, "B"), (3, "A")

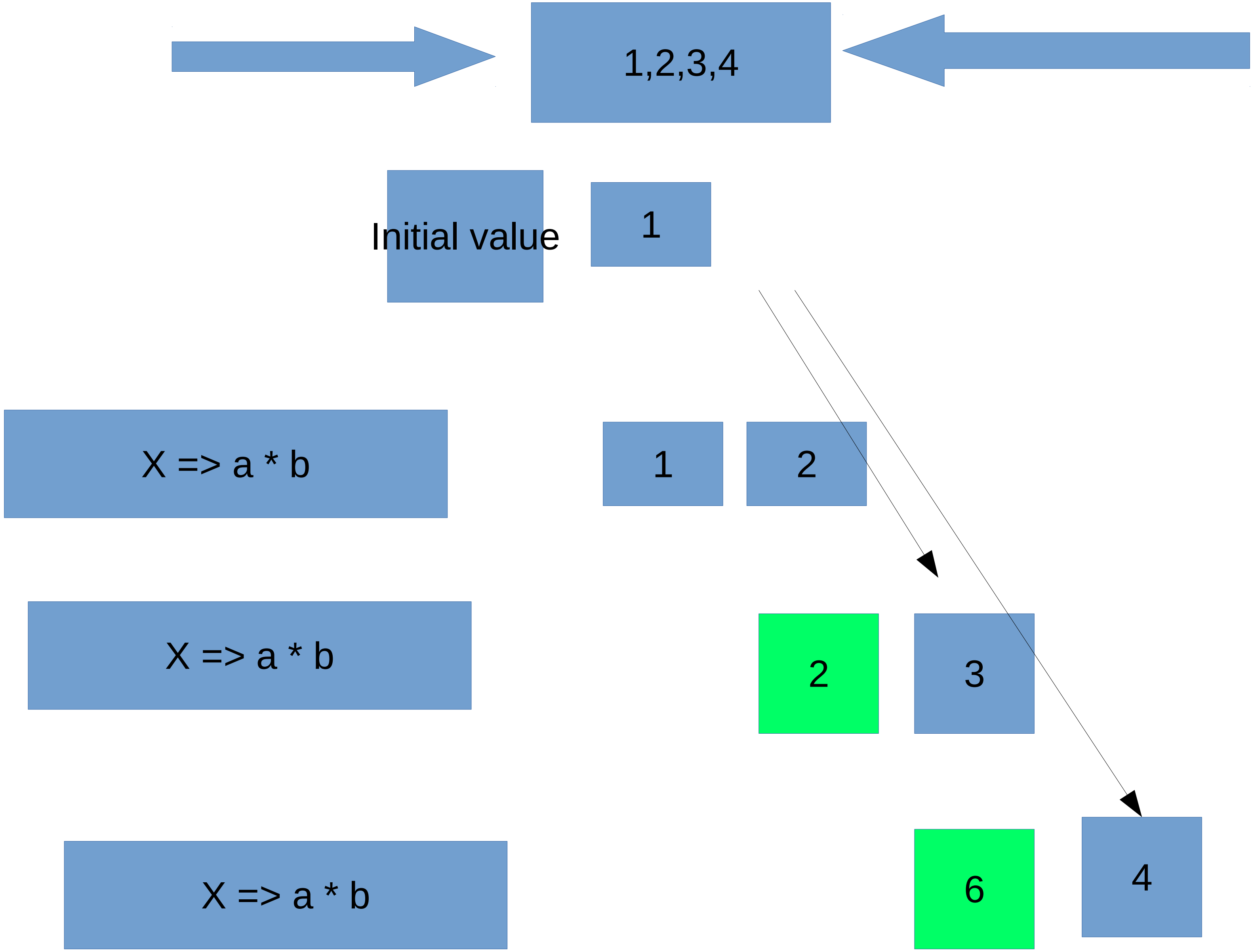
(3, "A")

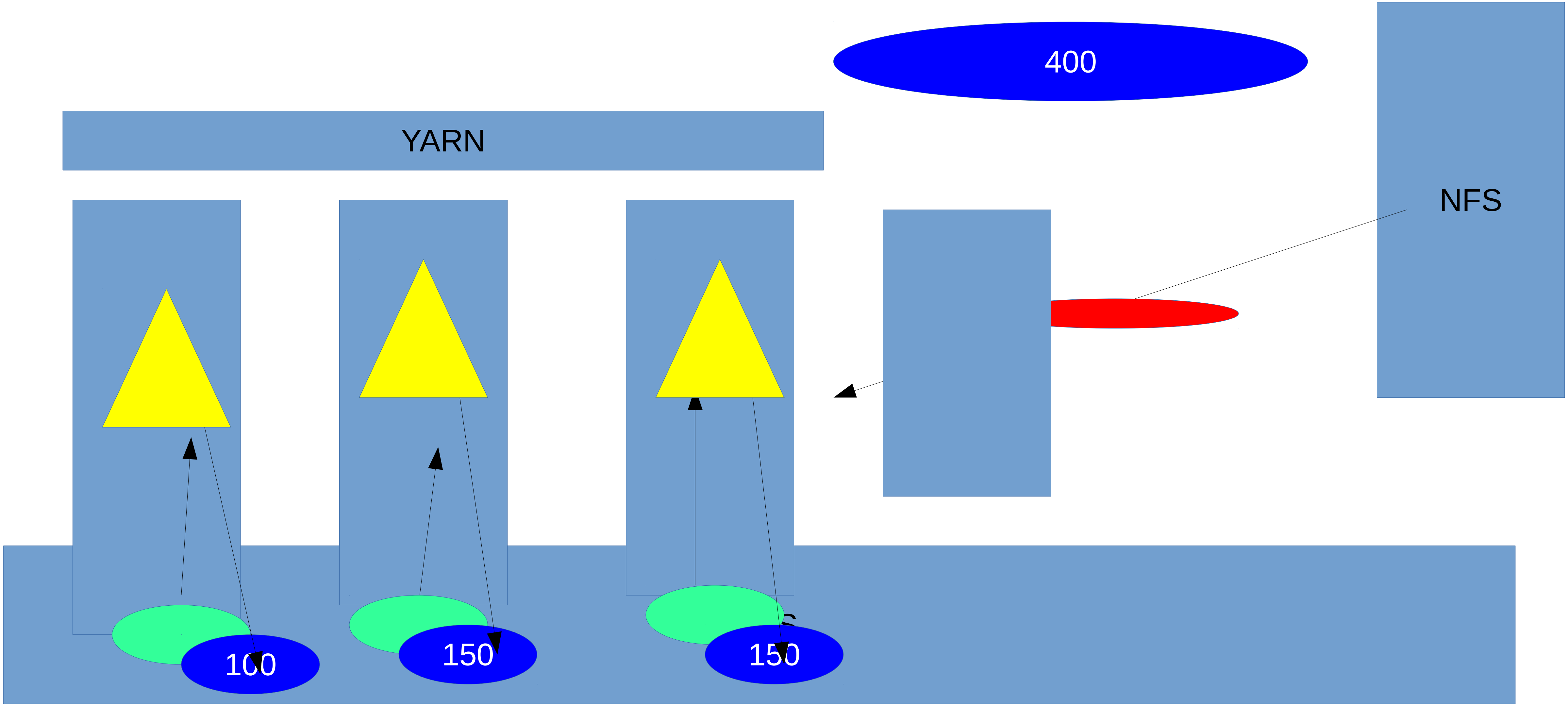
(1, "A"), (2, "D"), (3, "C")

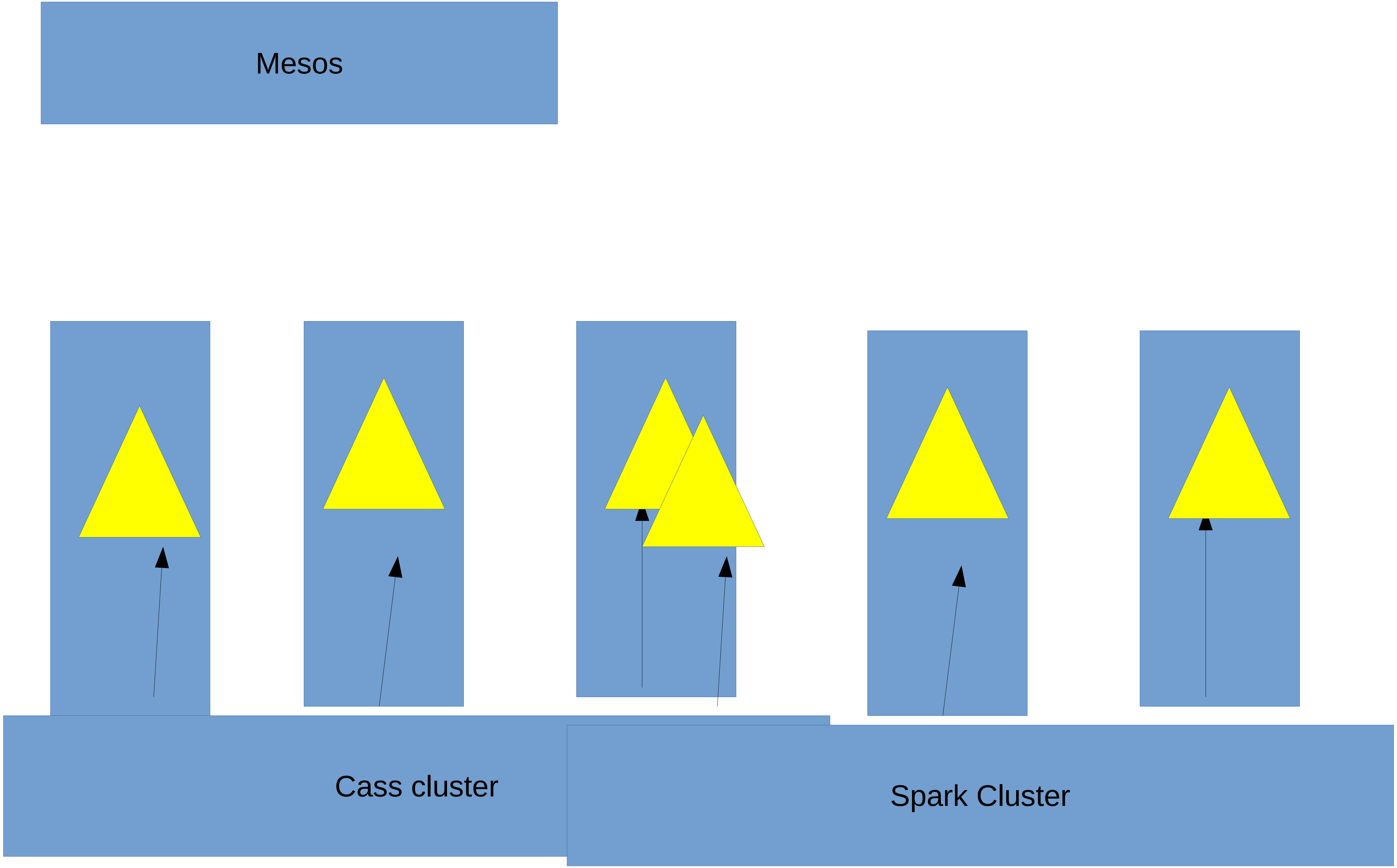
(1, "A"), (2, "D"), (3, "C"), (3, "A")

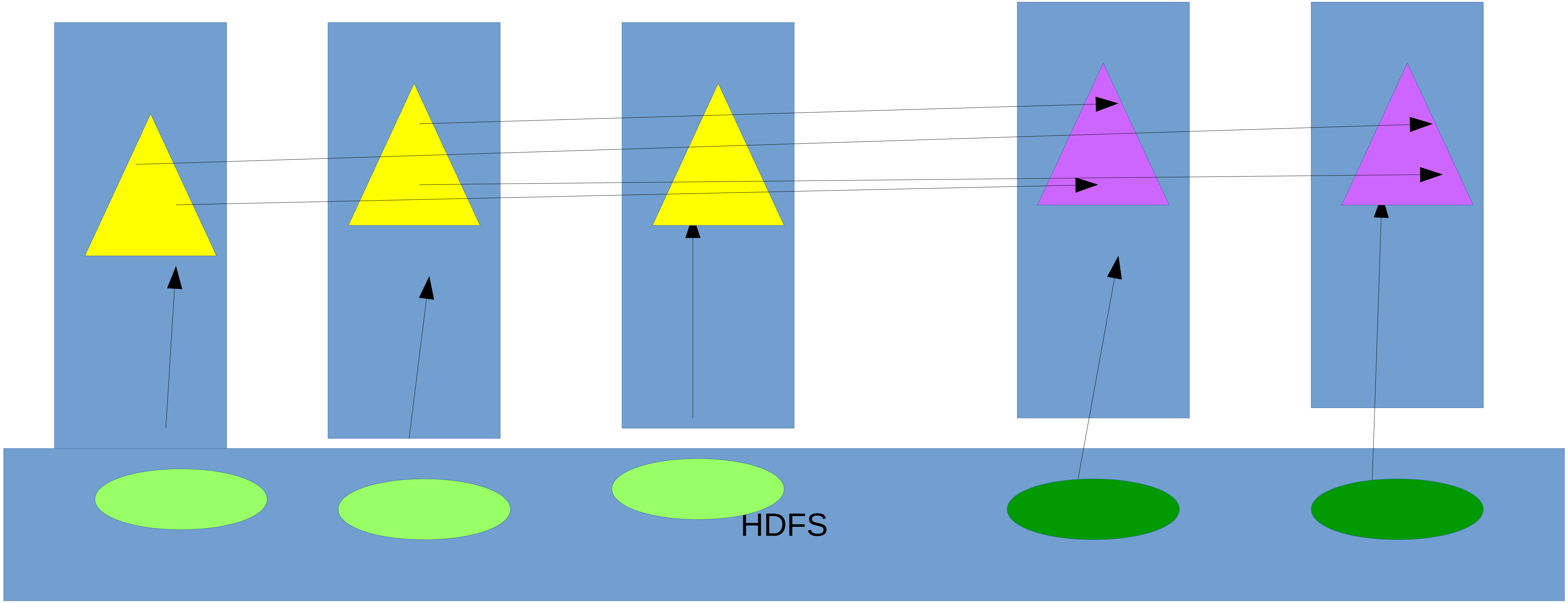
(3, "A")

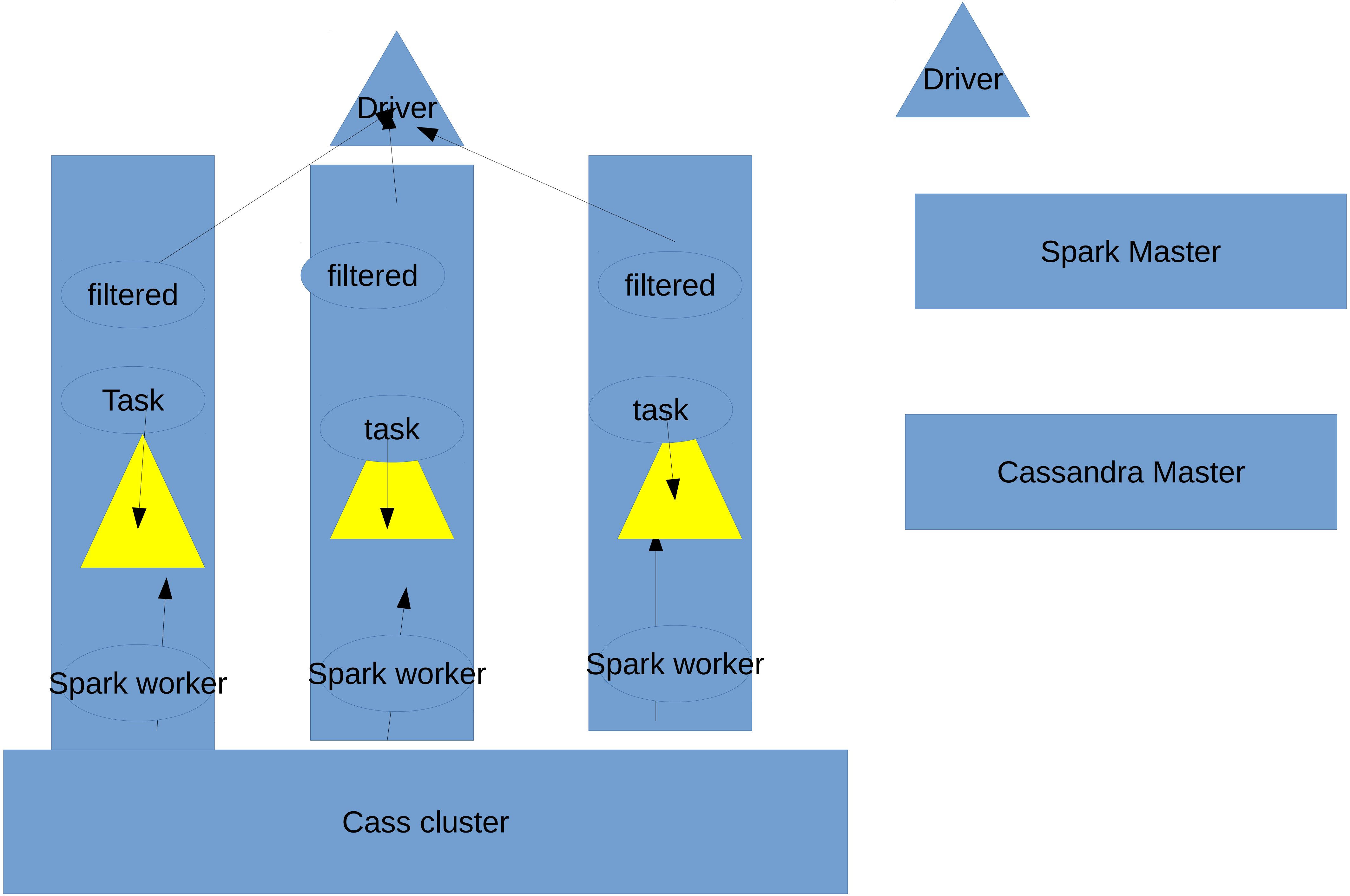
(3, "A"),(1, "A"), (2, "D"), (3, "C")


1,2,3,4
Initial value
1
X => a * b
1
2
X => a * b
2
3
X => a * b
6
4

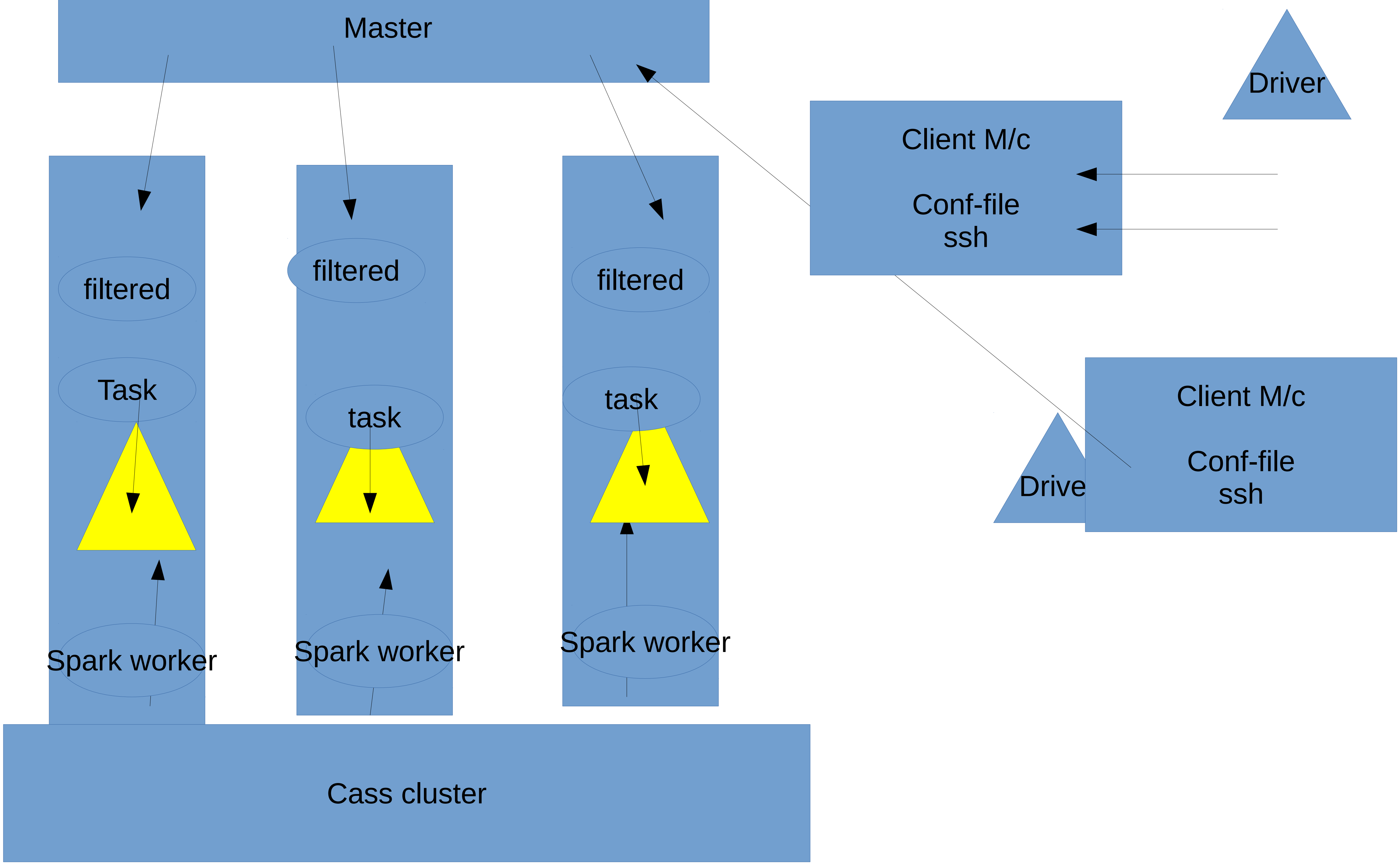

400
YARN
NFS
100
150
150


Mesos
Cass cluster
Spark Cluster


HDFS



Driver
Driver
Spark Master
Cassandra Master
filtered
filtered
filtered
Task
task
task
Spark worker
Spark worker
Spark worker
Cass cluster



Master

Driver

Client M/c

Conf-file
ssh

filtered

filtered

filtered

Task

task

task

Spark worker

Spark worker

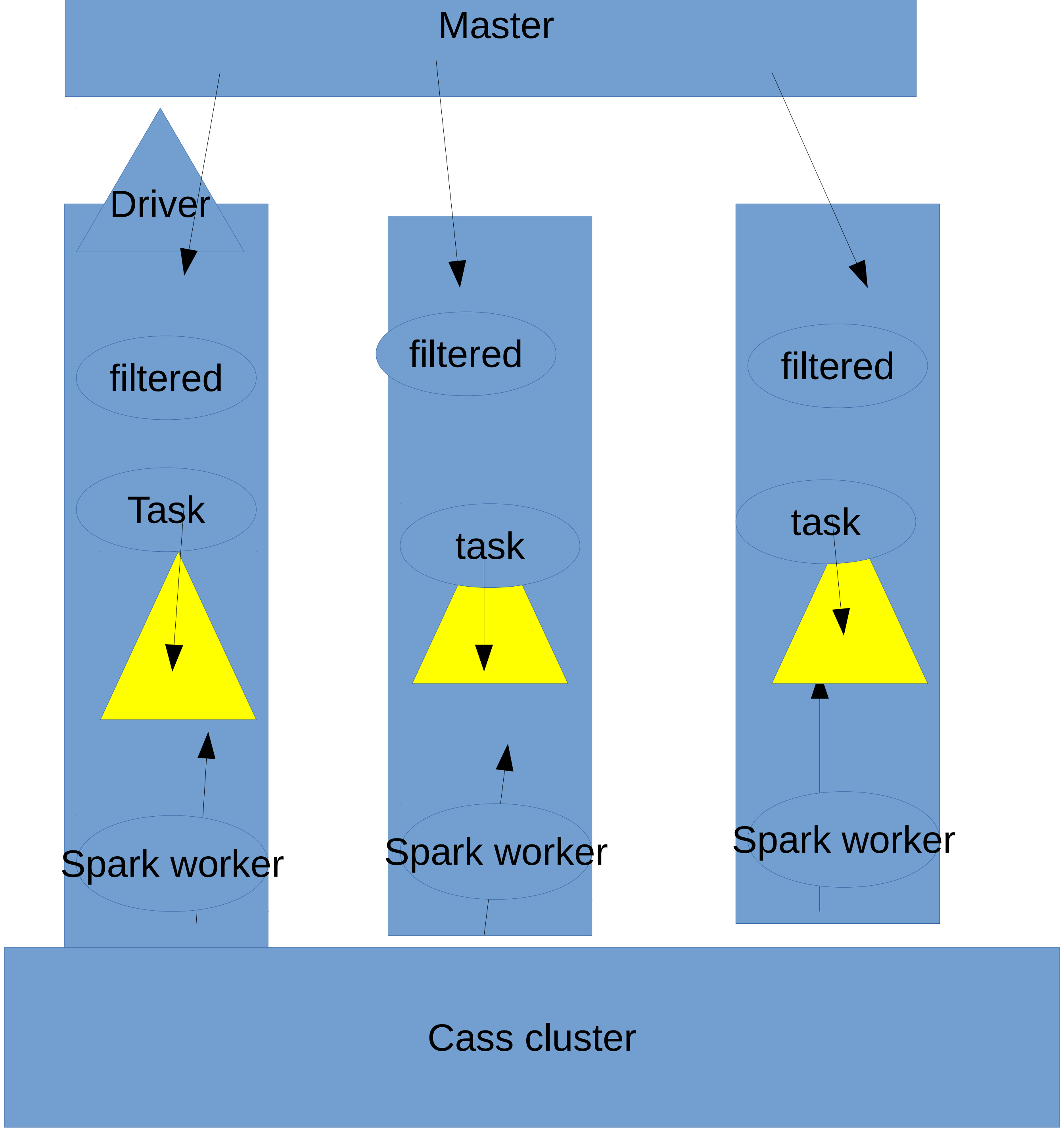
Spark worker

Drive

Client M/c

Conf-file
ssh

Cass cluster

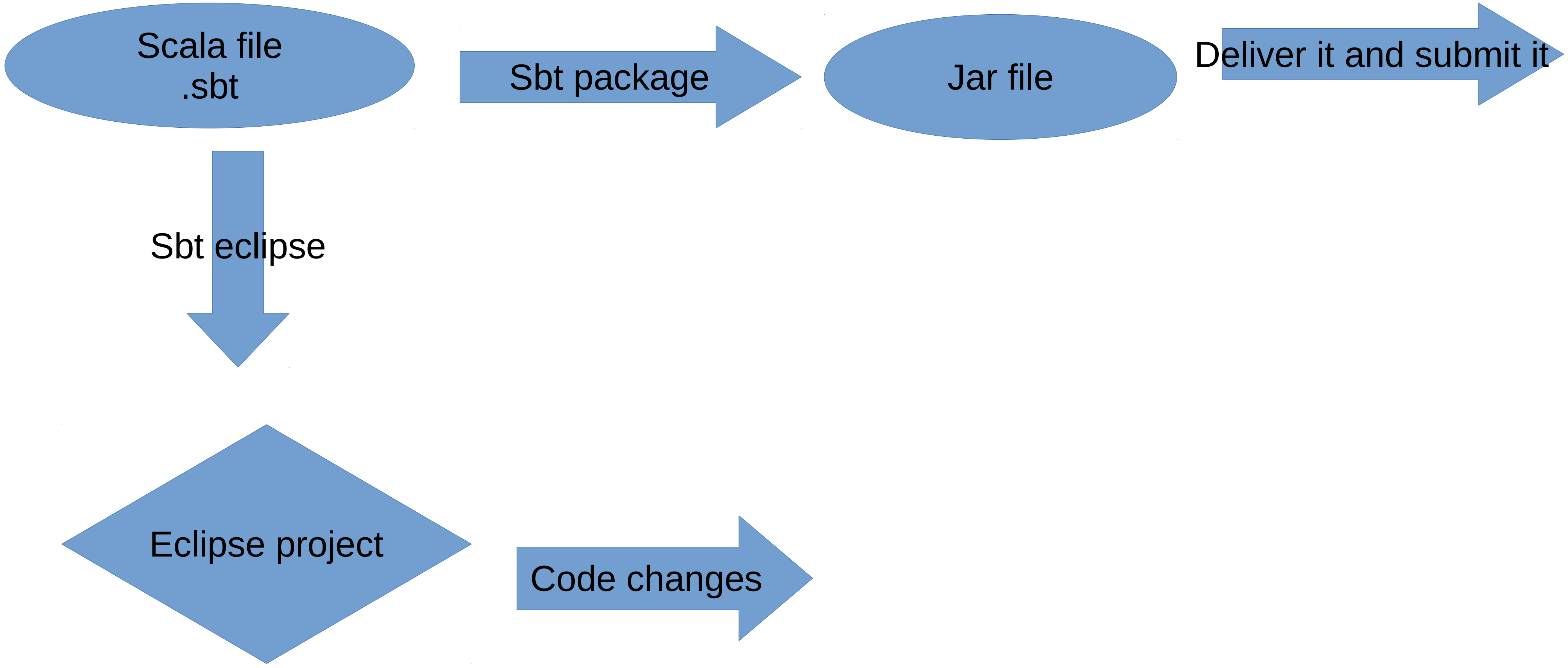

Master
Driver
filtered
filtered
filtered
Task
task
task
Spark worker
Spark worker
Spark worker
Cass cluster

