CS 265

Stratos Idreos

BIG DATA SYSTEMS

NoSQL | Neural Networks | SQL | Graph | Data Science

learning outcome

fundamental of storage

learning outcome
fundamental of storage

data structures, SQL, NoSQL, Big Data, Neural Networks, Statistics, Data Science

Quick summary:

6-10 research lectures by Stratos,
2 reviews per week as of week 4 (or 5)
1 presentation in semester per student
individual systems or group research project (or both)

(papers and systems project to be released in ~2 weeks)

Prerequisites

knowledge of algorithms, data structures, hardware, systems

Research track: open to CS165 students

Systems track allows taking the class without all prerequisites

preparation

Get familiar with the very basics of traditional database architectures:
Architecture of a Database System. By J. Hellerstein, M. Stonebraker and J. Hamilton. Foundations and Trends in Databases, 2007

Get familiar with very basics of modern database architectures:
The Design and Implementation of Modern Column-store Database Systems. By D. Abadi, P. Boncz, S. Harizopoulos, S. Idreos, S. Madden. Foundations and Trends in Databases, 2013

Get familiar with the very basics of modern large scale systems:
Massively Parallel Databases and MapReduce Systems. By Shivnath Babu and Herodotos Herodotou. Foundations and Trends in Databases, 2013

first reading

The Periodic Table of Data Structures.
Stratos Idreos, Kostas Zoumpatianos, Manos Athanassoulis, Niv Dayan, Brian Hentschel, Michael S. Kester, Demi Guo, Lukas Maas, Wilson Qin, Abdul Wasay, Yiyou Sun.
IEEE Data Engineering Bull. Sep, 2018

DASlab
@ Harvard SEAS

Check out: syllabus, preparation readings, project 0, systems project, online sections

http://daslab.seas.harvard.edu/classes/cs265/

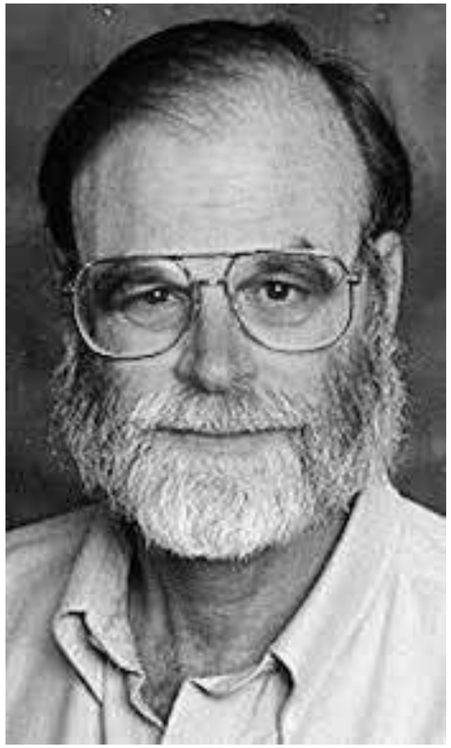
(+cs165 material slides/readings/sections)

Check out: syllabus, preparation readings, project 0, systems project, online sections

http://daslab.seas.harvard.edu/classes/cs265/

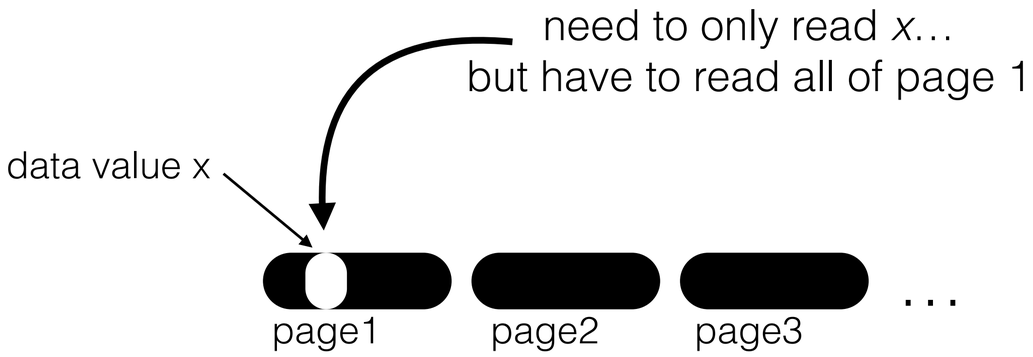
(+cs165 material slides/readings/sections)