1 million claims

TX

NY

CA



Scala file
.sbt

Sbt package

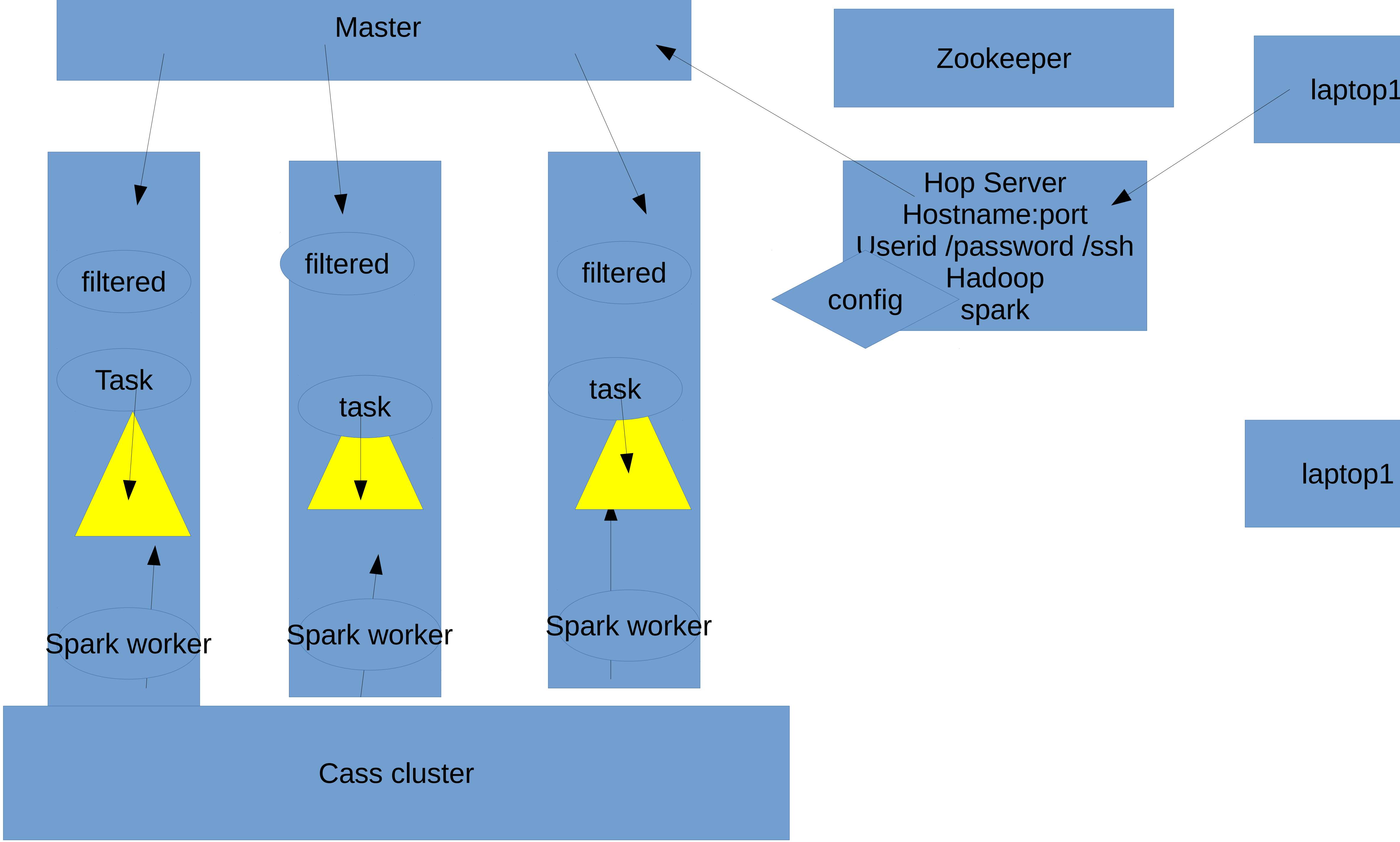
Jar file

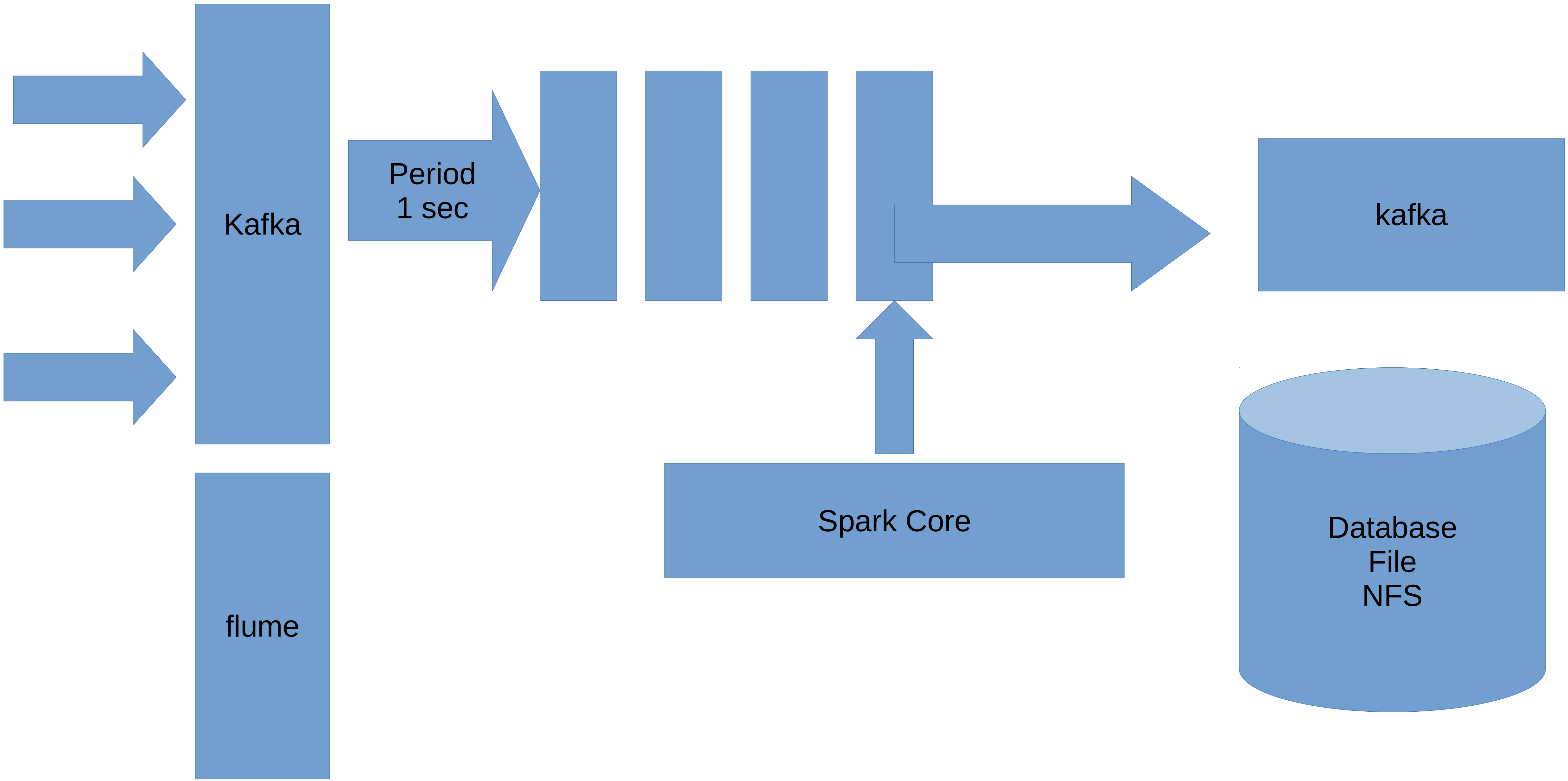
Deliver it and submit it

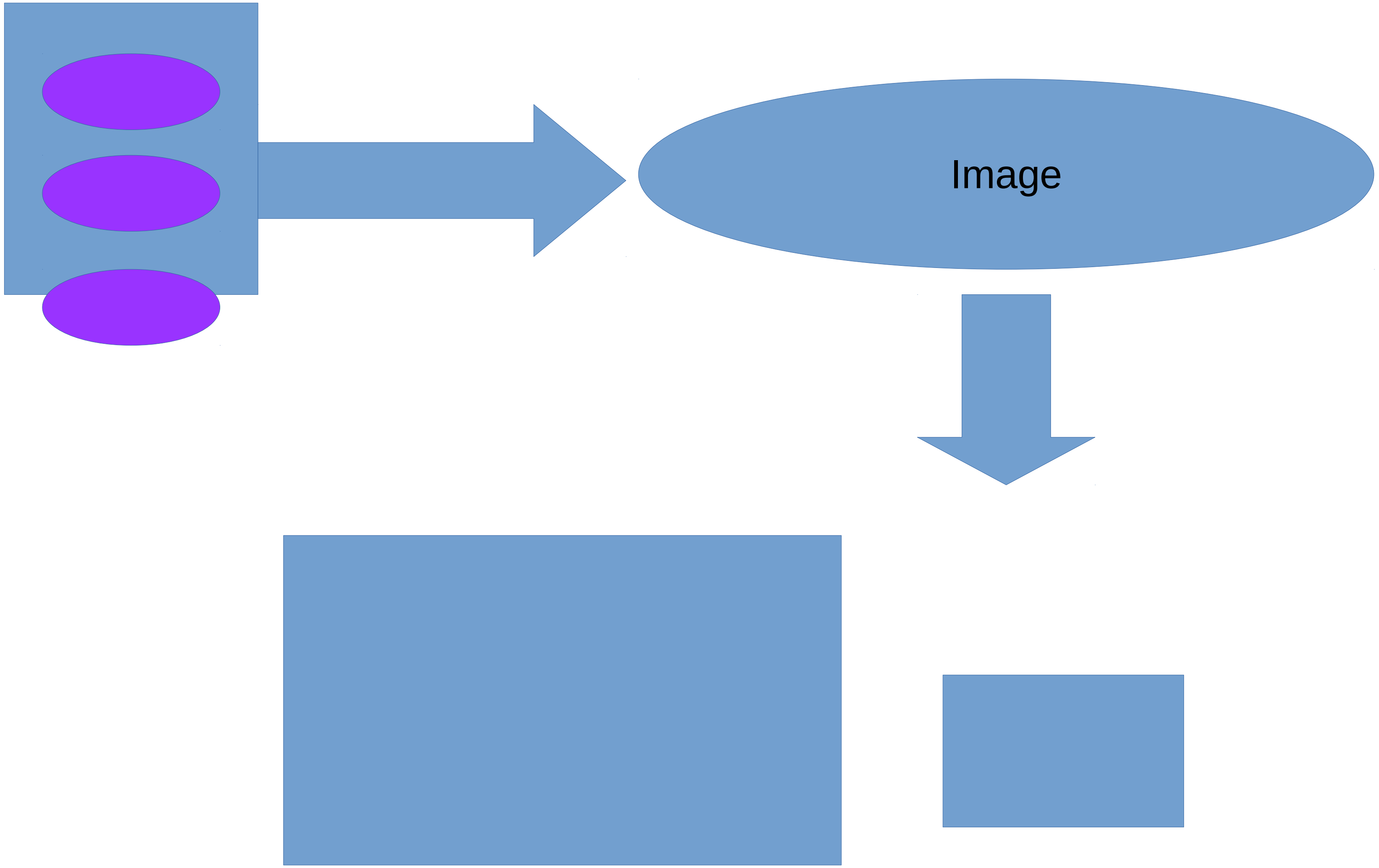
Sbt eclipse

Eclipse project

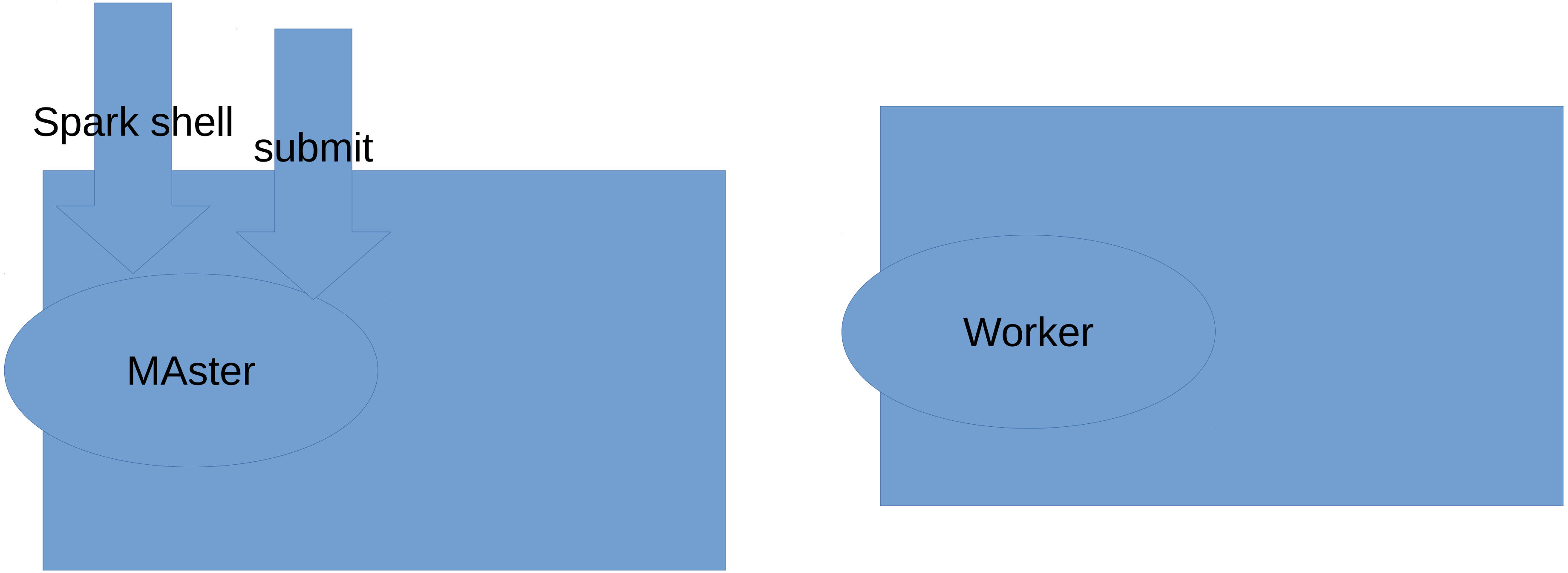
Code changes

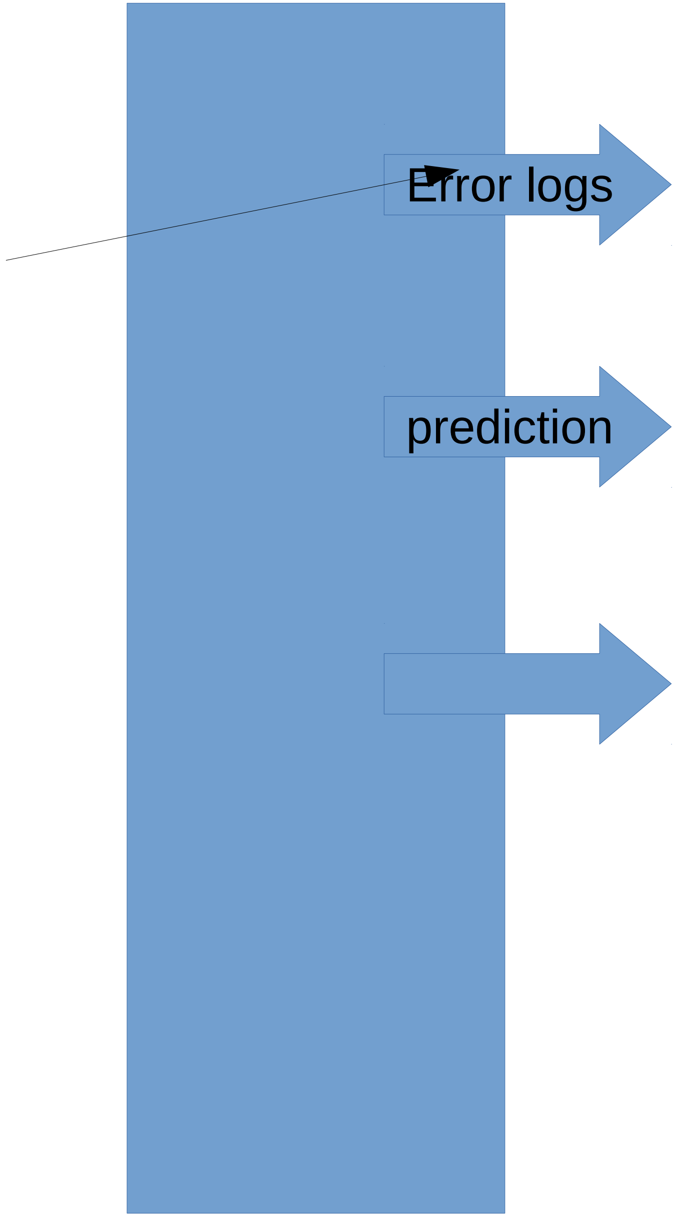

Master
Zookeeper
laptop1
Hop Server
Hostname:port
Userid /password /ssh
Hadoop
spark
config
filtered
filtered
filtered
Task
task
task
Spark worker
Spark worker
Spark worker
laptop1
Cass cluster

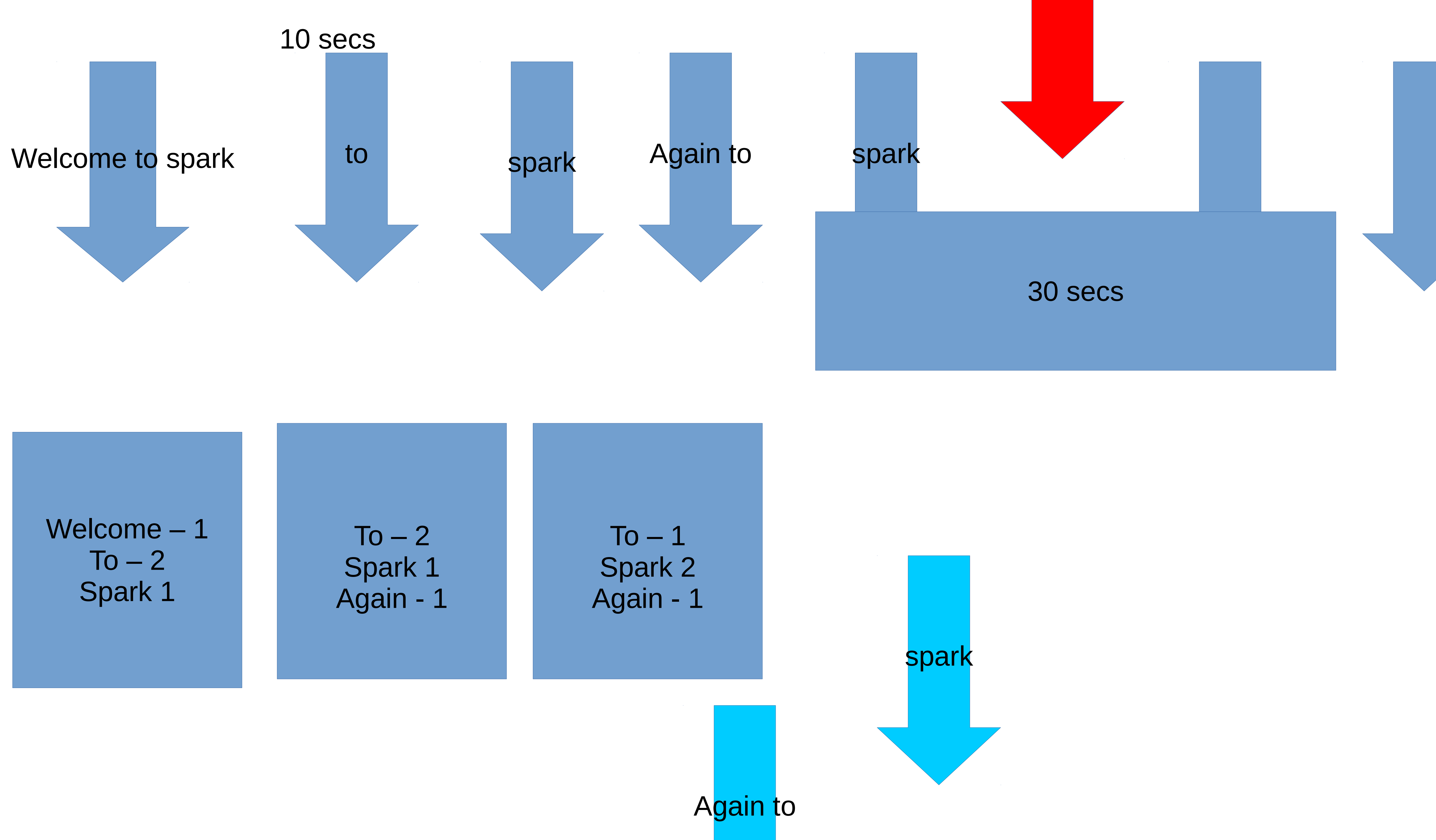

Kafka
flume
Period
1 sec
Spark Core
kafka
Database
File
NFS



Image


Spark shell
submit
MAster
Worker


Error logs
prediction



10 secs

Welcome to spark

to

spark

Again to

spark

30 secs

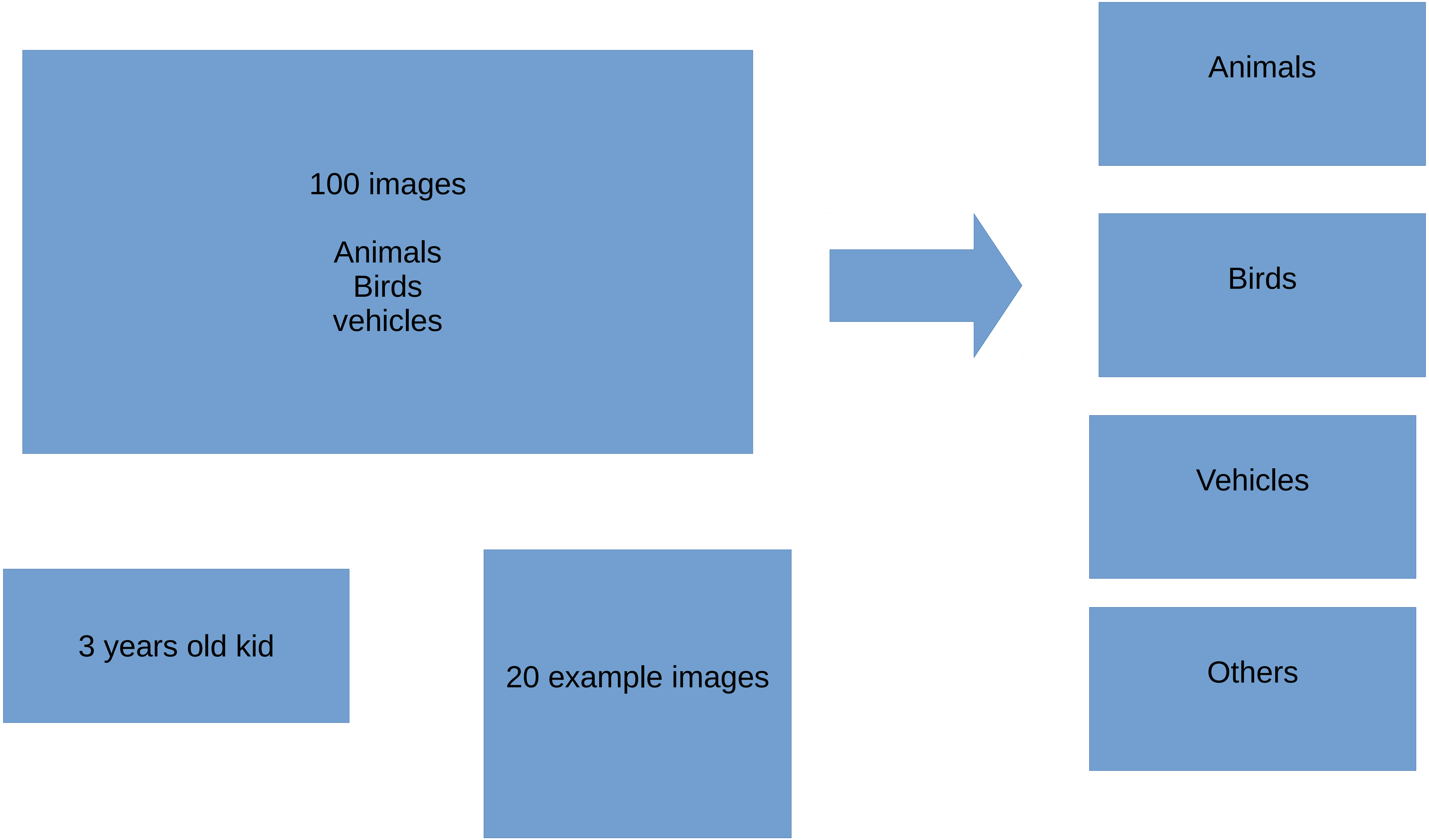
Welcome – 1
To – 2
Spark 1

To – 2
Spark 1
Again - 1

To – 1
Spark 2
Again - 1

spark

Again to



100 images

Animals
Birds
vehicles

Animals

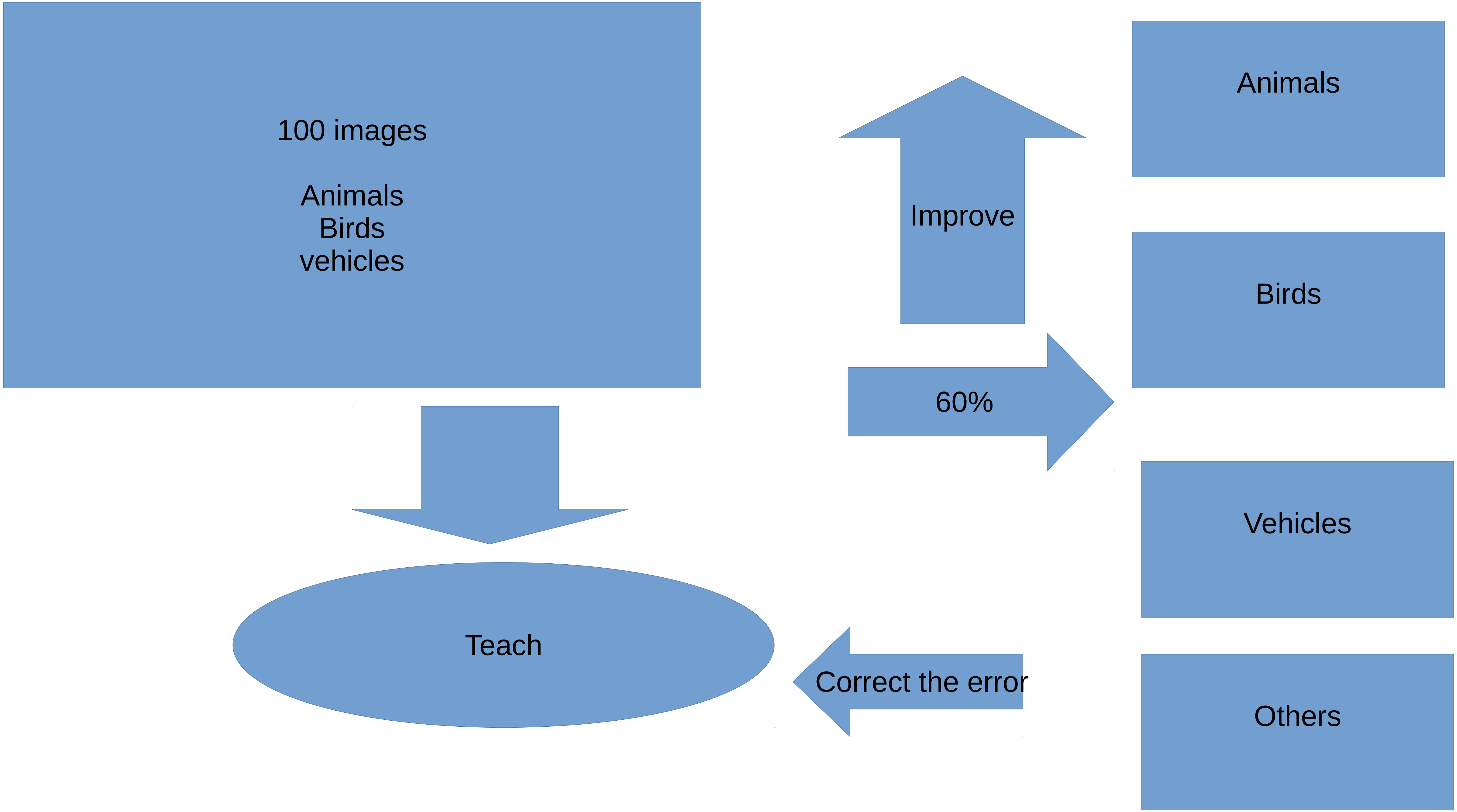
Birds

Vehicles

Others

3 years old kid

20 example images



100 images

Animals
Birds
vehicles

Teach

Correct the error

60%

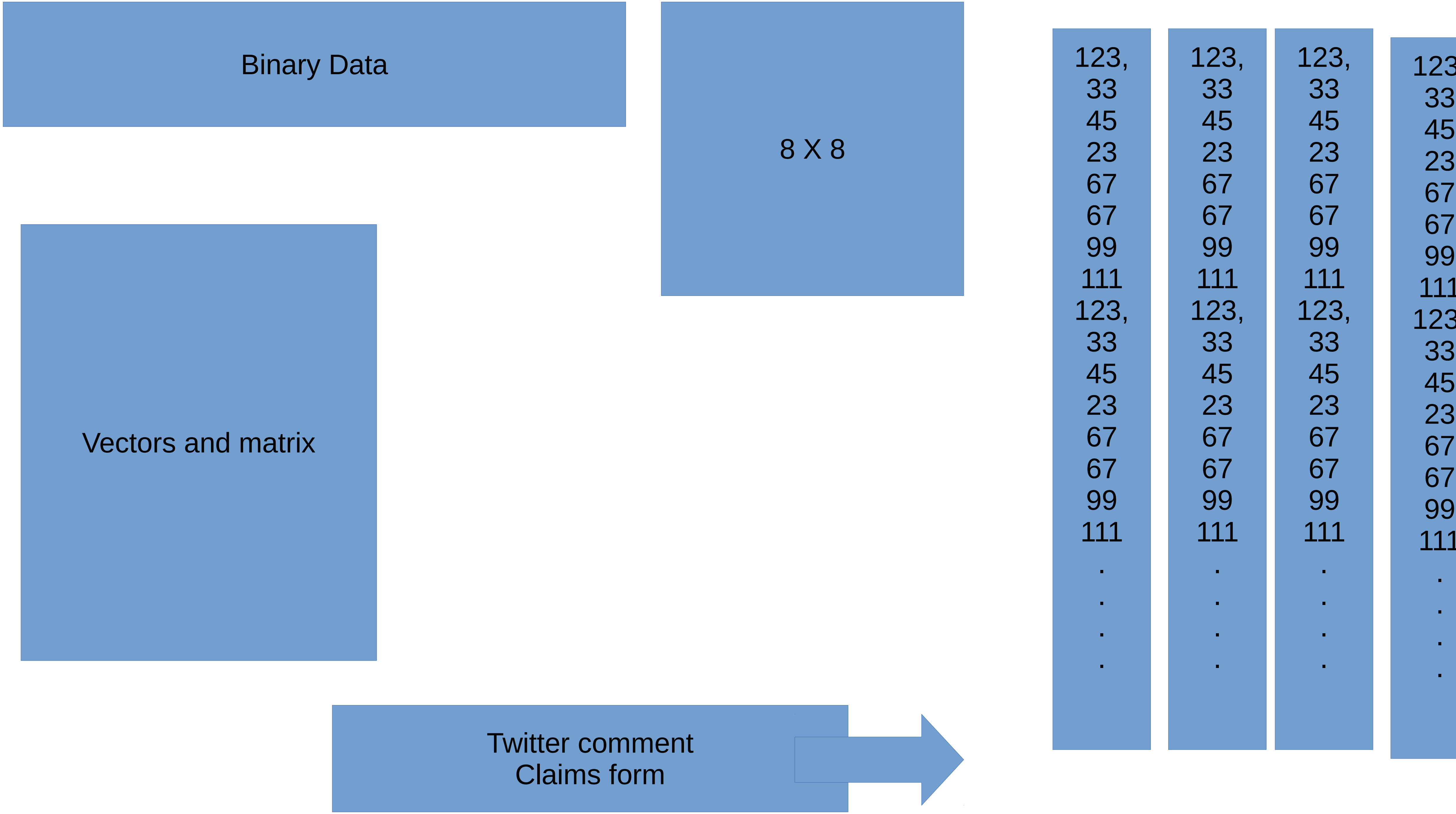
Improve

Animals

Birds

Vehicles

Others



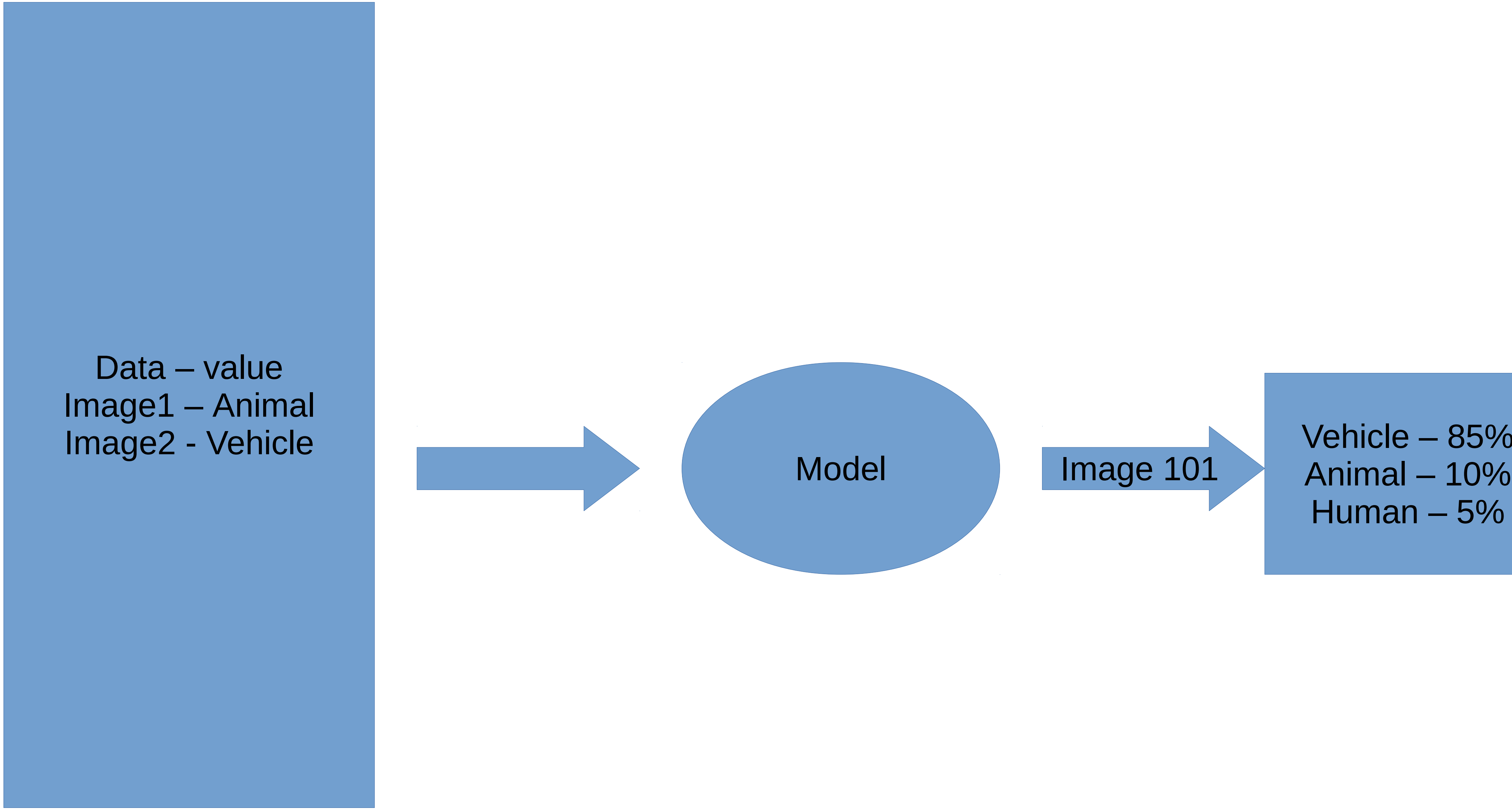
Binary Data

8 X 8

Vectors and matrix

Twitter comment
Claims form

123,	123,	123,	123
33	33	33	33
45	45	45	45
23	23	23	23
67	67	67	67
67	67	67	67
99	99	99	99
111	111	111	111
123,	123,	123,	123
33	33	33	33
45	45	45	45
23	23	23	23
67	67	67	67
67	67	67	67
99	99	99	99
111	111	111	111
.	.	.	.
.	.	.	.
.	.	.	.
.	.	.	.