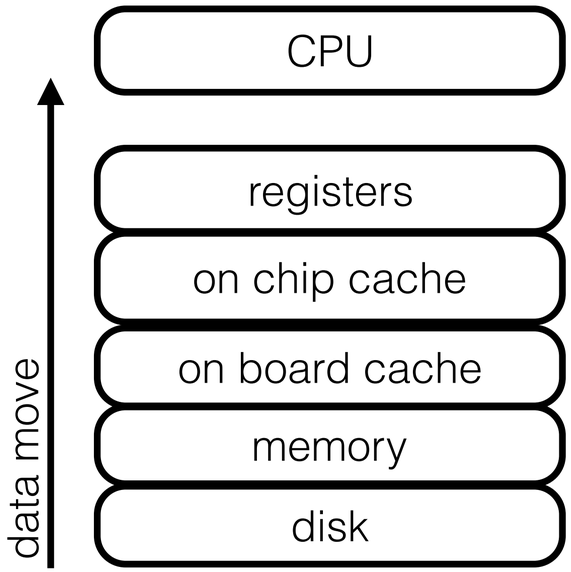
questions on logistics?

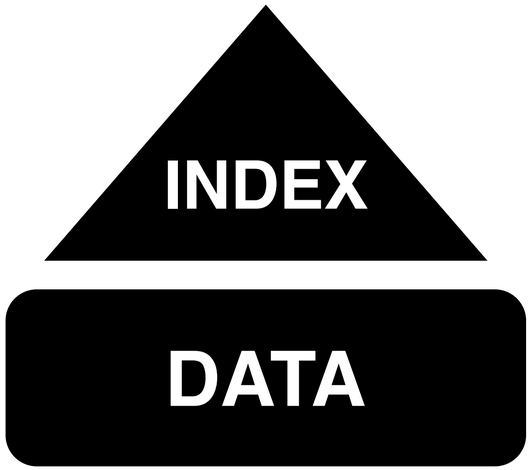


Jim Gray, IBM, Tandem, DEC, Microsoft
ACM Turing award
ACM SIGMOD Edgar F. Codd Innovations award

registers	**my head** ~0
2x on chip cache	**this room** 1 min
10x on board cache	**this building** 10 min
100x memory	**New York** 1.5 hours
100Kx disk	**Pluto** 2 years