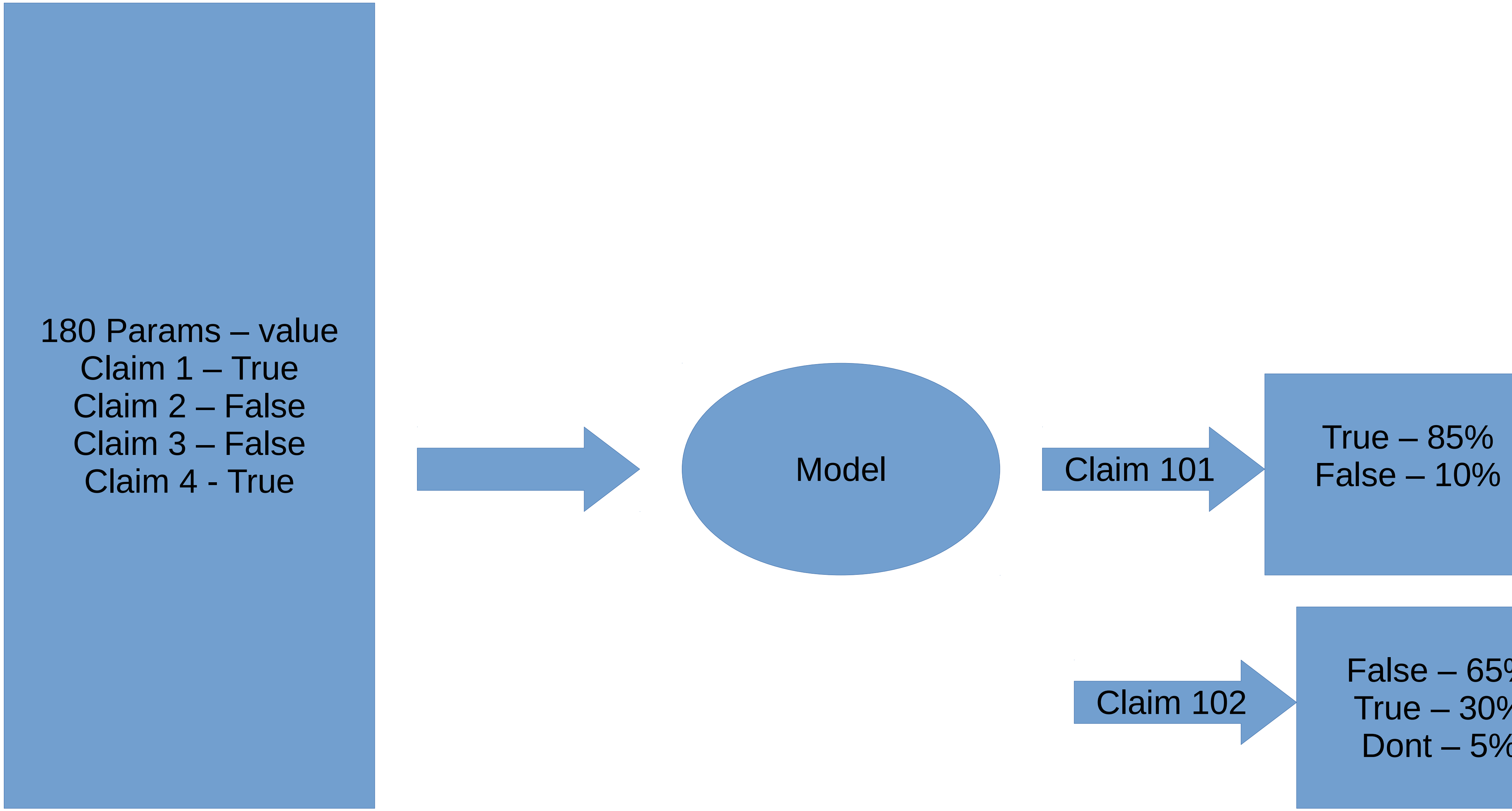


Data – value
Image1 – Animal
Image2 - Vehicle

Model

Image 101

Vehicle – 85%
Animal – 10%
Human – 5%



180 Params – value
Claim 1 – True
Claim 2 – False
Claim 3 – False
Claim 4 - True

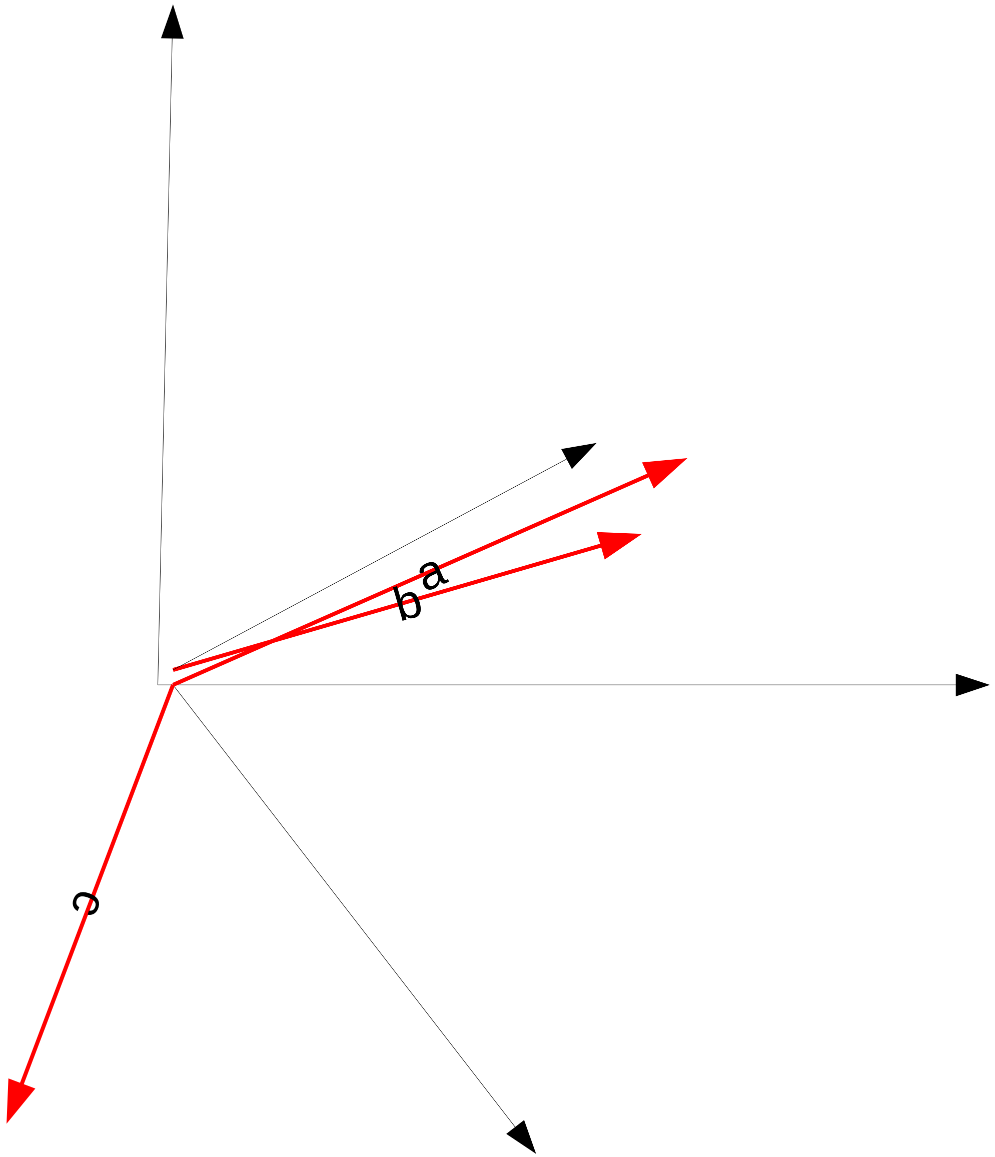
Model

Claim 101

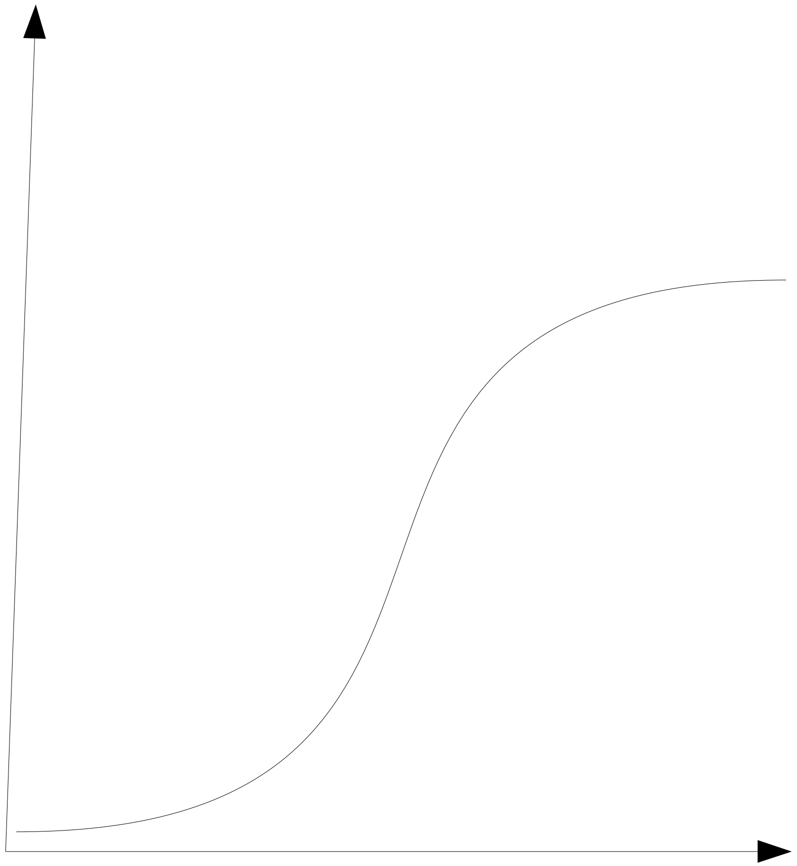
True – 85%
False – 10%

Claim 102

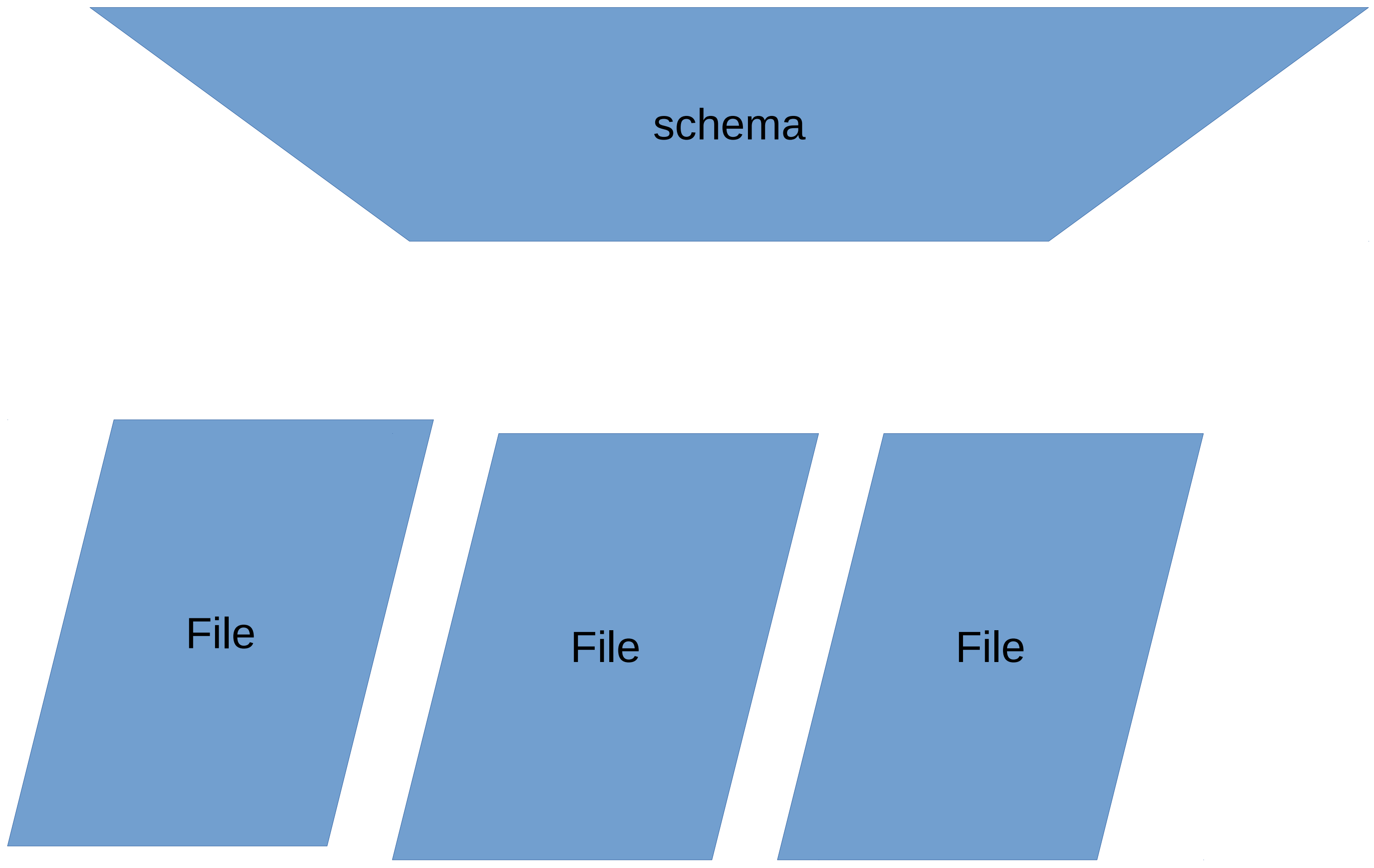
False – 65
True – 30
Dont – 5


a
b
c

6” = $3
8” = $4
10” - $4.75
12 – $5.5



50,50,2,100,2,2

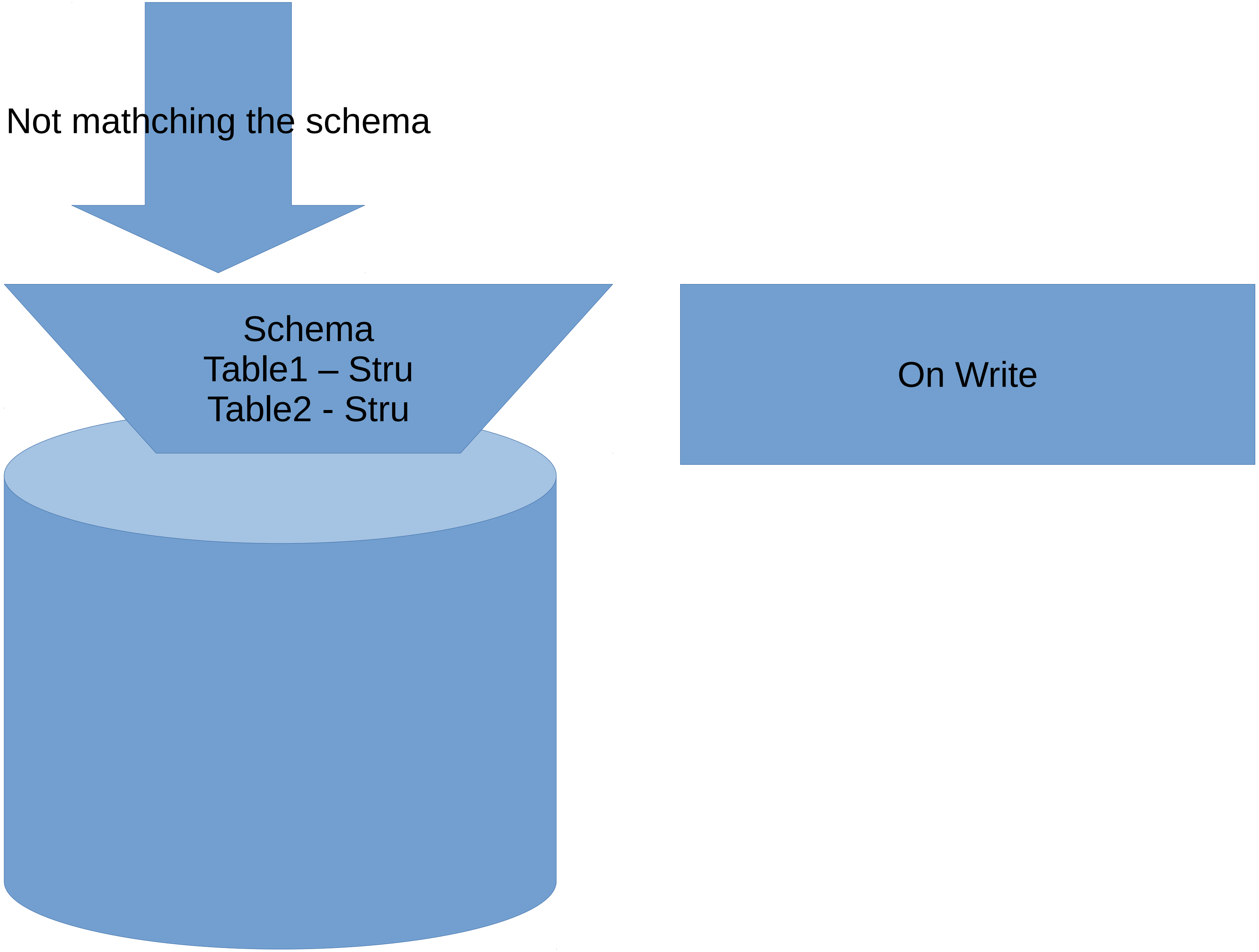


schema

File

File

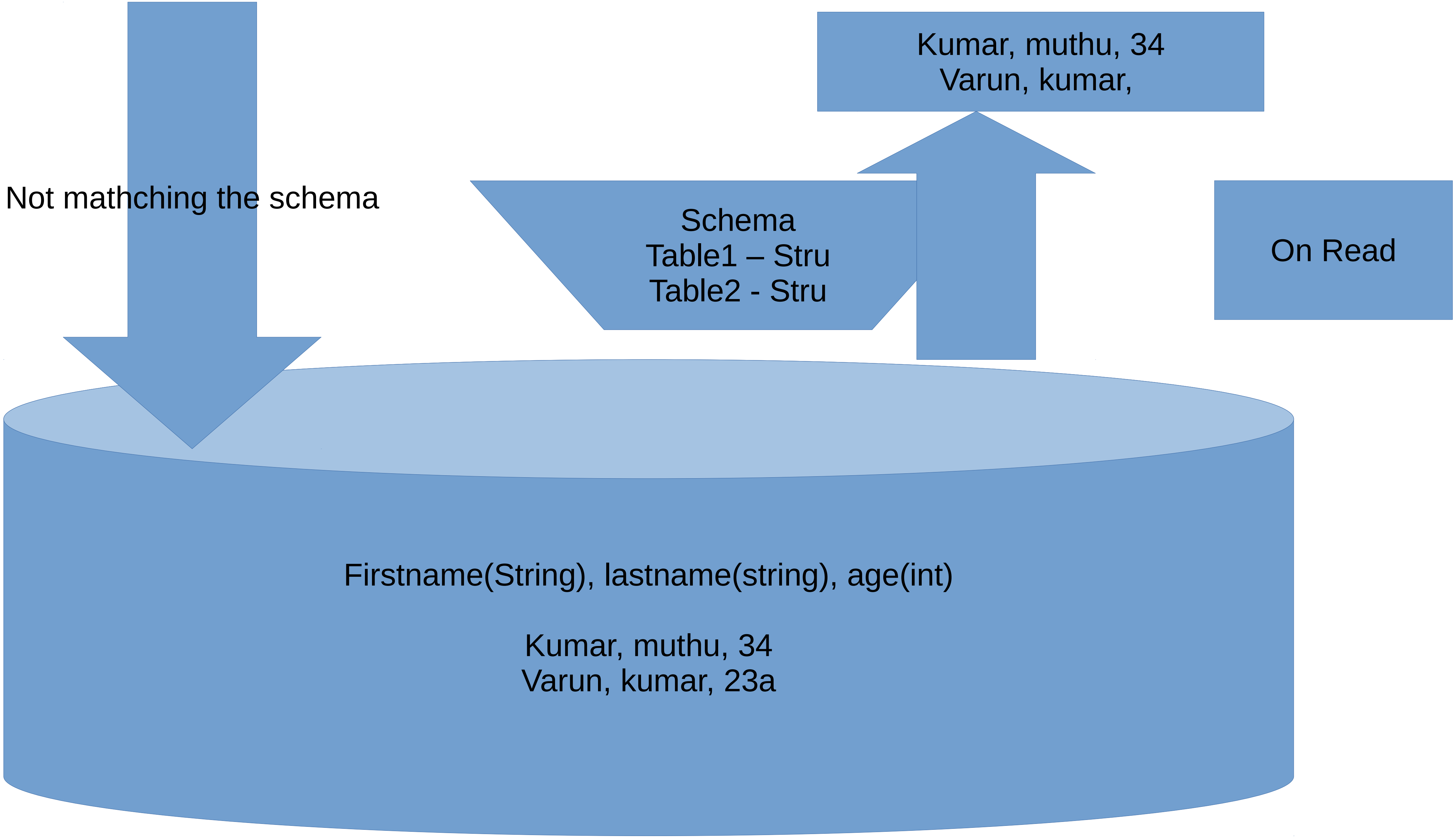
File



Not mathching the schema

Schema
Table1 – Stru
Table2 - Stru

On Write



Not mathching the schema

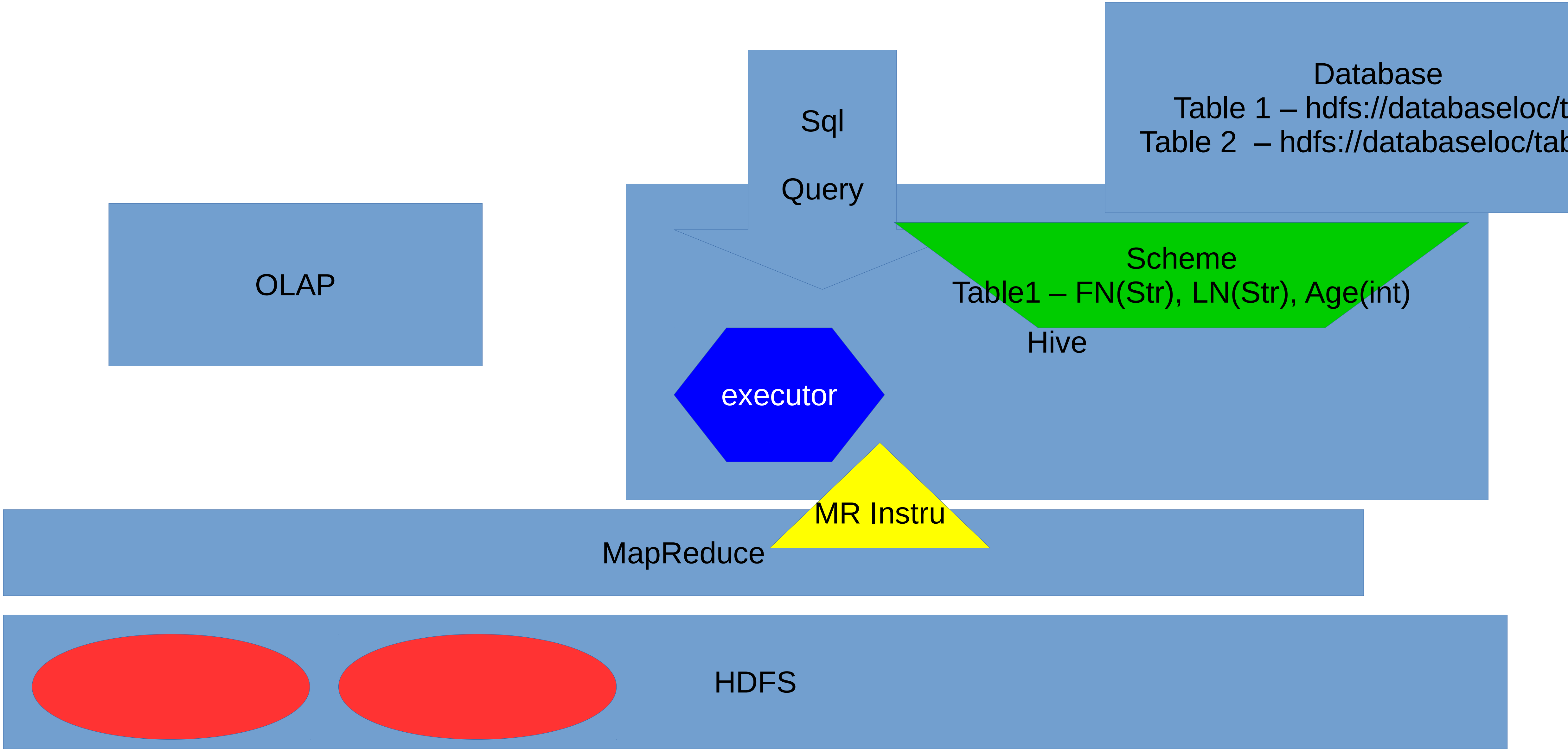
Schema
Table1 – Stru
Table2 - Stru

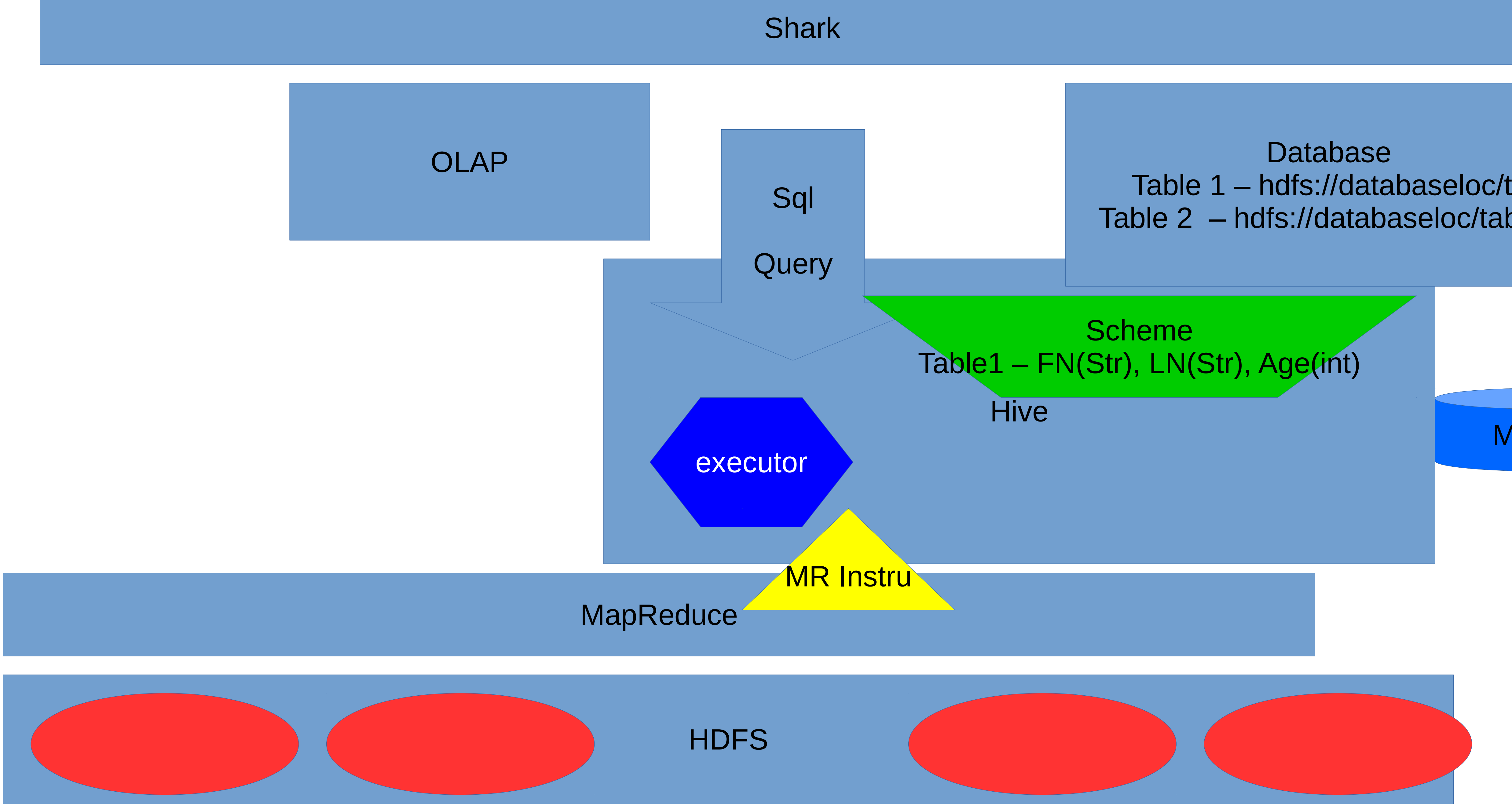
Kumar, muthu, 34
Varun, kumar,

On Read

Firstname(String), lastname(string), age(int)

Kumar, muthu, 34
Varun, kumar, 23a


Database
Table 1 – hdfs://databaseloc/t
Table 2 – hdfs://databaseloc/tab
Sql
Query
OLAP
Scheme
Table1 – FN(Str), LN(Str), Age(int)
Hive
executor
MR Instru
MapReduce
HDFS



Shark

OLAP

Sql

Query

Database
Table 1 – hdfs://databaseloc/t
Table 2 – hdfs://databaseloc/tab

Scheme
Table1 – FN(Str), LN(Str), Age(int)

Hive

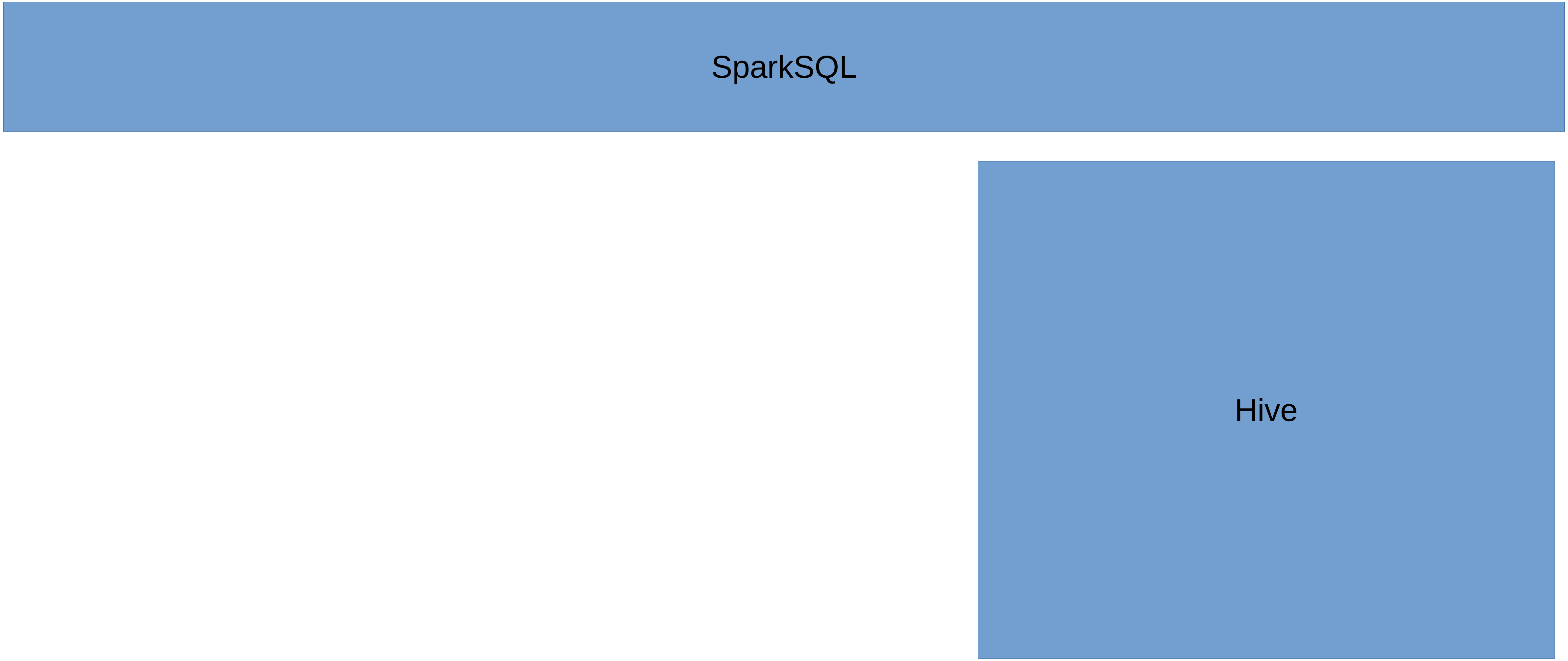
M

executor

MR Instru

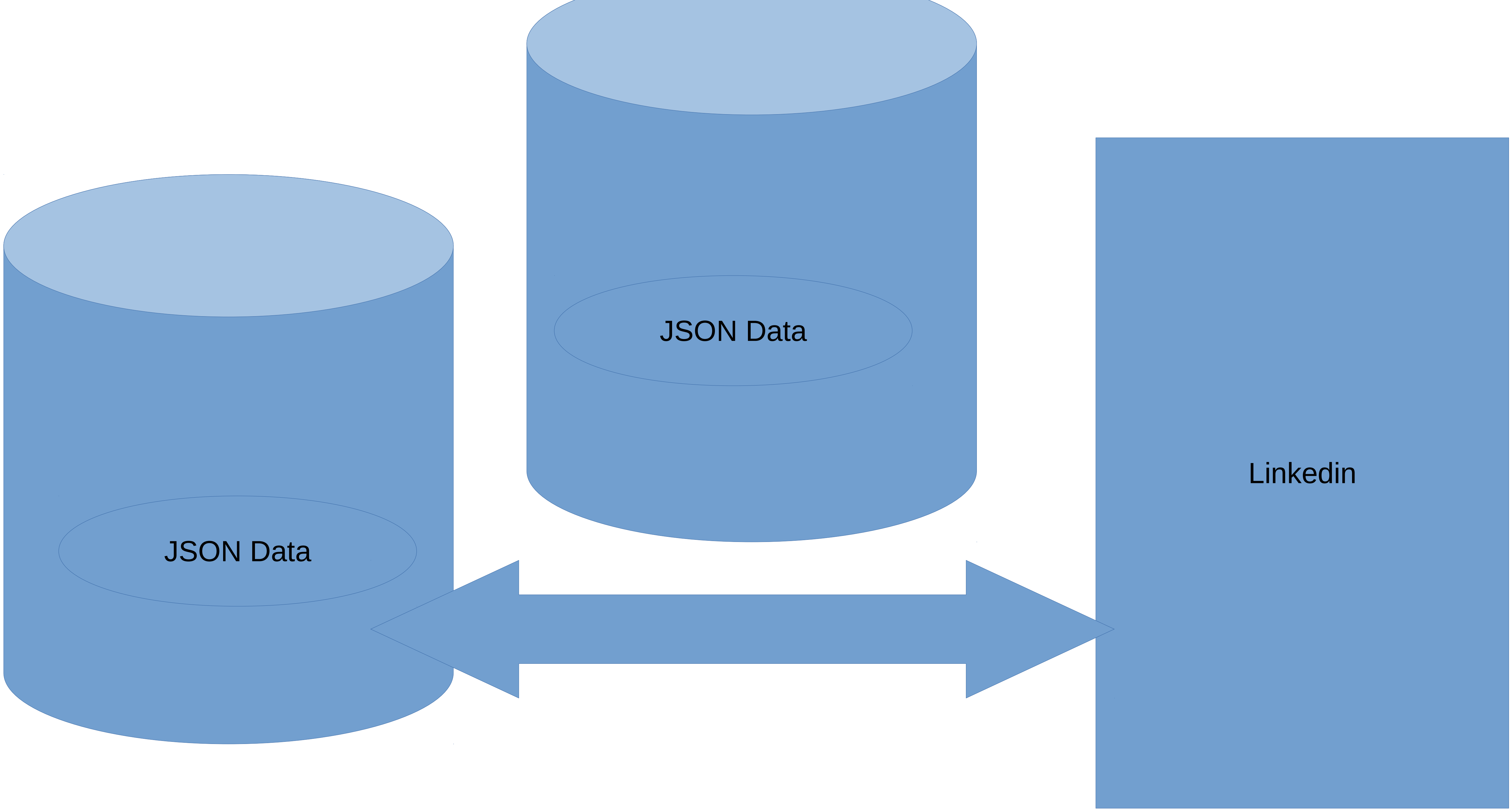
MapReduce

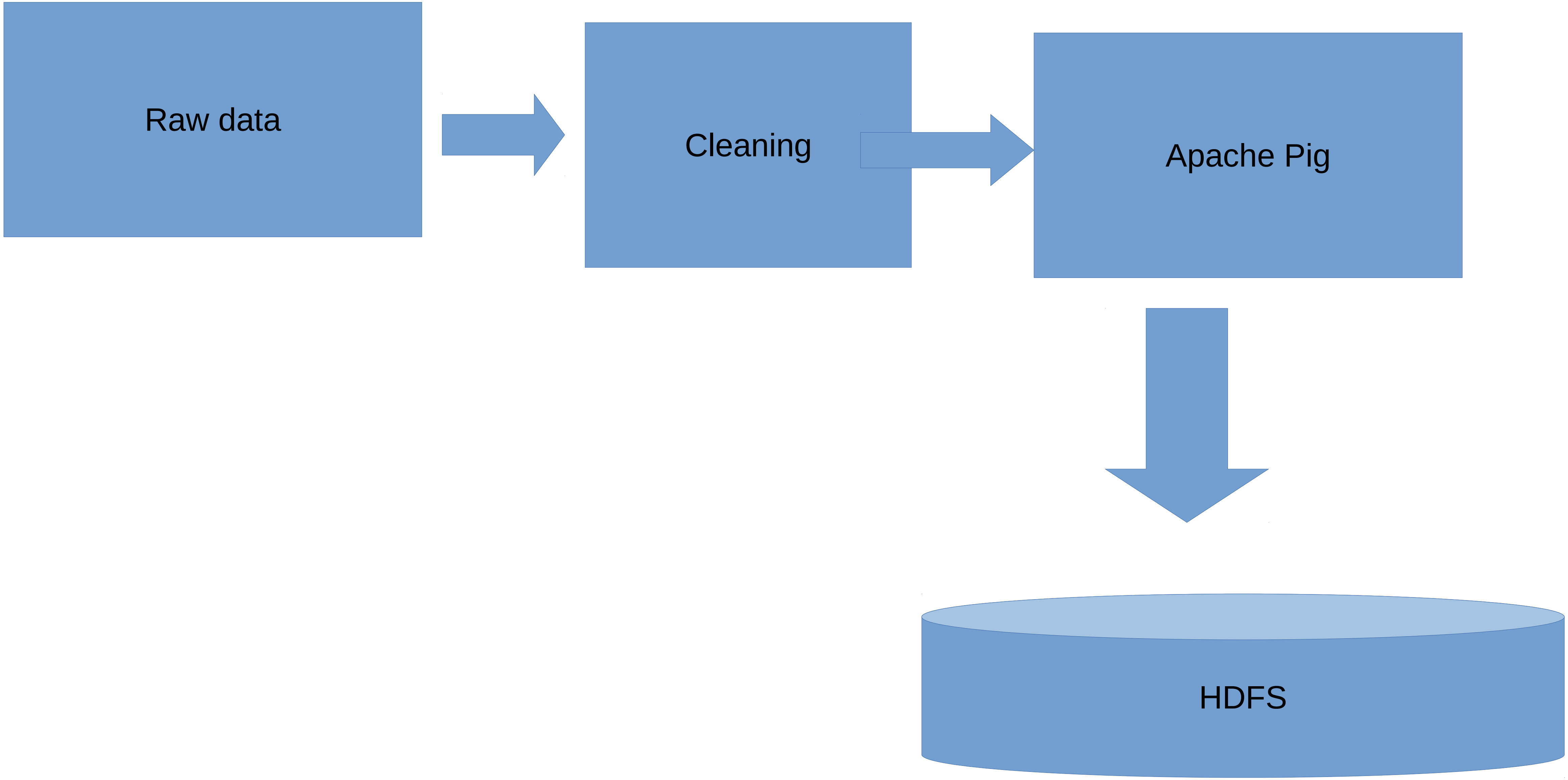
HDFS



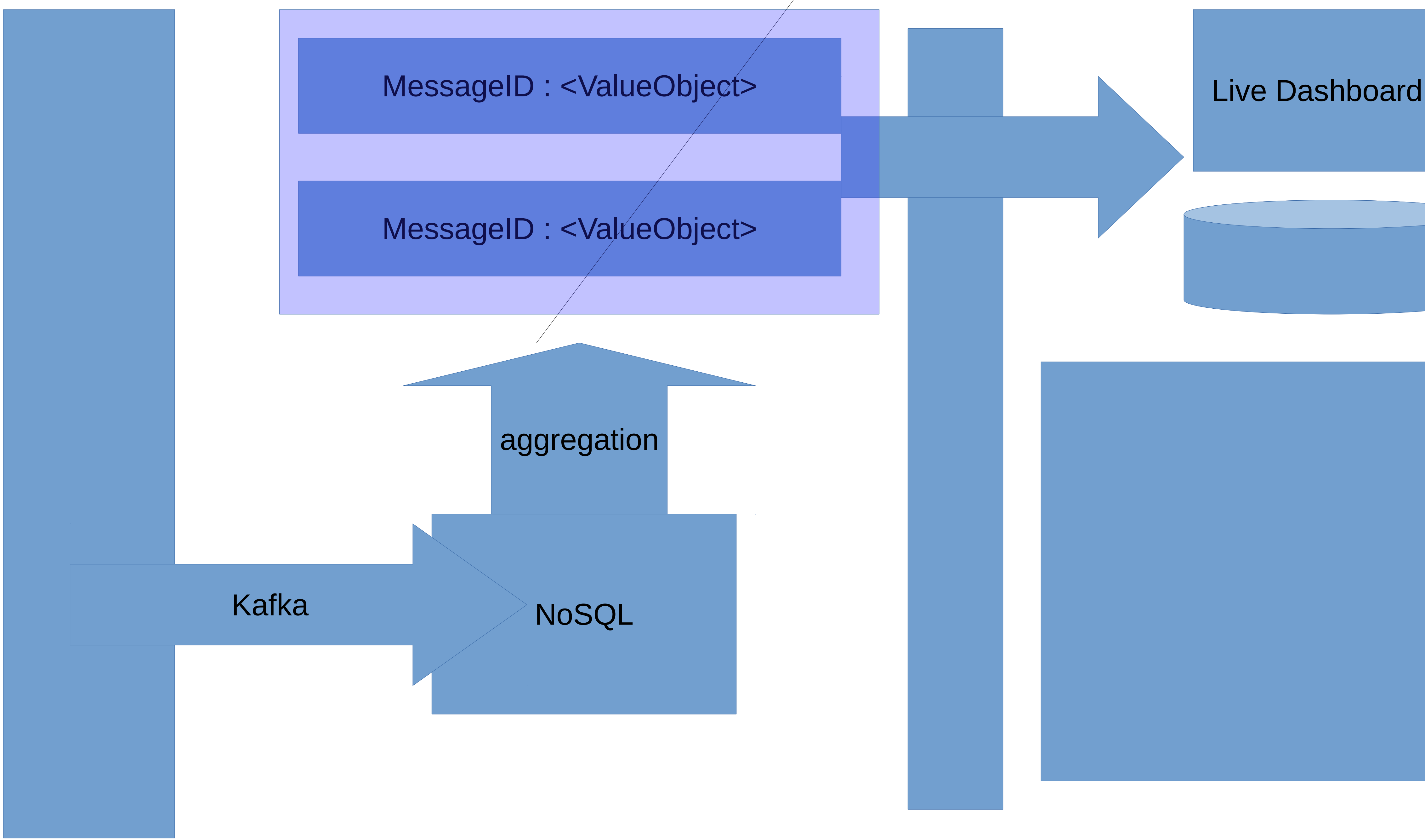
SparkSQL

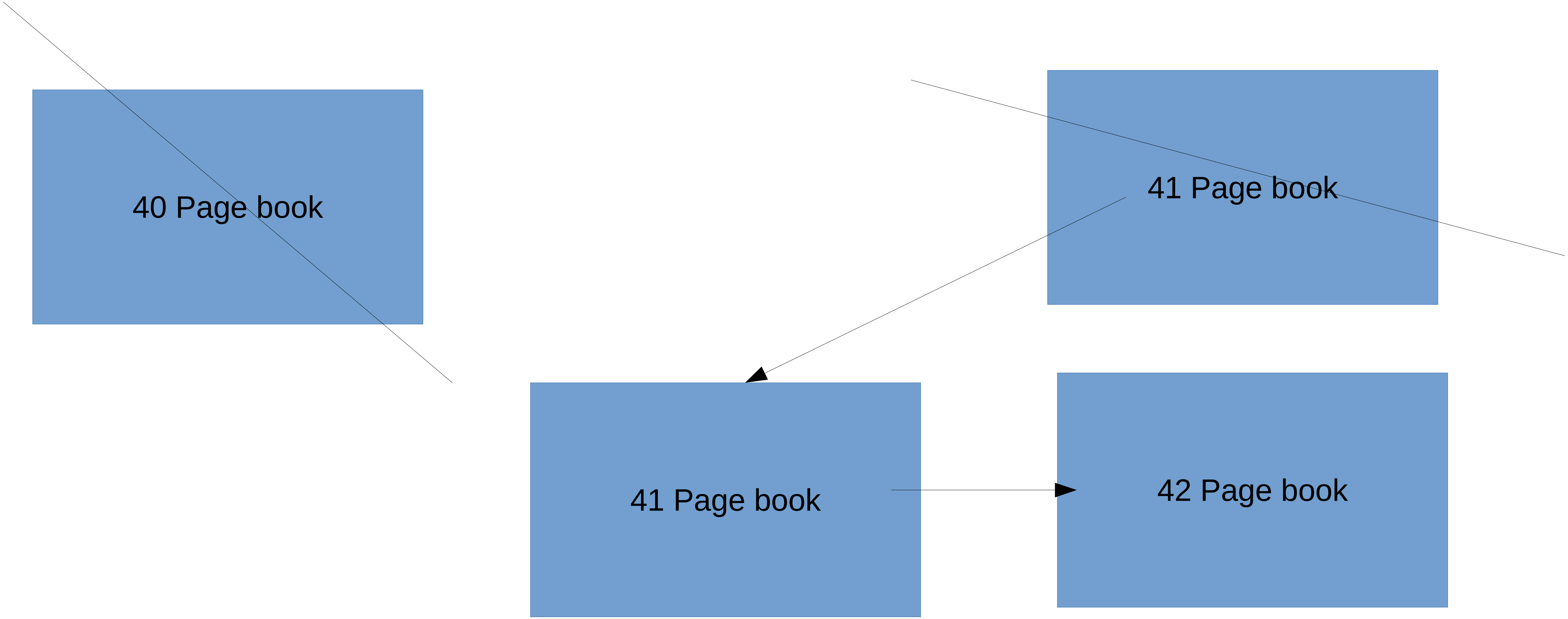
Hive


JSON Data
JSON Data
Linkedin



Raw data
Cleaning
Apache Pig
HDFS


MessageID : <ValueObject>
MessageID : <ValueObject>
Live Dashboard
aggregation
Kafka
NoSQL