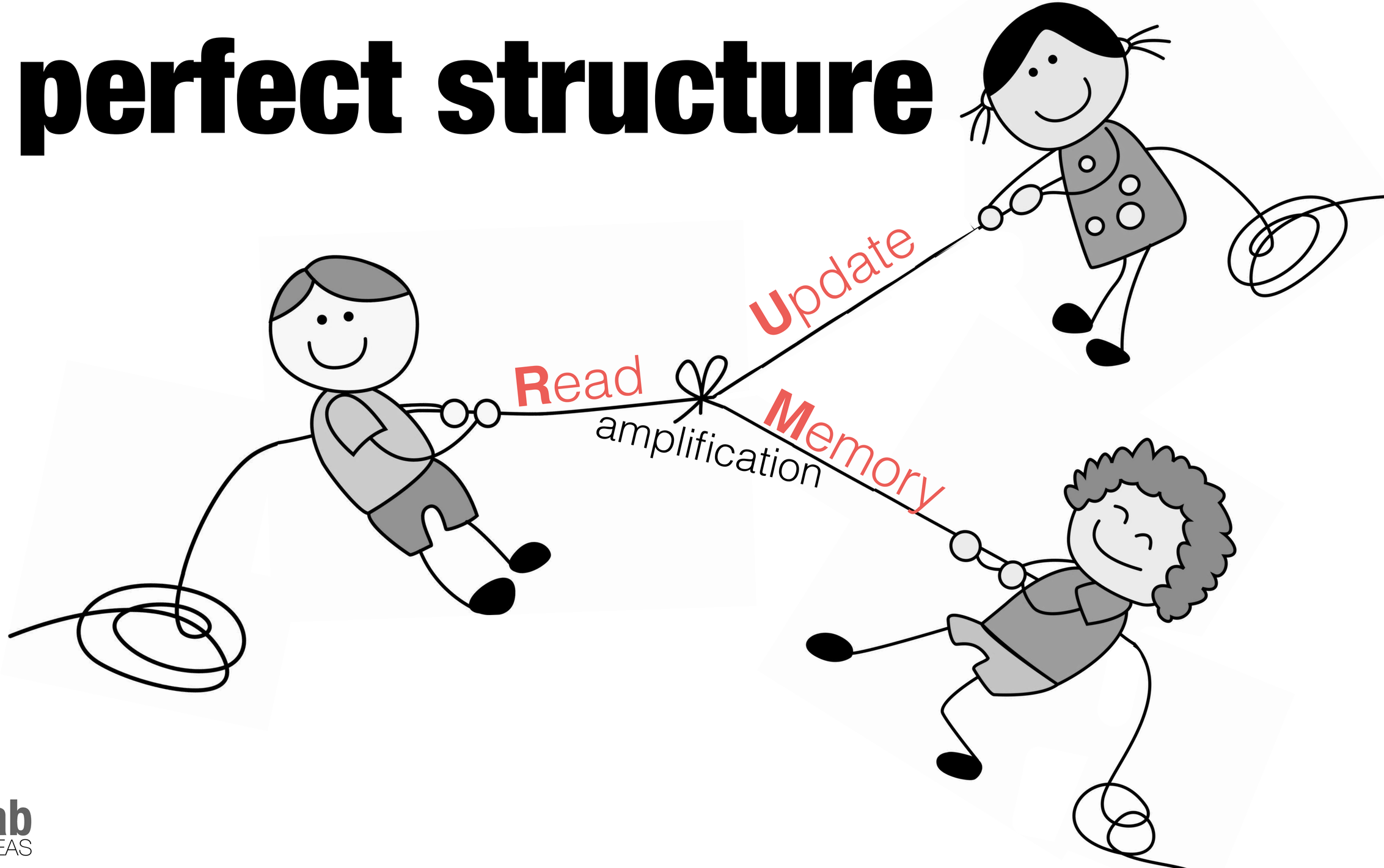

need to only read *x*…
but have to read all of page 1
data value x
page1
page2
page3
…