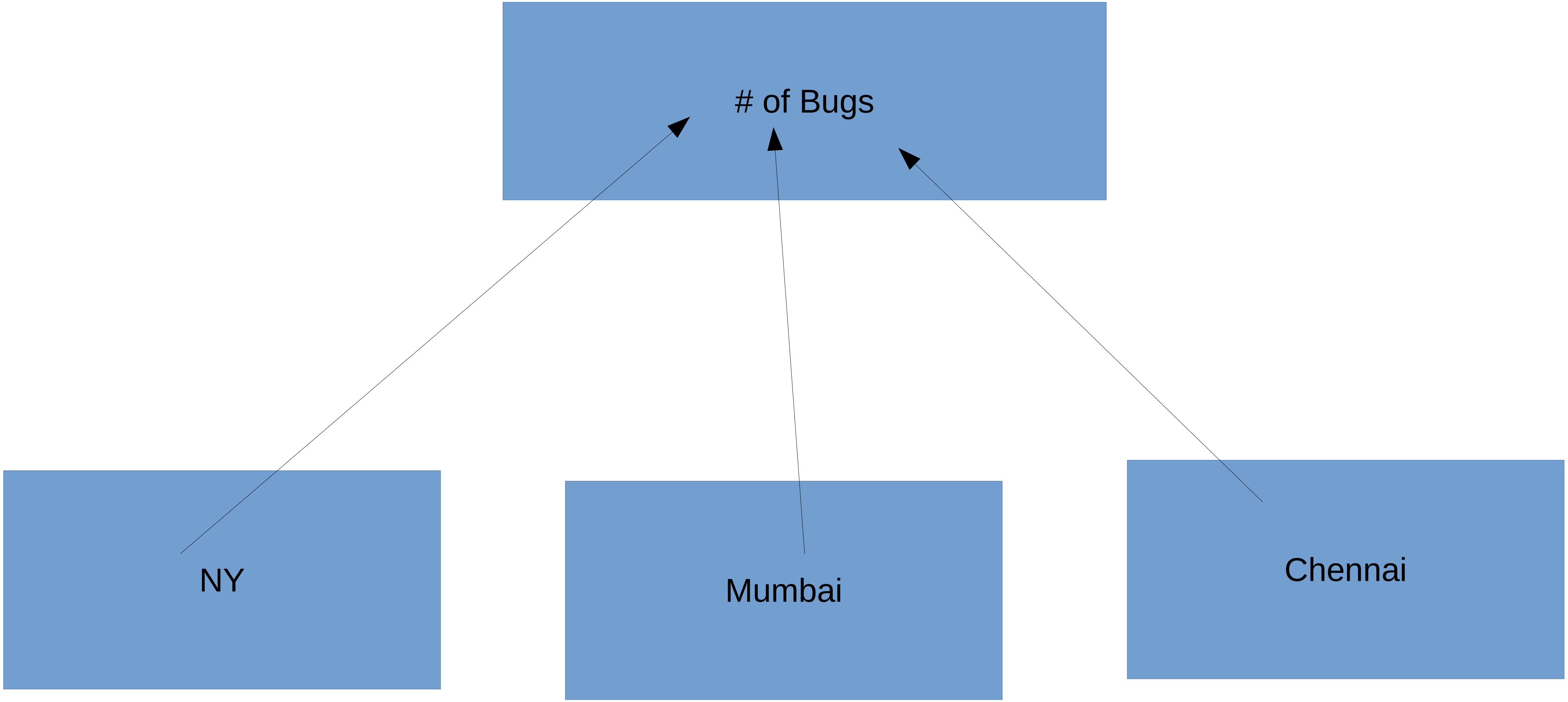


40 Page book

41 Page book

41 Page book

42 Page book

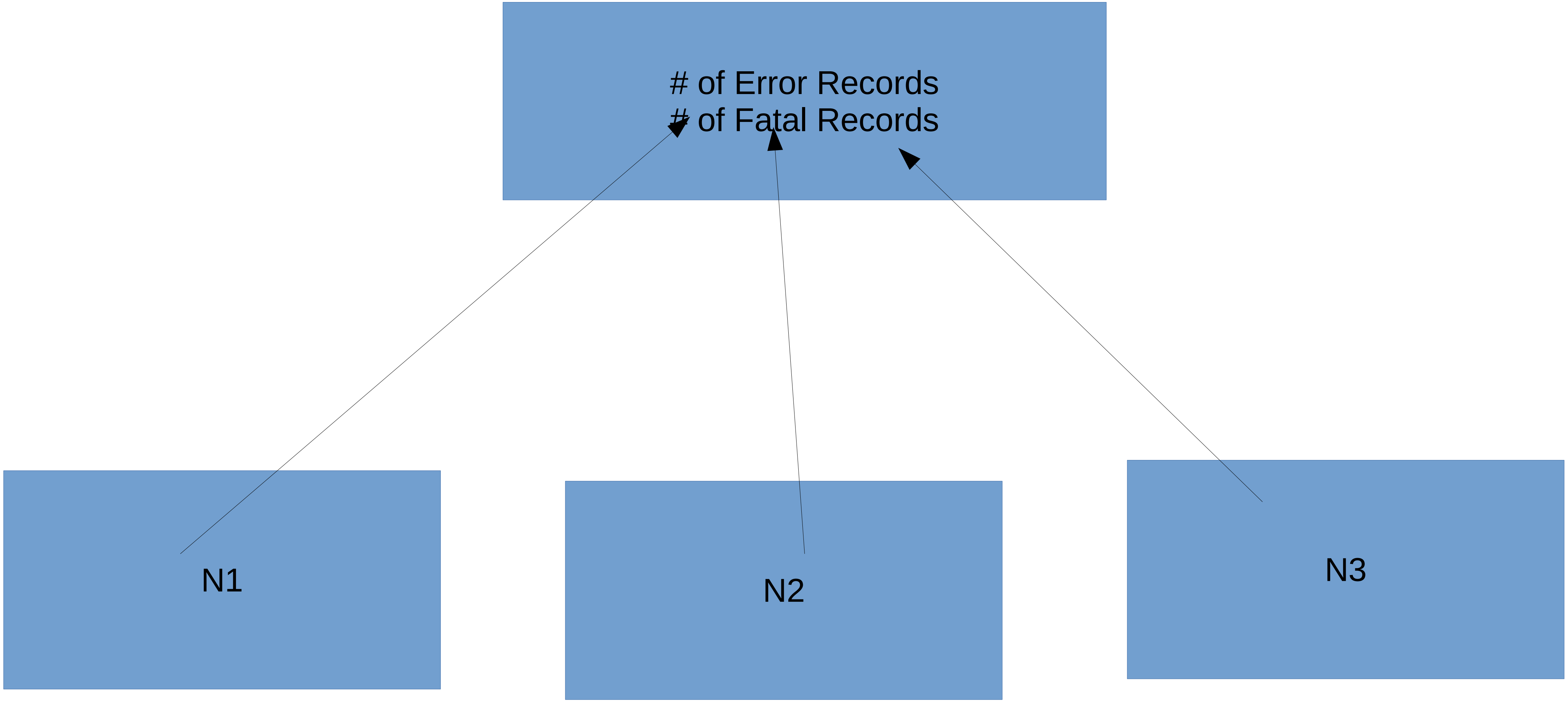


# of Bugs

NY

Mumbai

Chennai

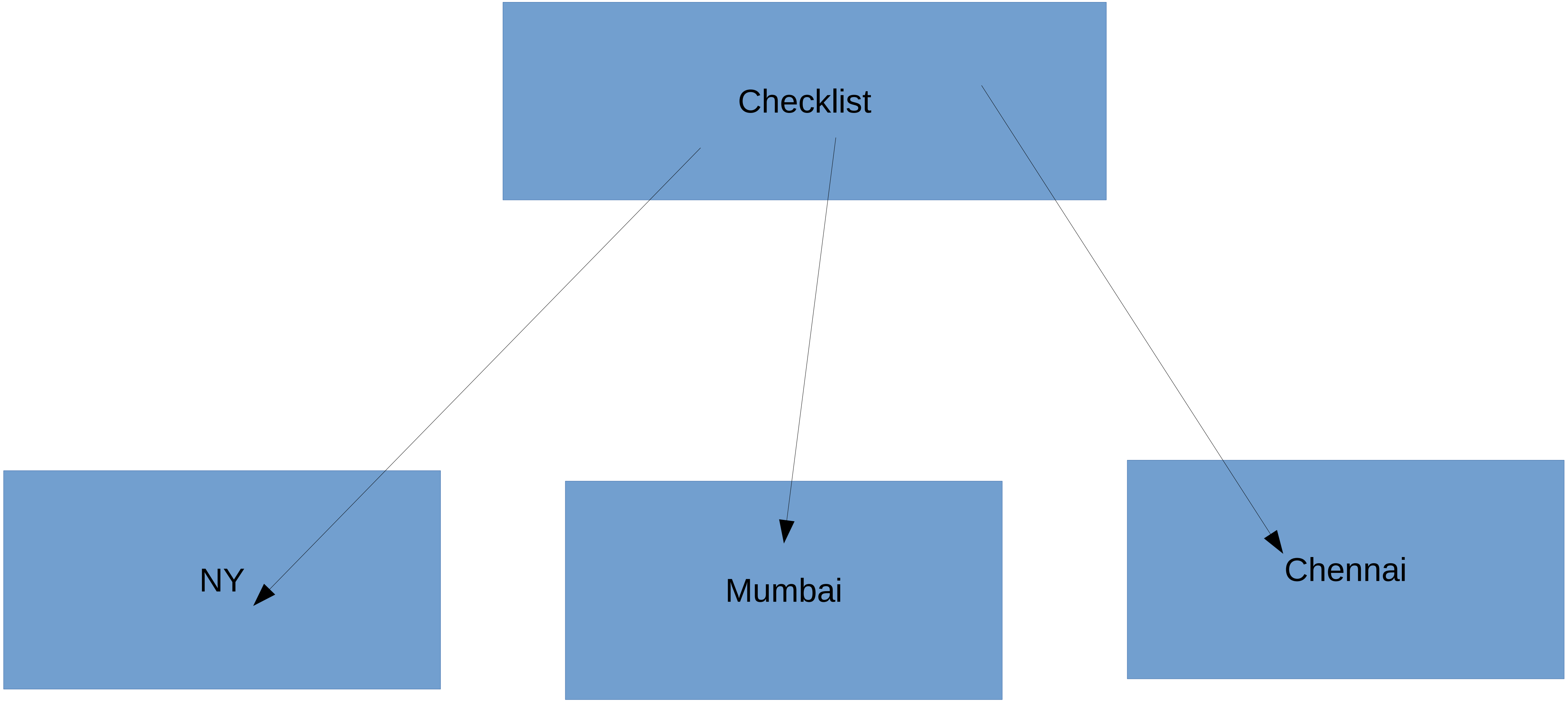


# of Error Records
# of Fatal Records

N1

N2

N3

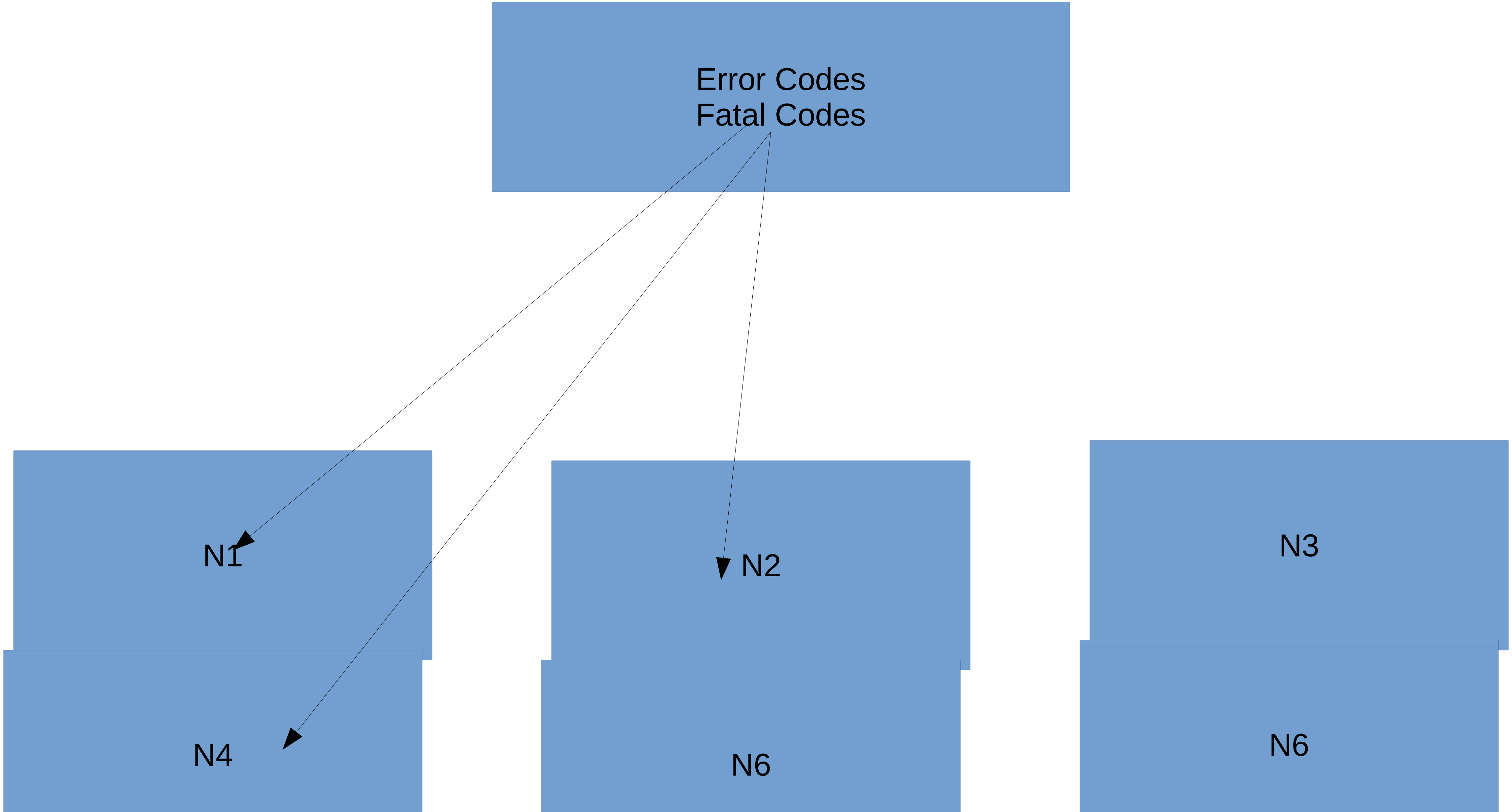


Checklist

NY

Mumbai

Chennai



Error Codes
Fatal Codes

N1

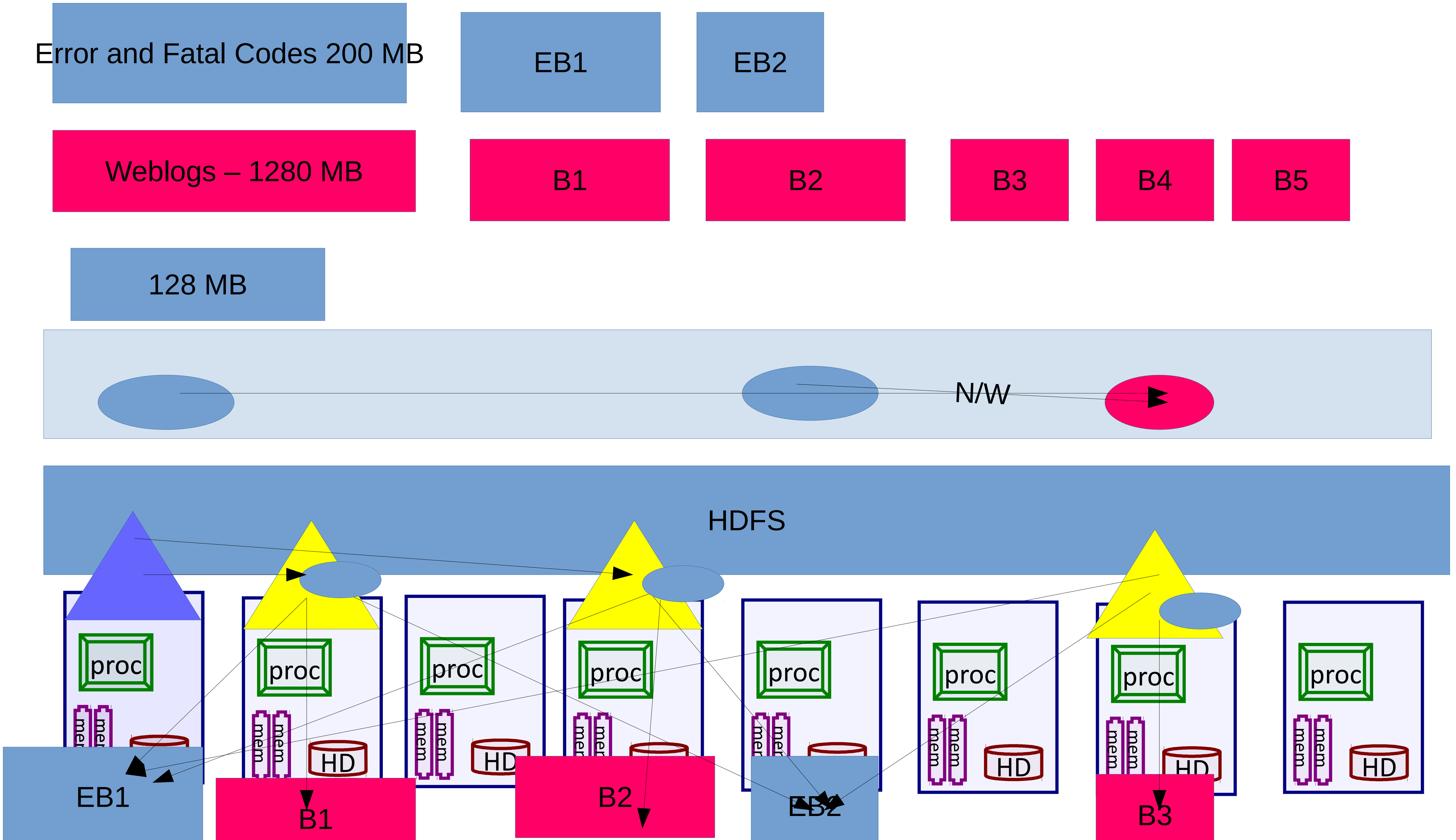
N2

N3

N4

N6

N6



Error and Fatal Codes 200 MB

EB1

EB2

Weblogs – 1280 MB

B1

B2

B3

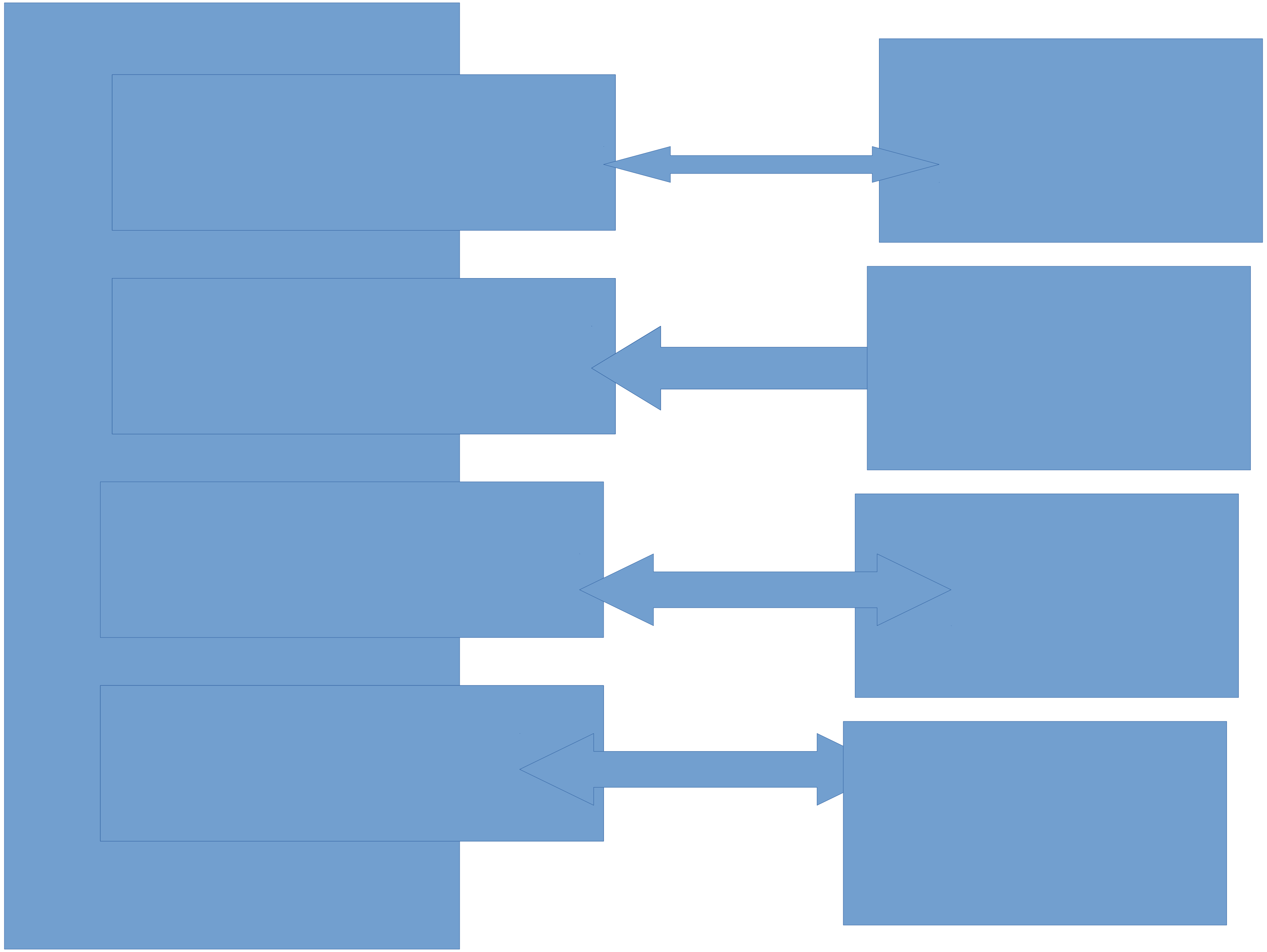
B4

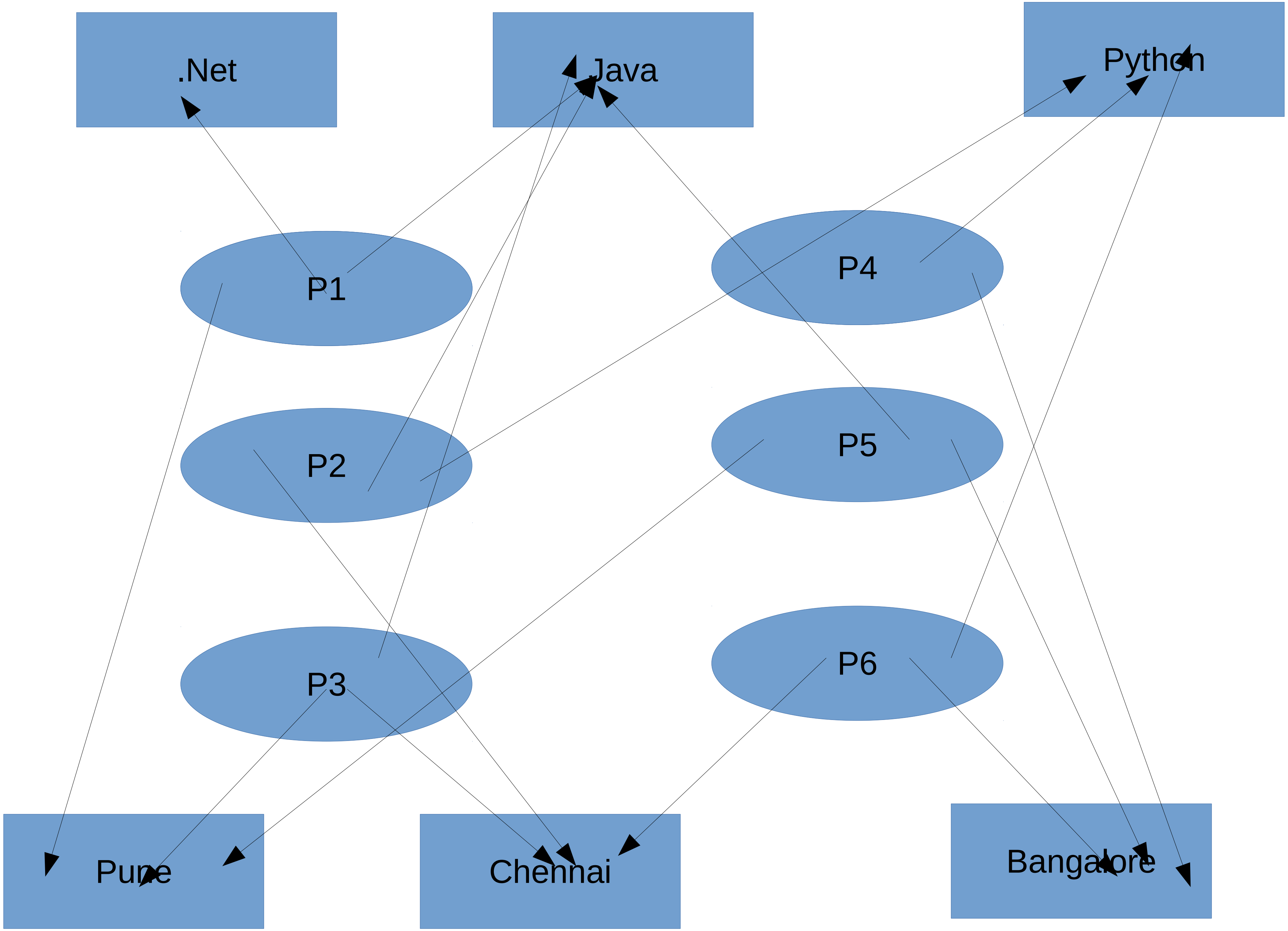
B5

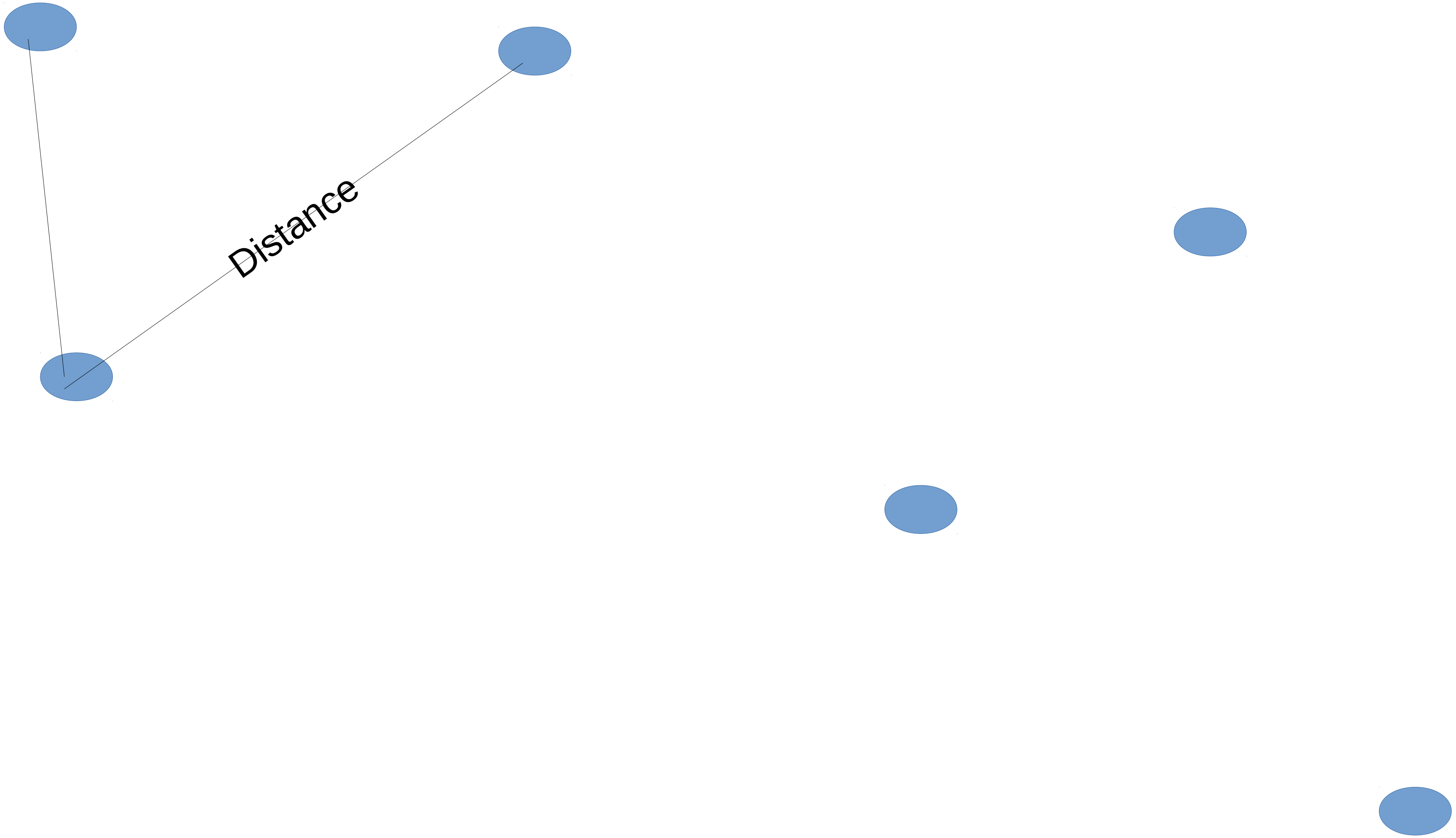
128 MB

N/W

HDFS

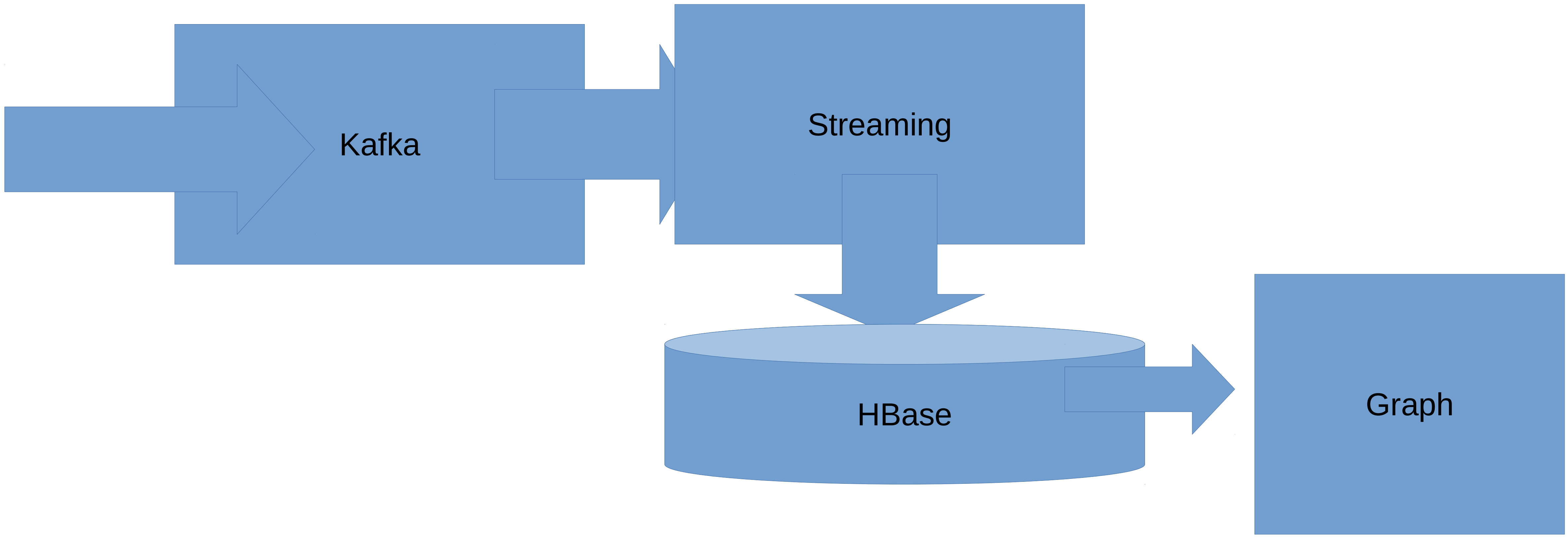


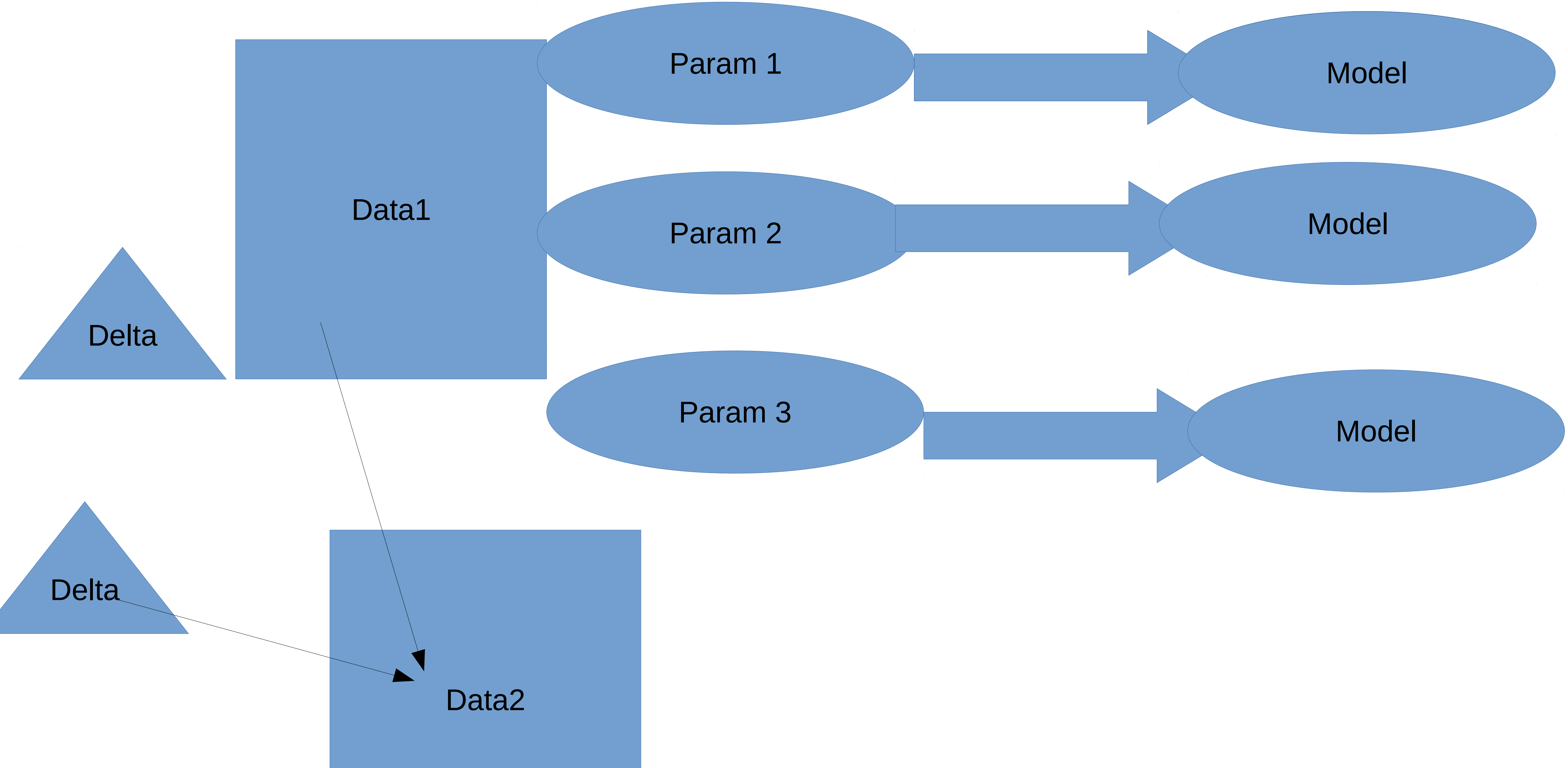

.Net
Java
Python
P1
P4
P2
P5
P3
P6
Pune
Chennai
Bangalore


Distance

Profile1 -> 5,6,9,12
Profile2 -> 3,7,12,20

2 -> Java, .Net, Python
3 -> Python


Kafka
Streaming
HBase
Graph



Param 1

Model

Data1

Param 2

Model

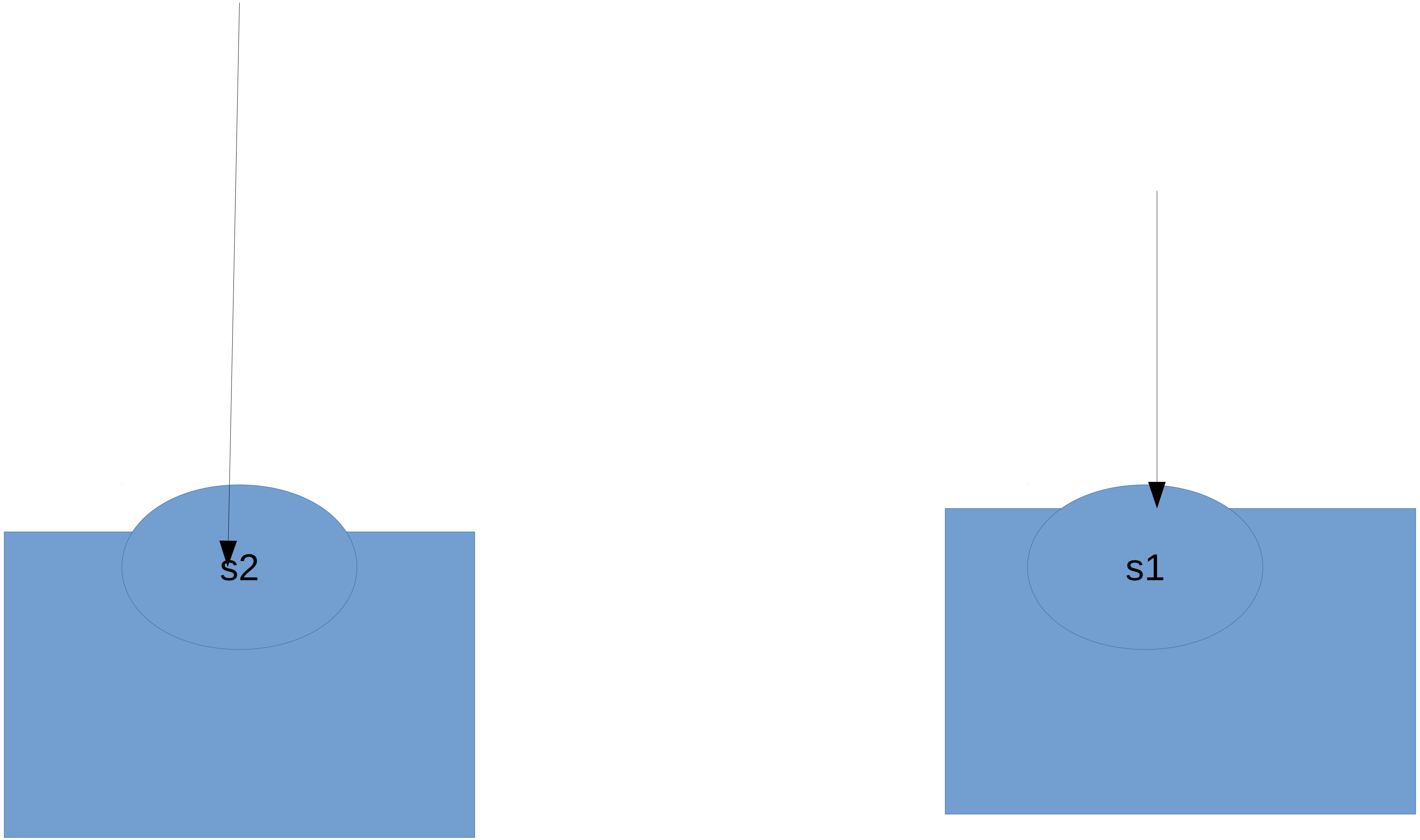
Delta

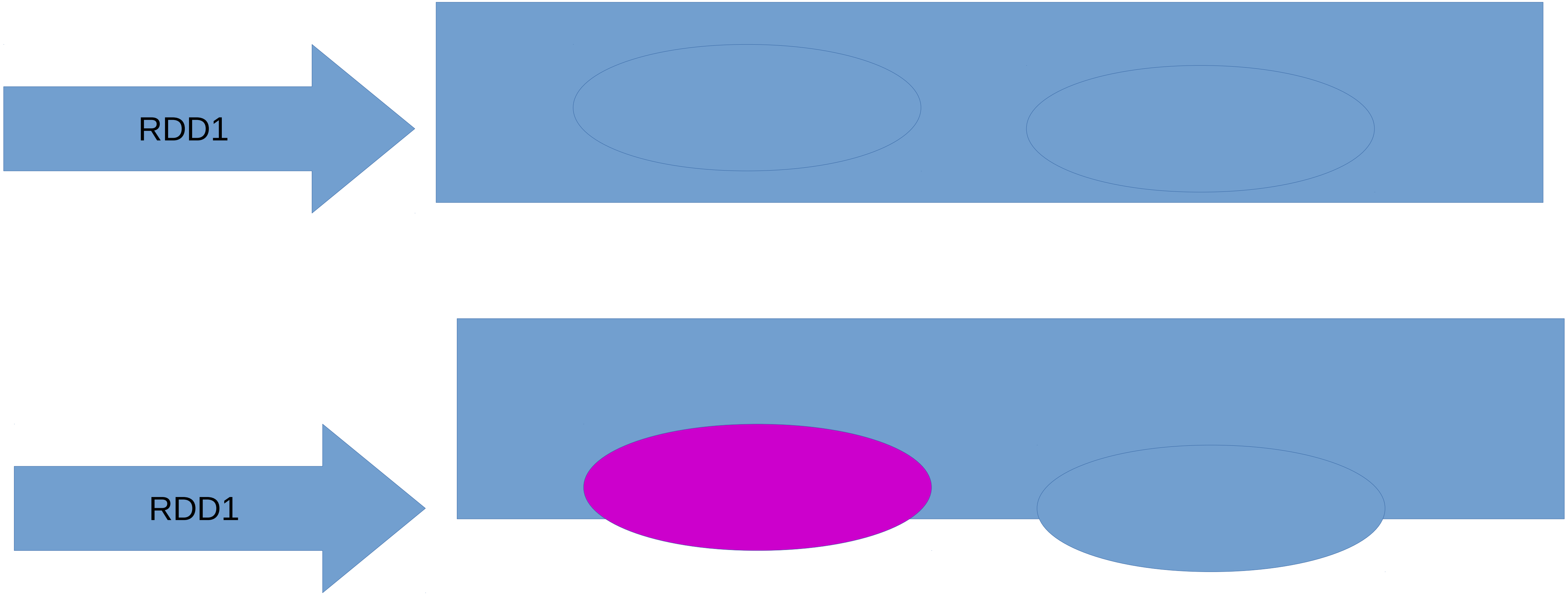
Param 3

Model

Delta

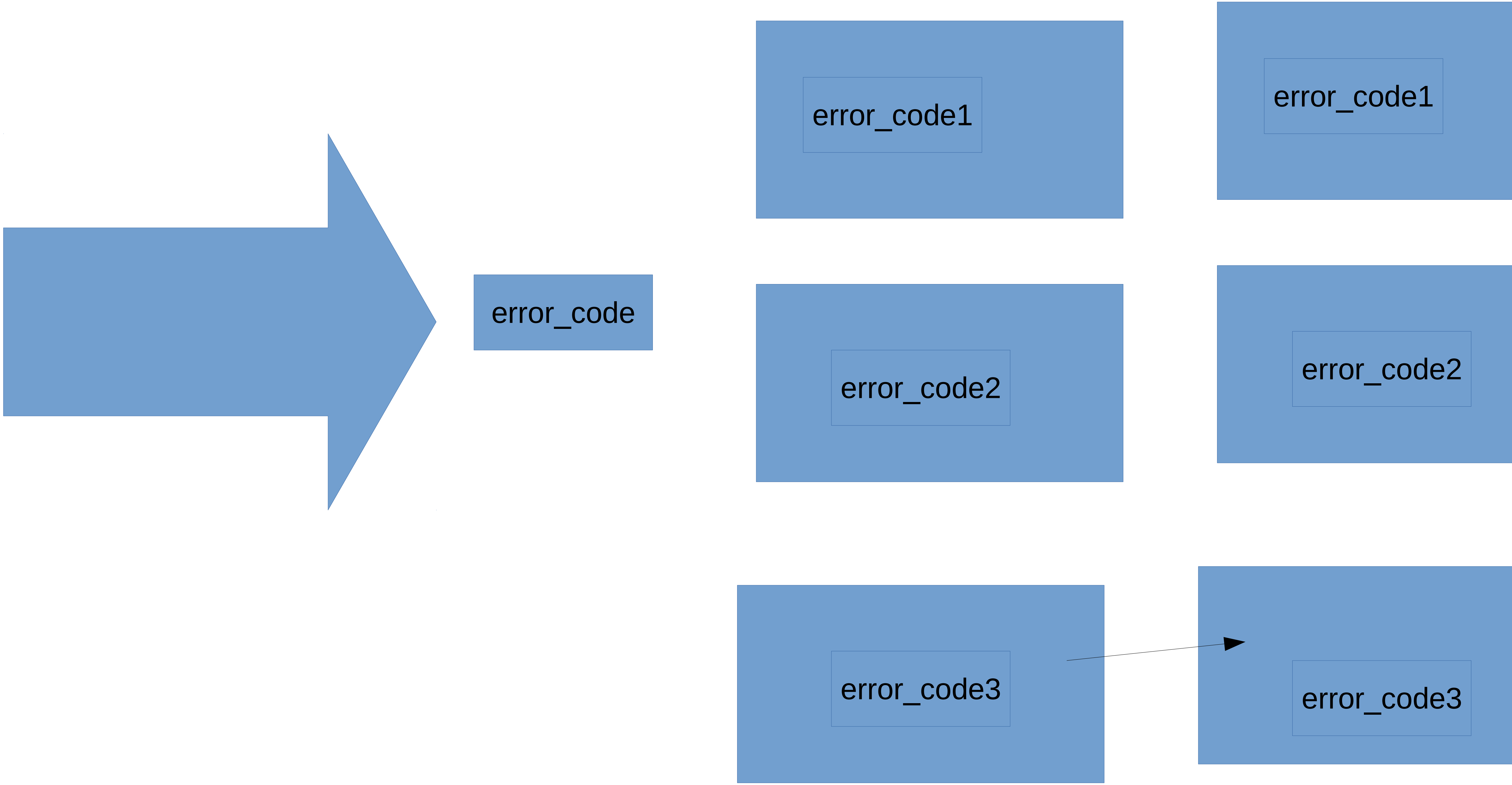
Data2


s2
s1



RDD1

RDD1



error_code

error_code1

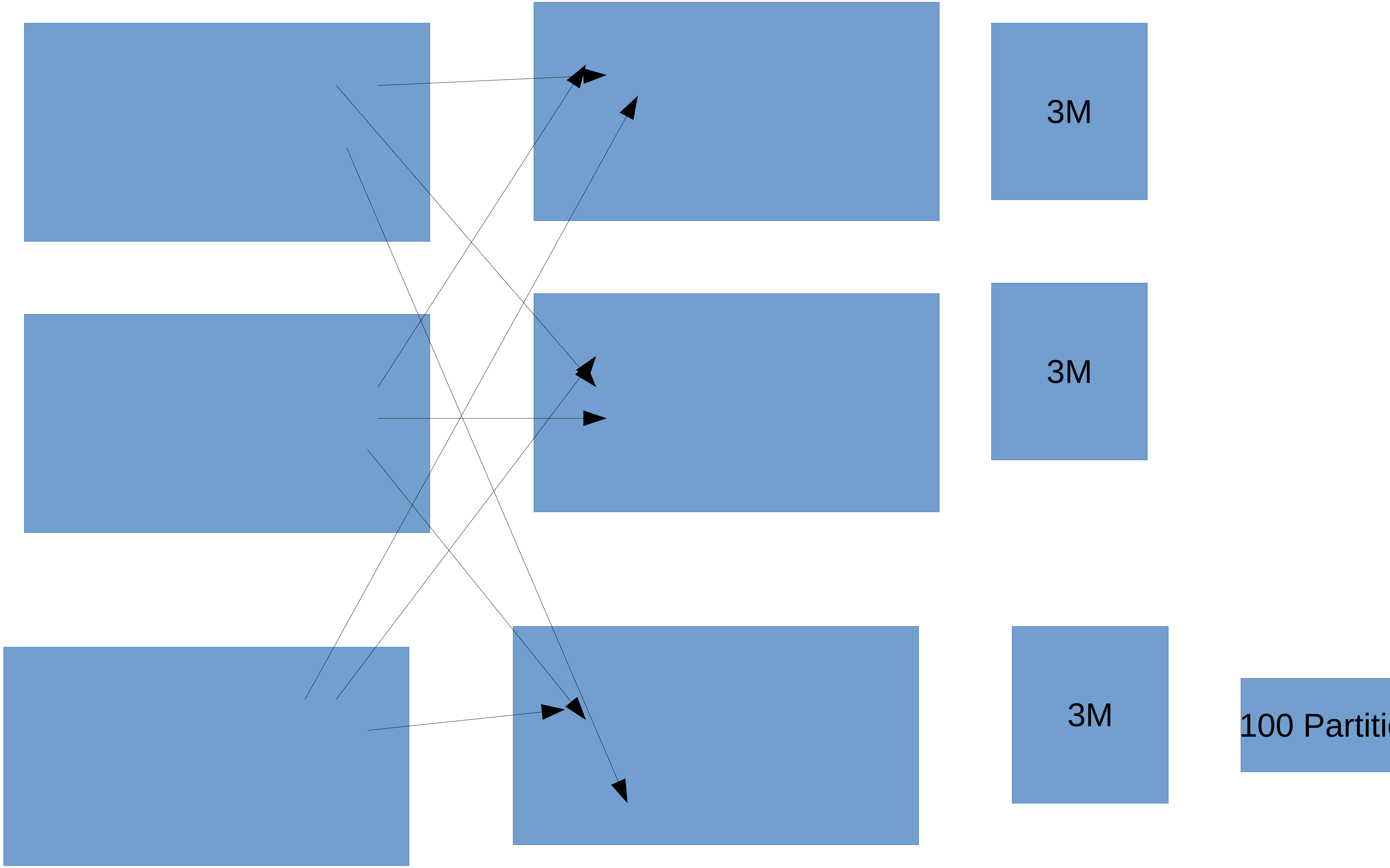
error_code1

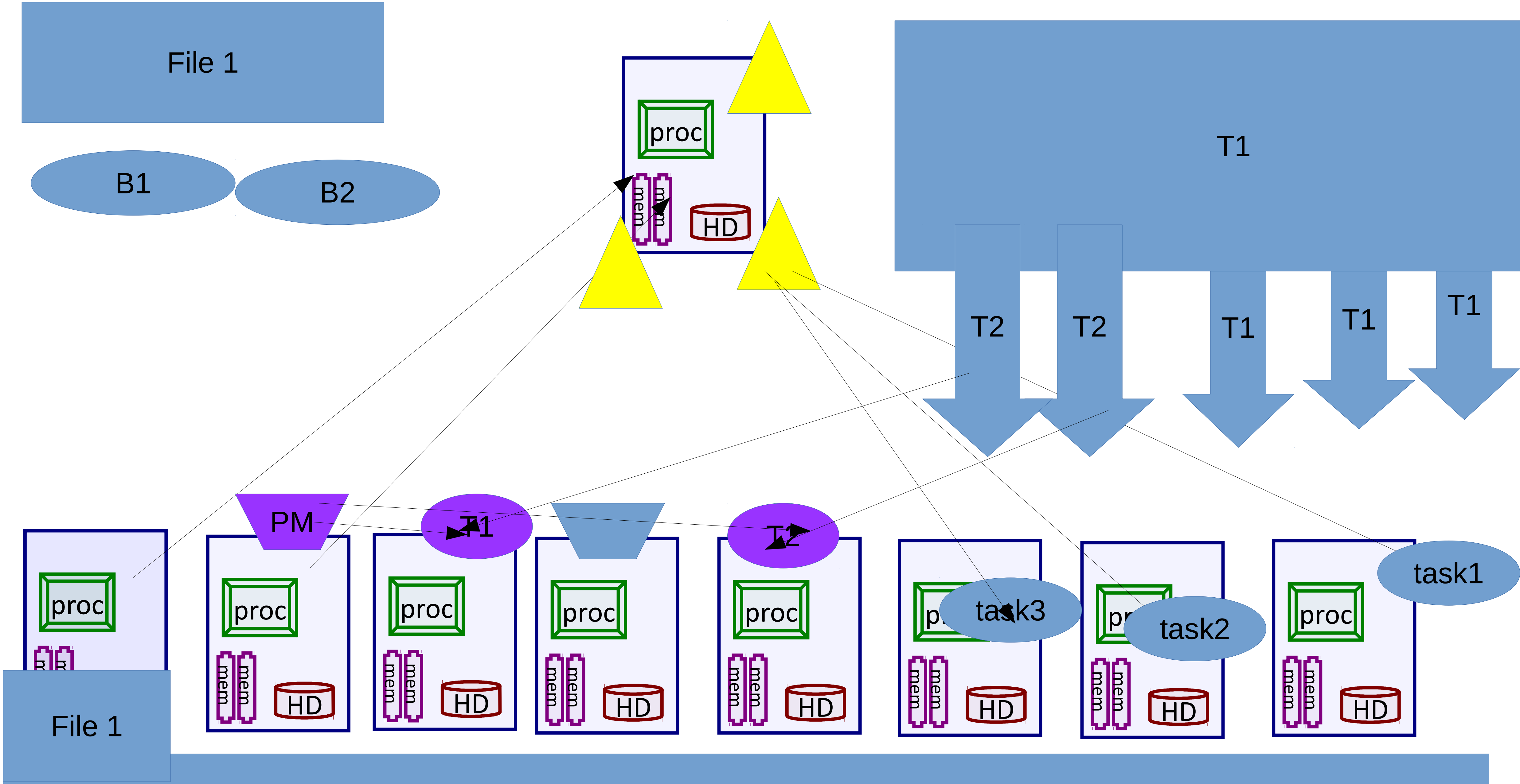
error_code2

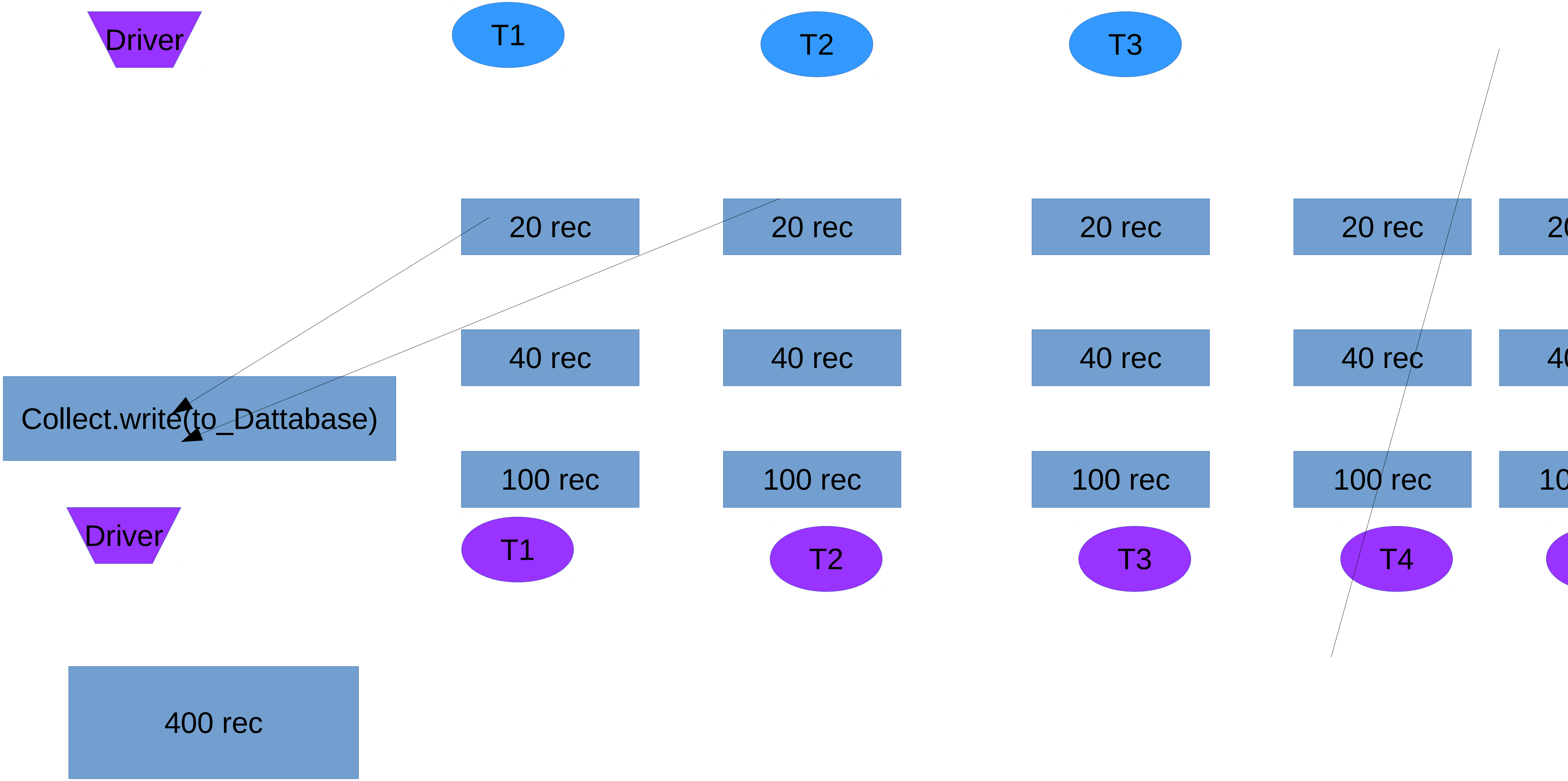
error_code2

error_code3

error_code3


3M
3M
3M
100 Partiti


File 1
B1
B2
proc
mem
mem
HD
T1
T2
T2
T1
T1
T1
PM
T1
T2
proc
proc
proc
proc
proc
task3
task2
task1
proc
mem
mem
HD
HD
HD
HD
HD
HD
HD
File 1



Driver

T1

T2

T3

20 rec

20 rec

20 rec

20 rec

40 rec

40 rec

40 rec

40 rec

Collect.write(to_Dattabase)

100 rec

100 rec

100 rec

100 rec

Driver

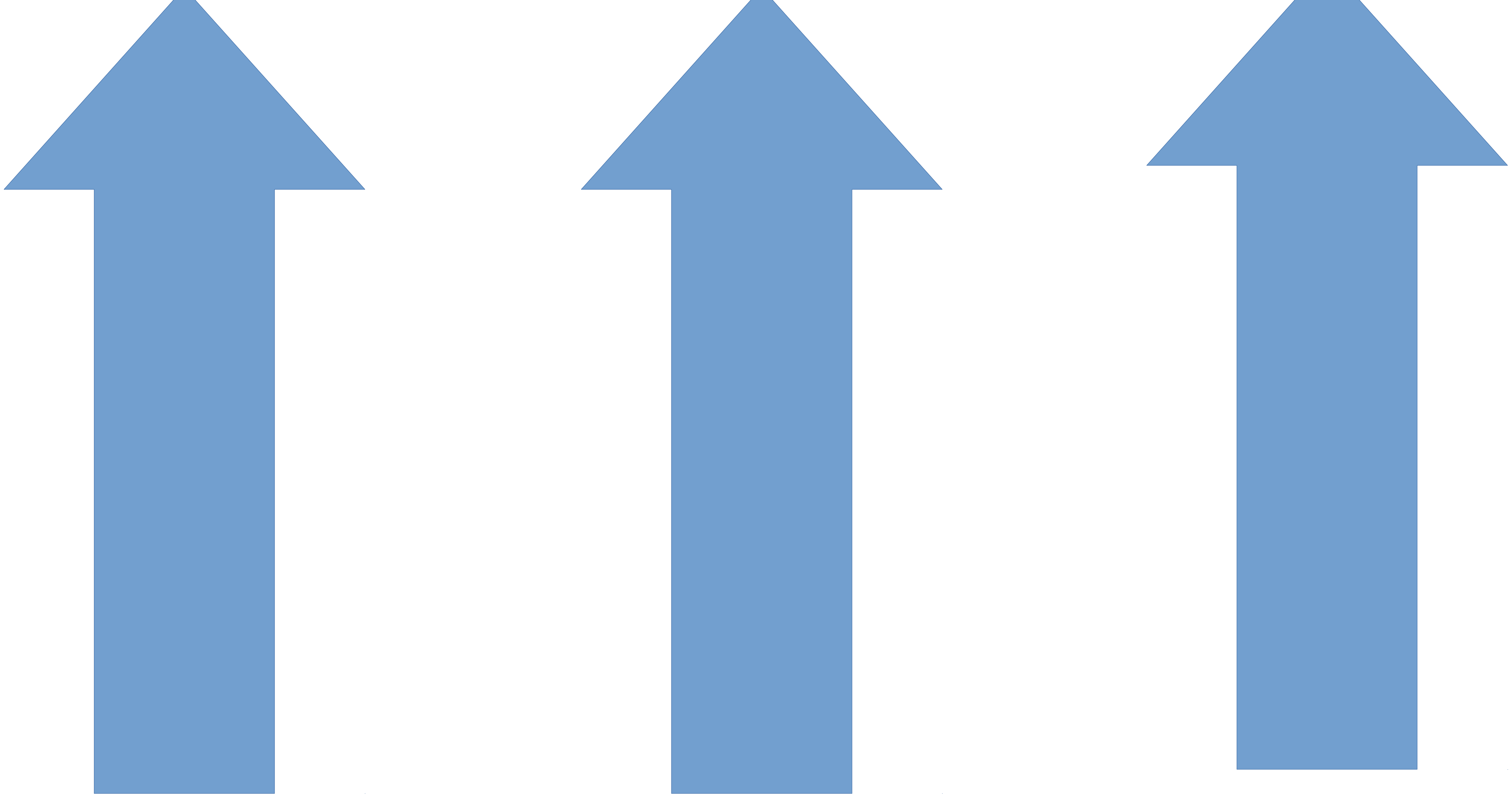
T1

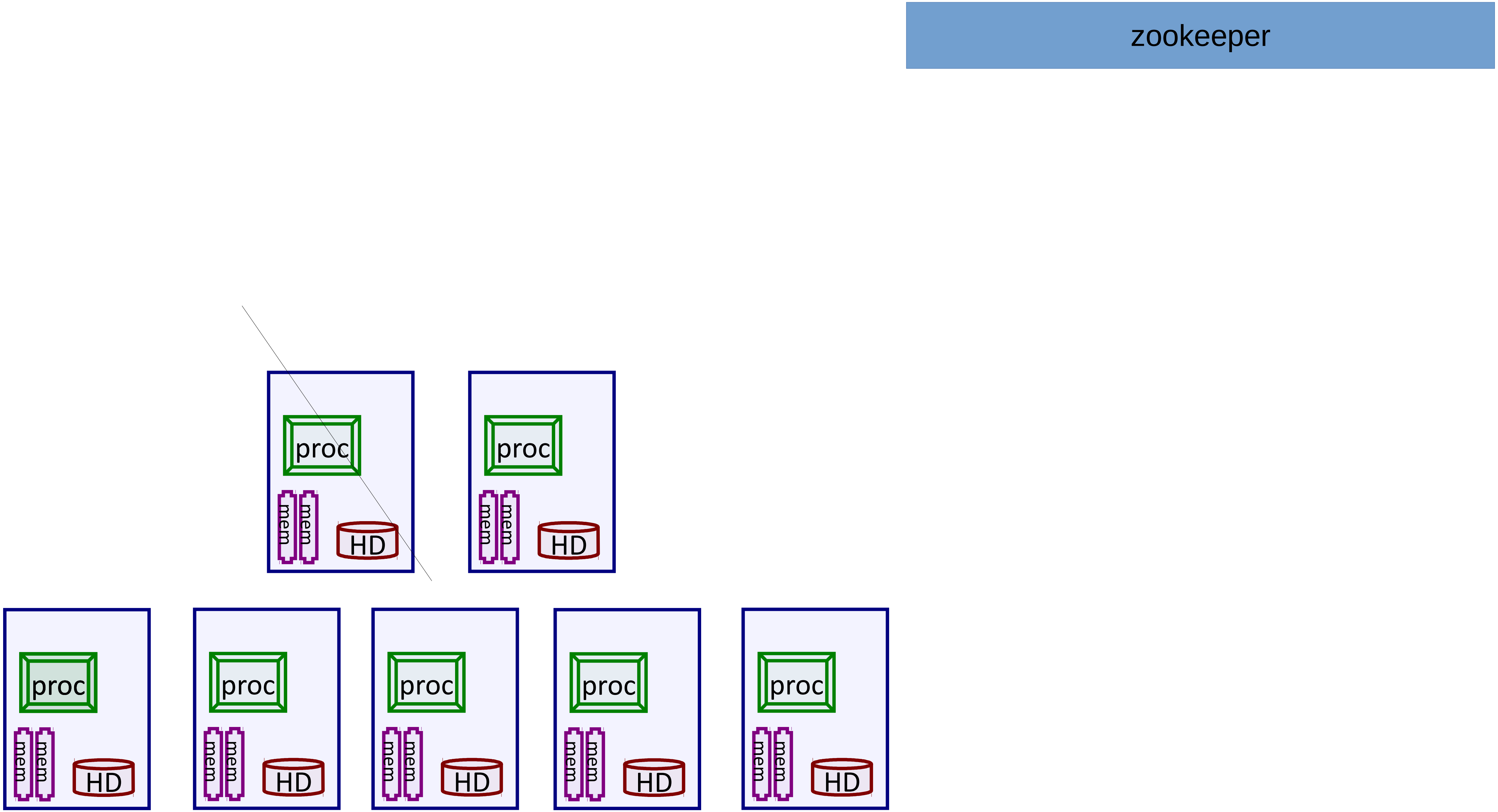
T2

T3

T4

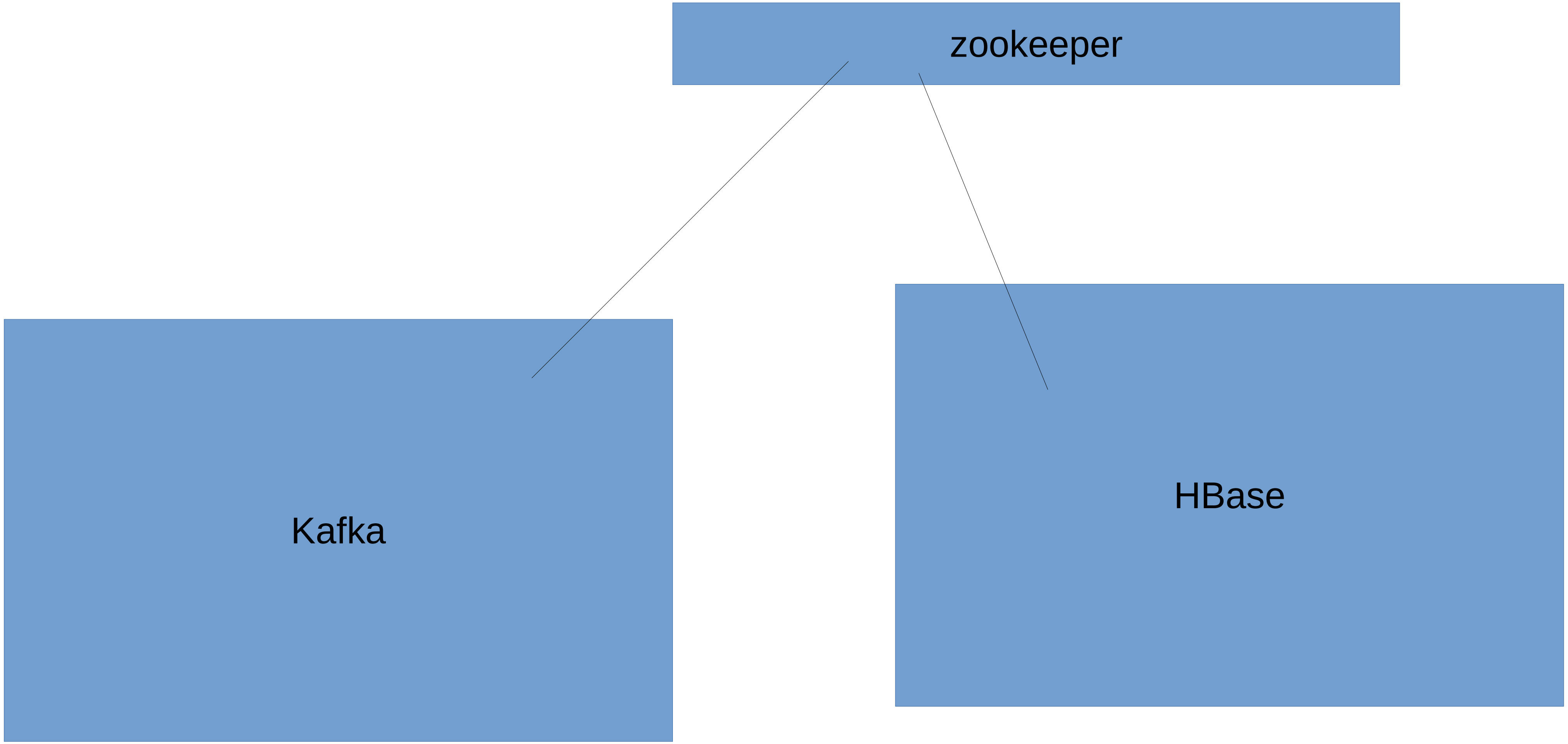
400 rec





zookeeper

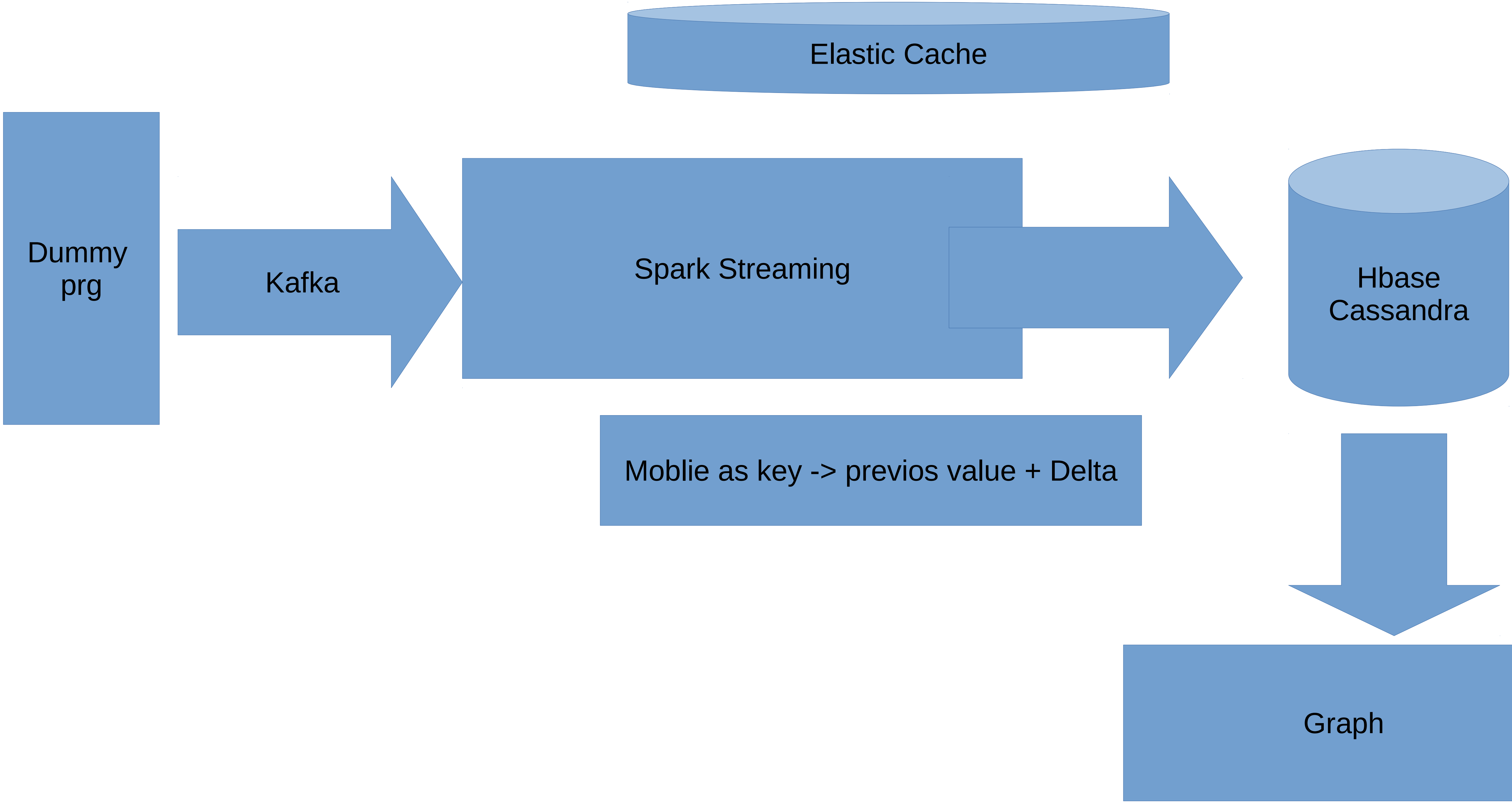
proc
mem
mem
HD
proc
mem
mem
HD
proc
mem
mem
HD
proc
mem
mem
HD
proc
mem
mem
HD
proc
mem
mem
HD
proc
mem
mem
HD