CPU
registers
on chip cache
on board cache
memory
disk
data move



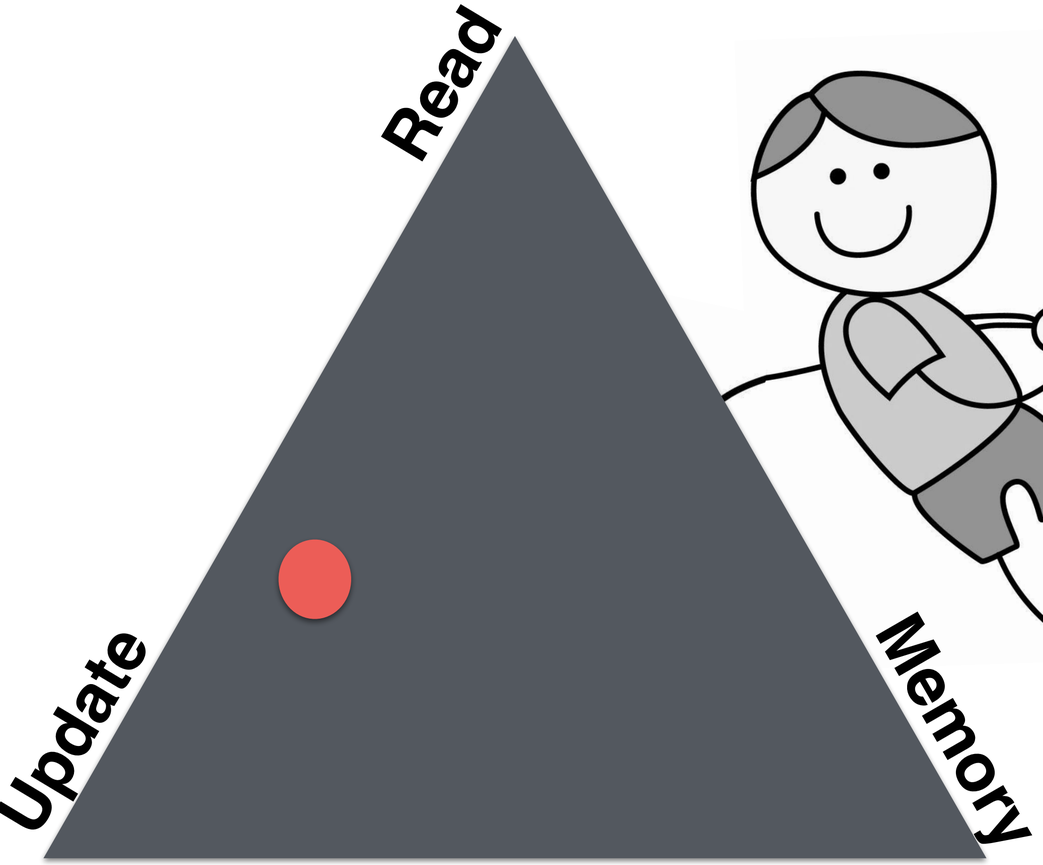
HOW
TO STORE
DATA

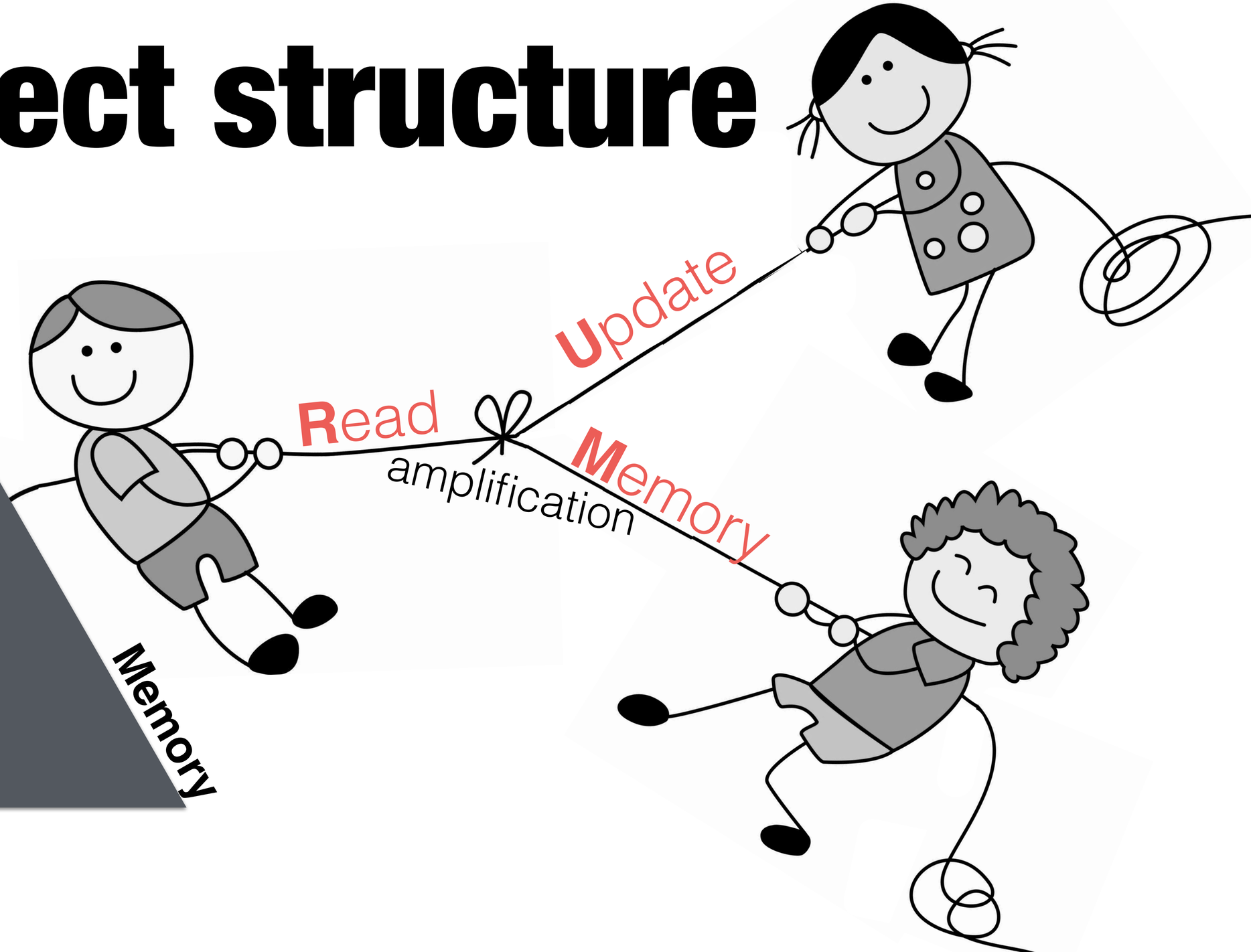
no perfect structure



DASlab
@ Harvard SEAS

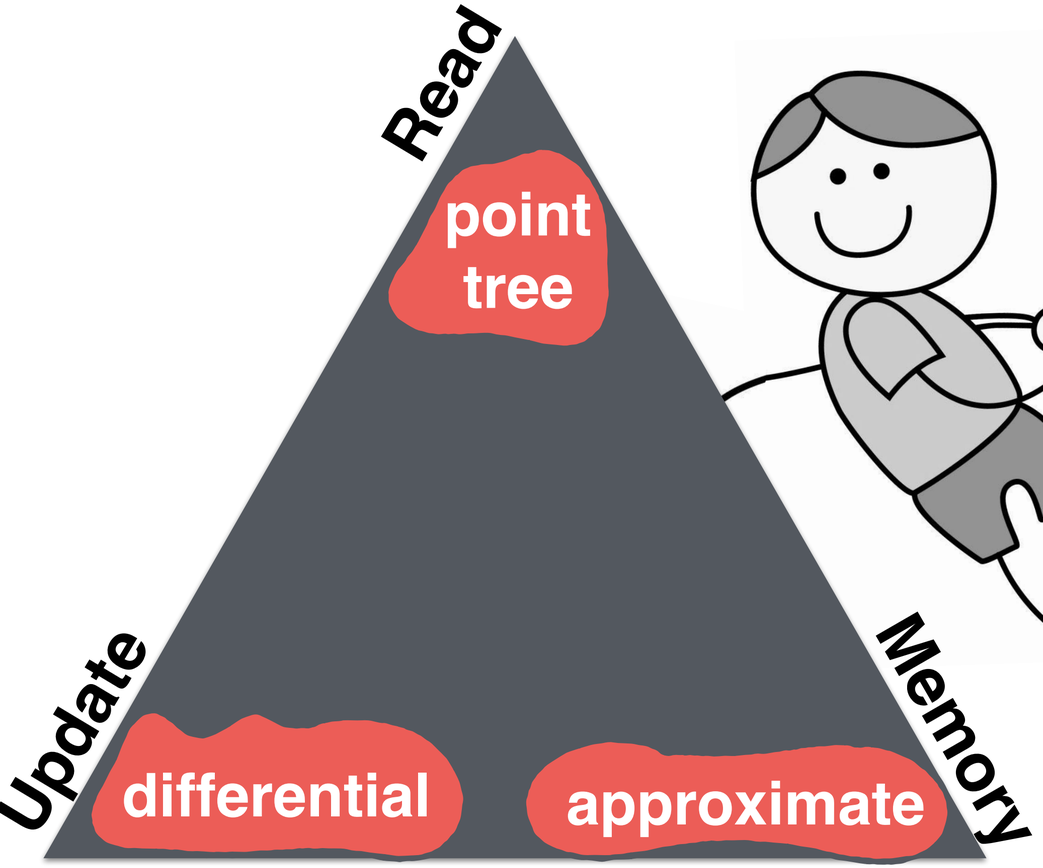
no perfect structure