zookeeper

Kafka

HBase



Elastic Cache

Dummy prg

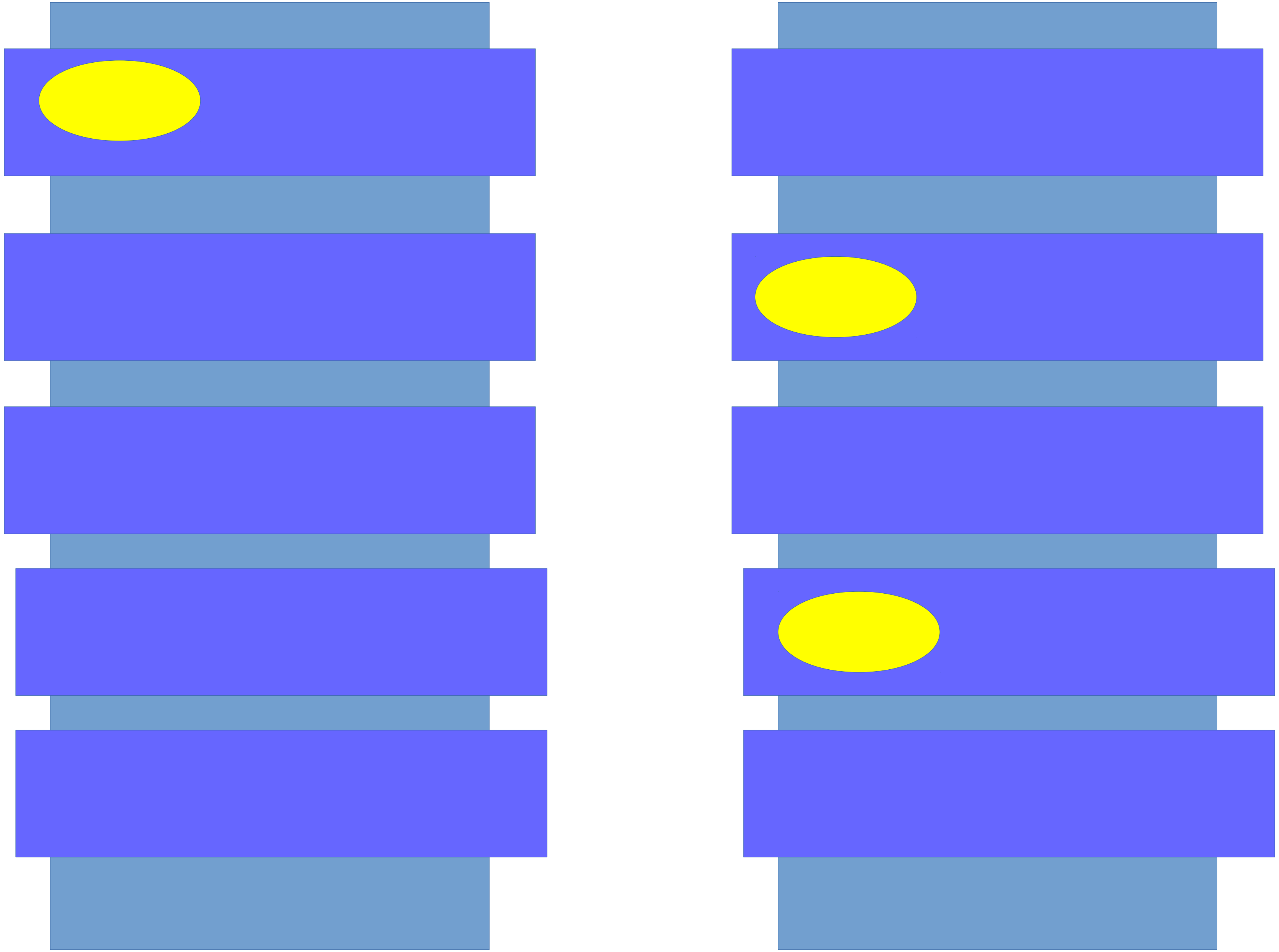
Kafka

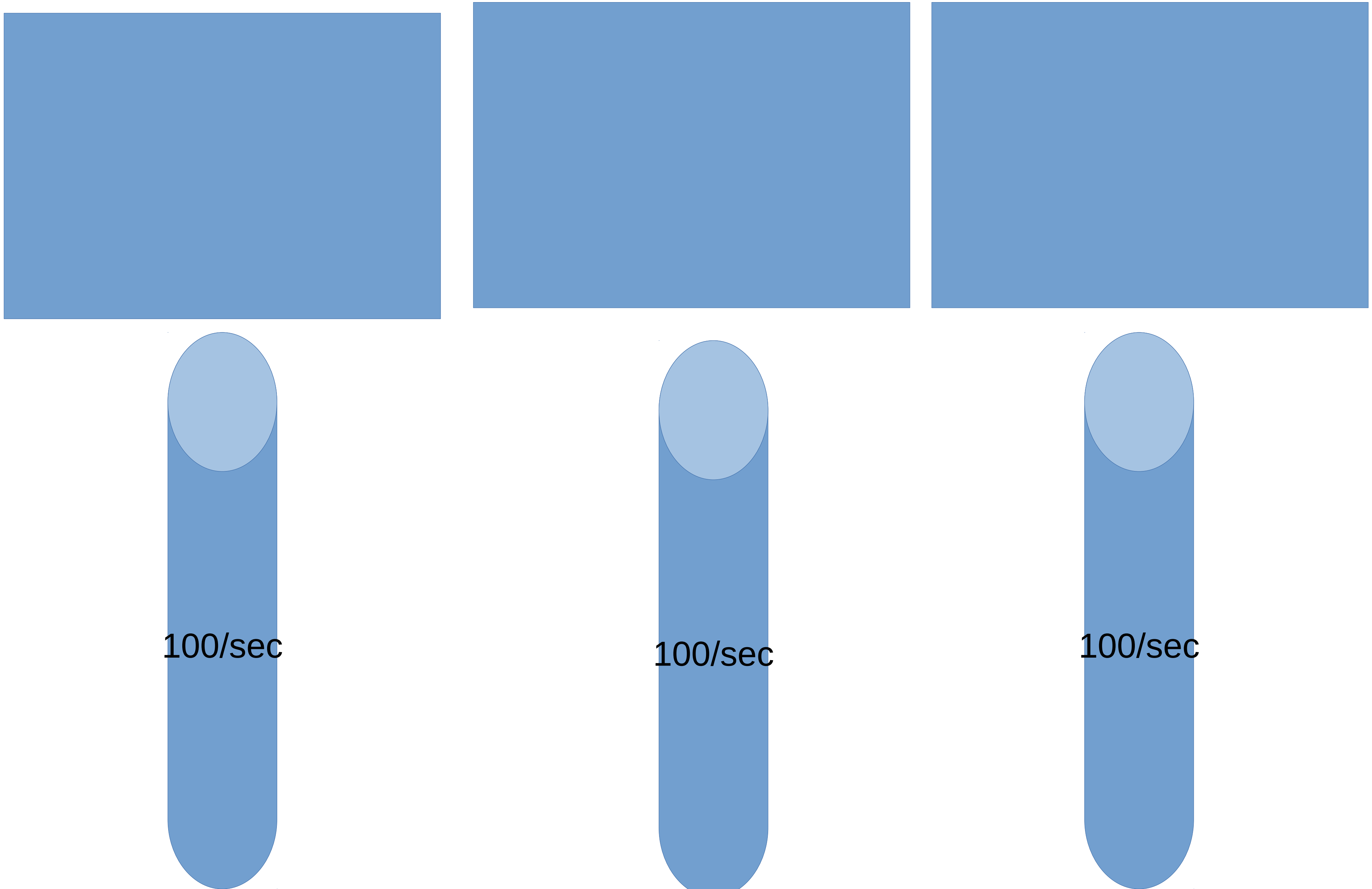
Spark Streaming

Hbase Cassandra

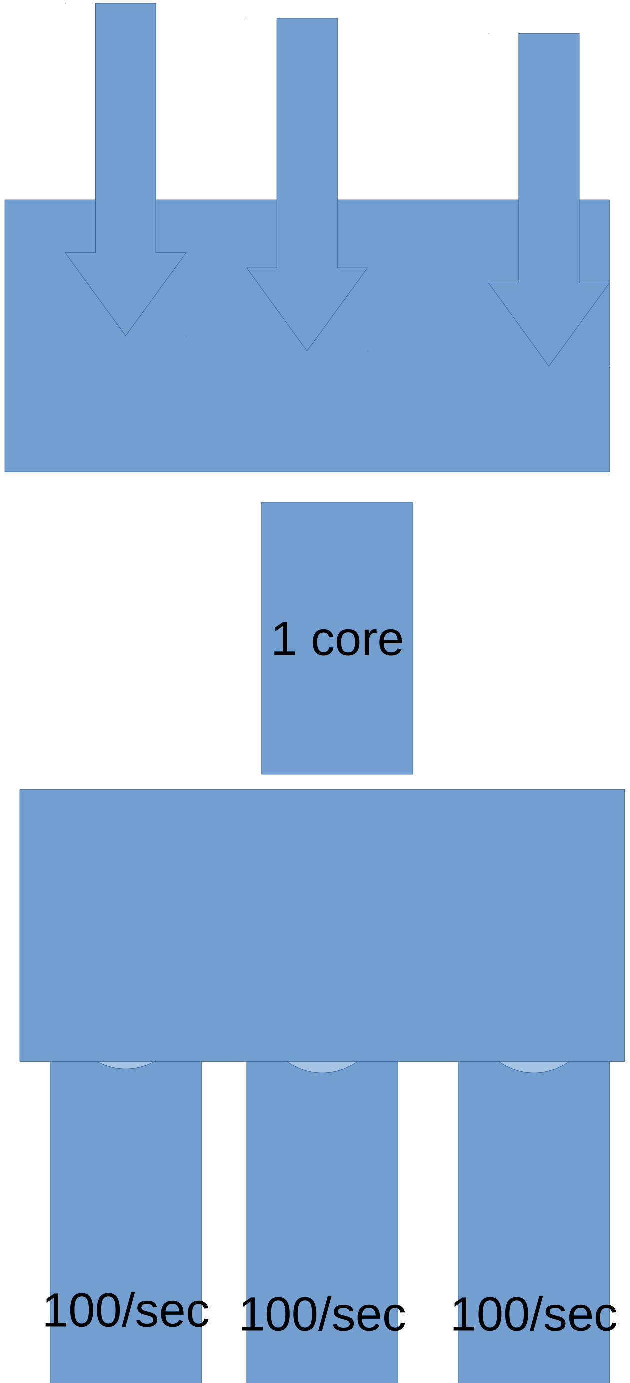
Moblie as key -> previos value + Delta

Graph

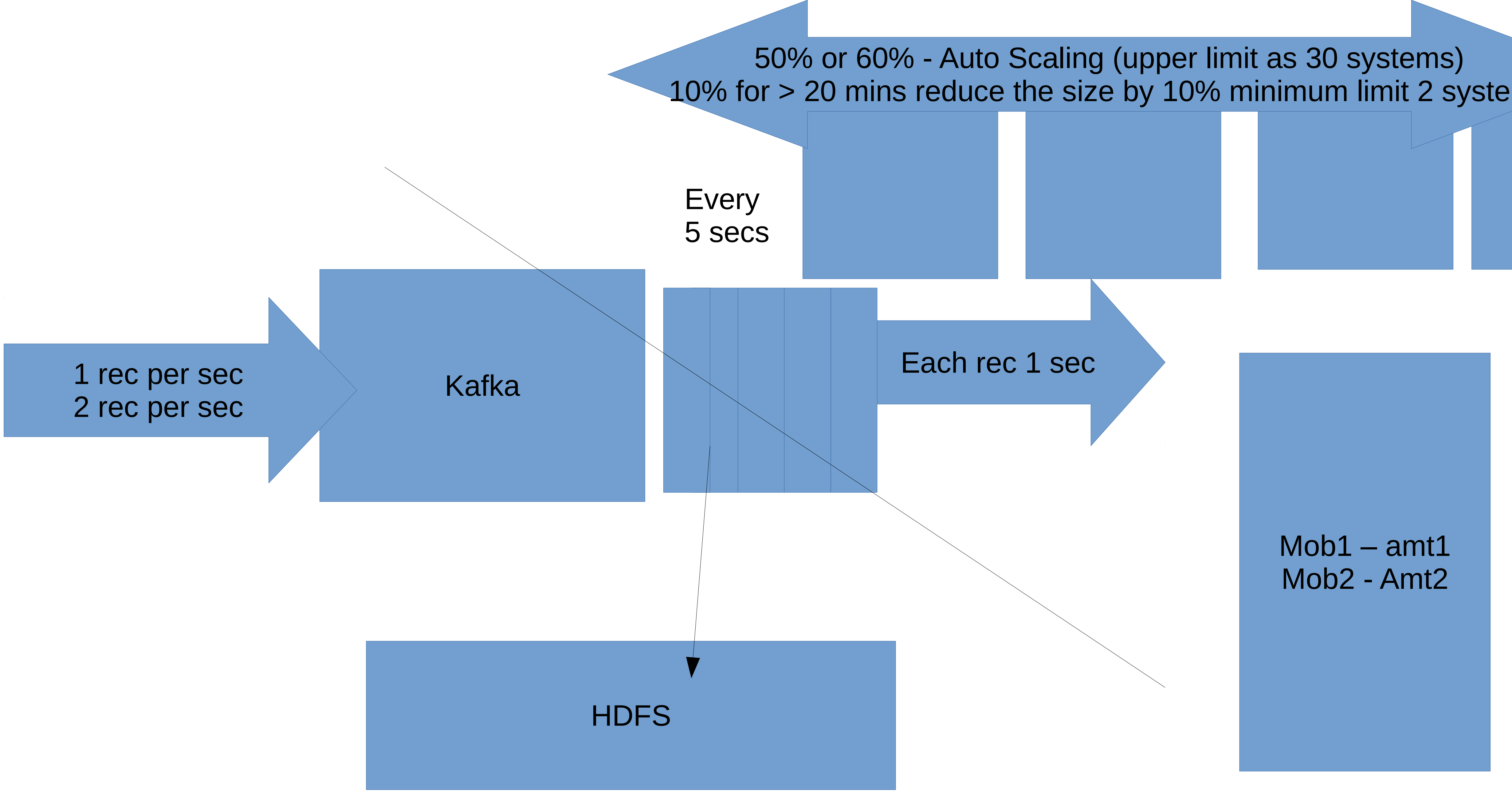




100/sec
100/sec
100/sec



1 core
100/sec
100/sec
100/sec



50% or 60% - Auto Scaling (upper limit as 30 systems)

10% for > 20 mins reduce the size by 10% minimum limit 2 syste

Every

5 secs

1 rec per sec

2 rec per sec

Kafka

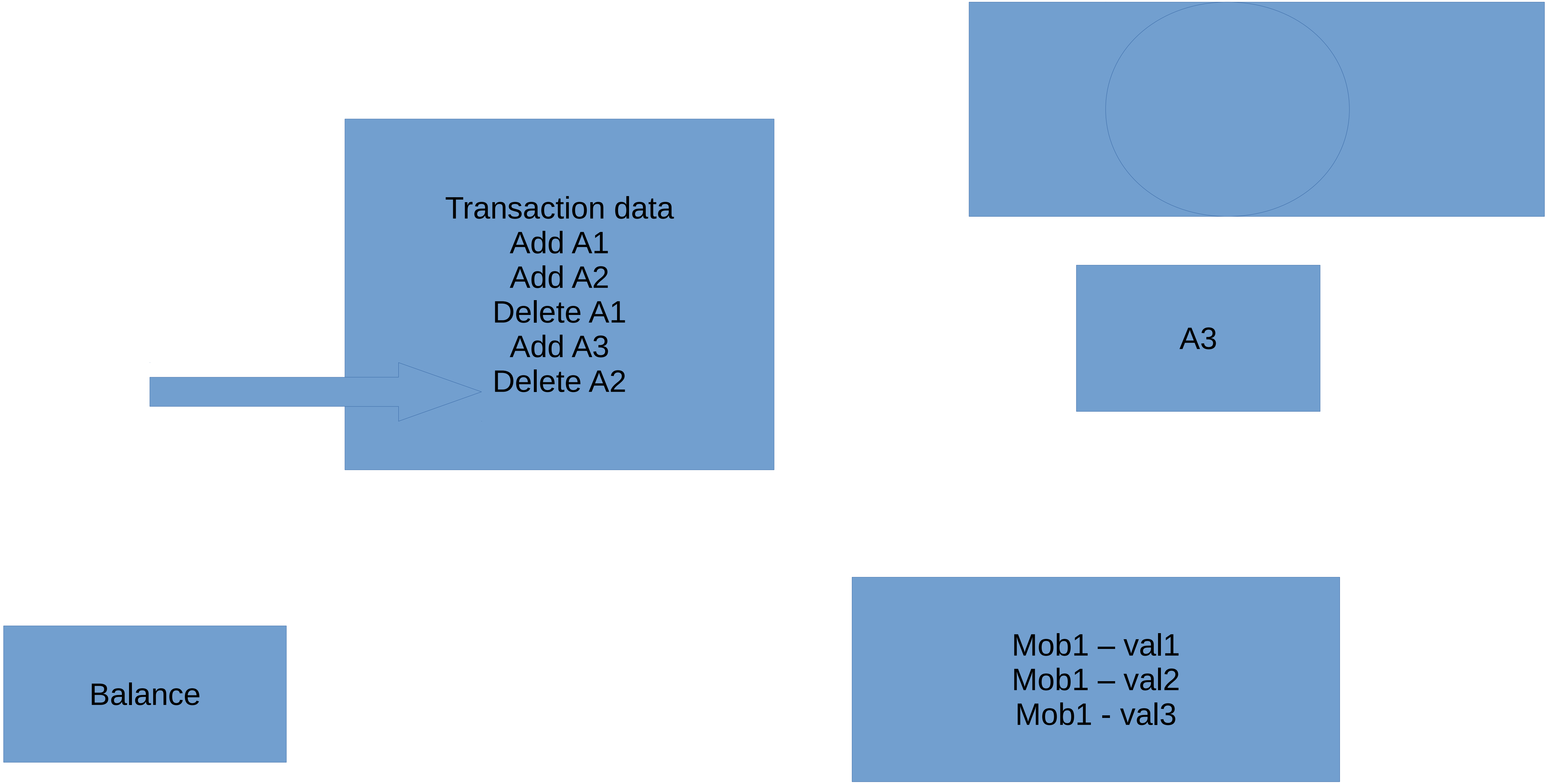
Each rec 1 sec

Mob1 – amt1

Mob2 - Amt2

HDFS

1) Amt of data in memory
3) Duration of microbatch
2) How frequent update

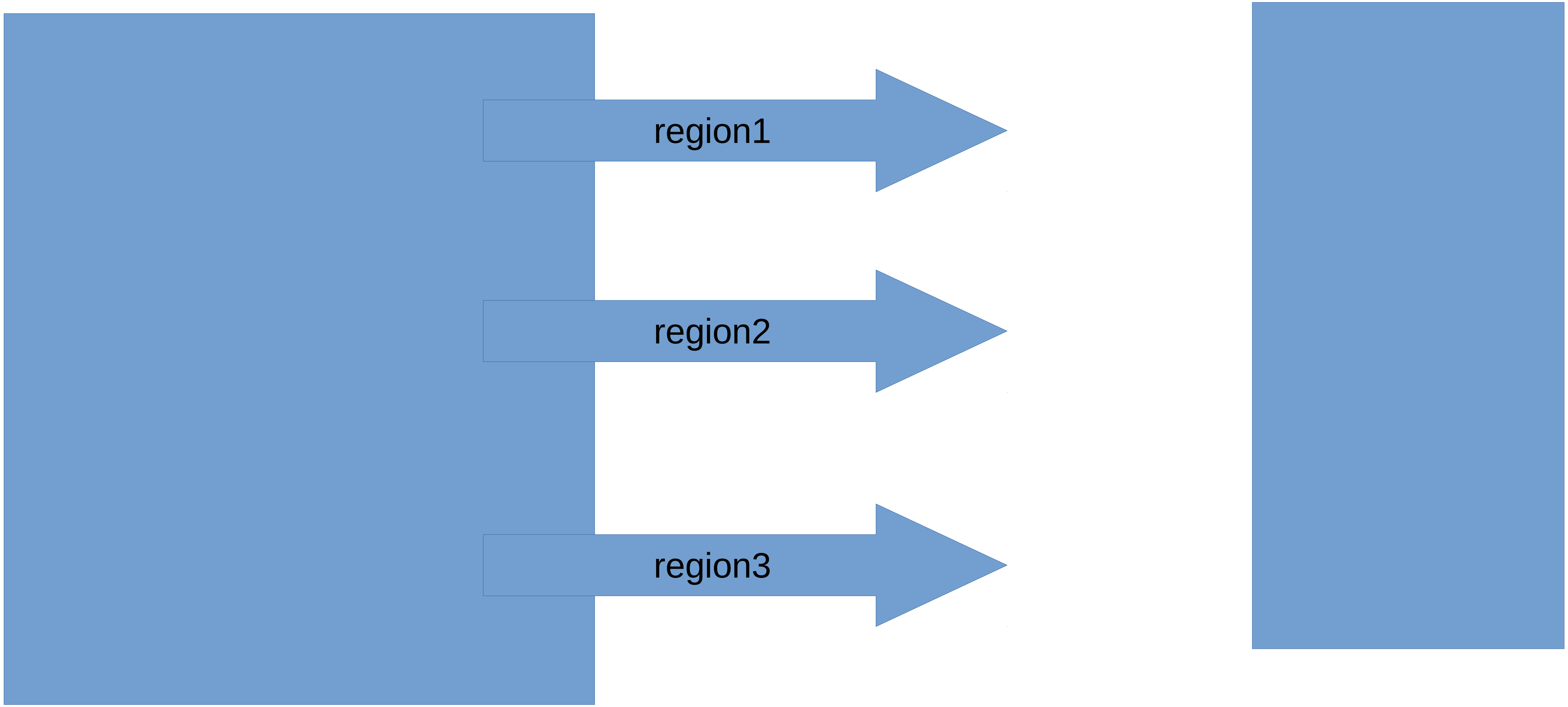


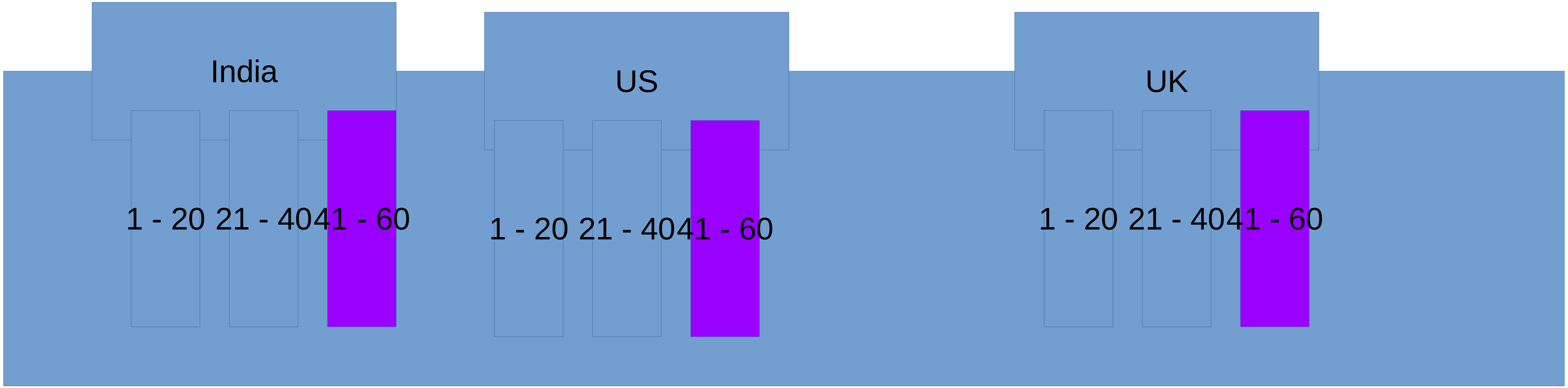
Transaction data
Add A1
Add A2
Delete A1
Add A3
Delete A2

A3

Balance

Mob1 – val1
Mob1 – val2
Mob1 - val3


region1
region2
region3



India

1 - 20

21 - 40

41 - 60

US

1 - 20

21 - 40

41 - 60

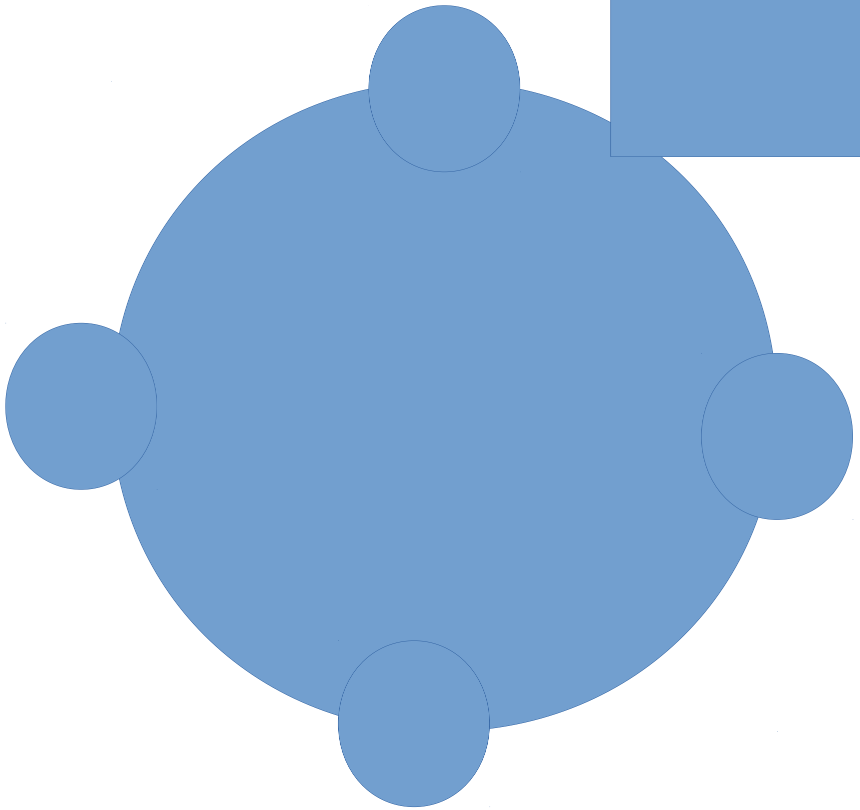
UK

1 - 20

21 - 40

41 - 60

1 – 3 – part1
4 – 6 - part2



Function(Data)

Data.subdata.element == 1
Return 1
Data.subdata.element > 10
Return 3

Return 0 - 3