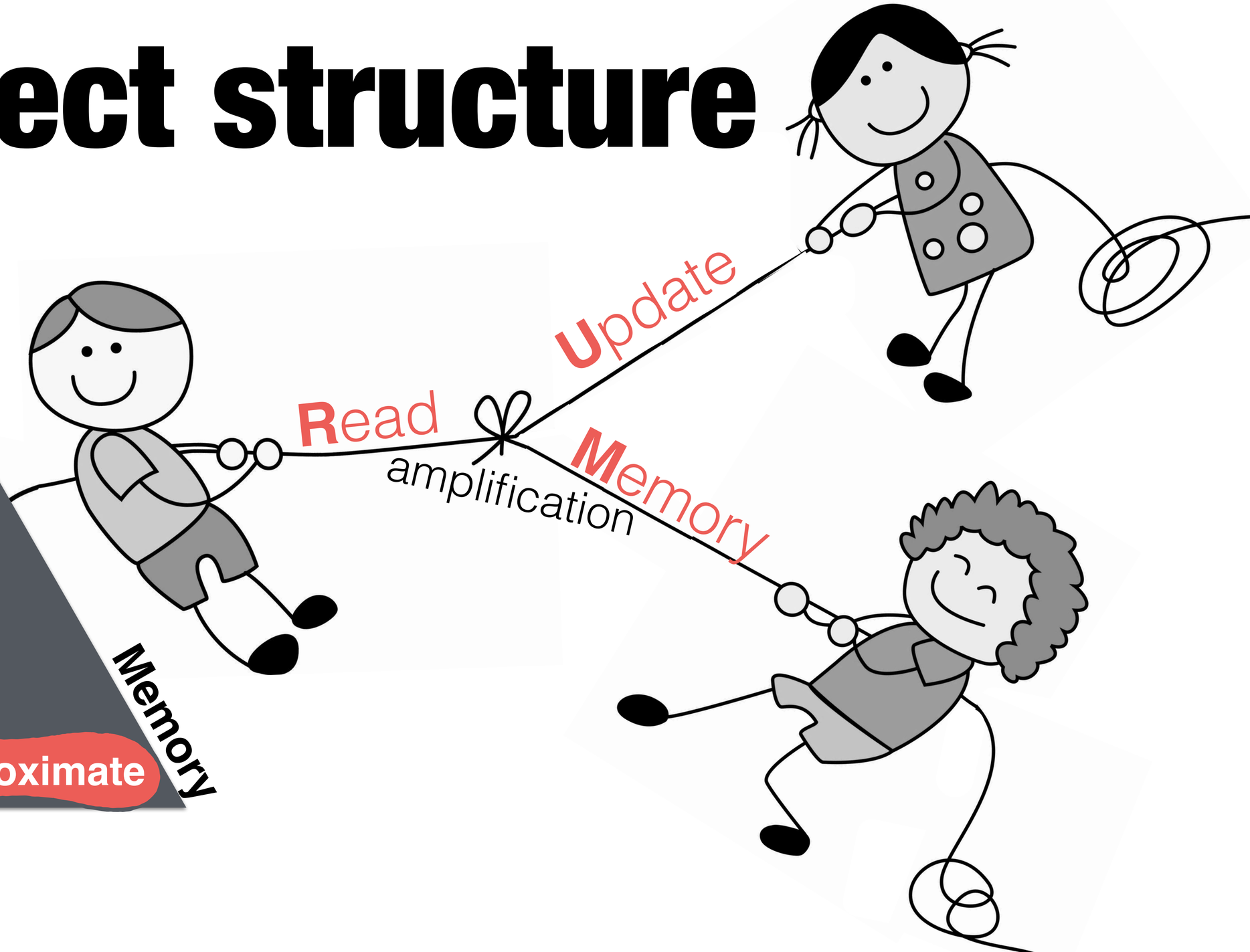




DASlab
@ Harvard SEAS

no perfect structure



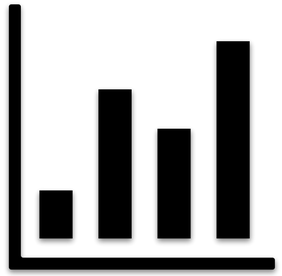


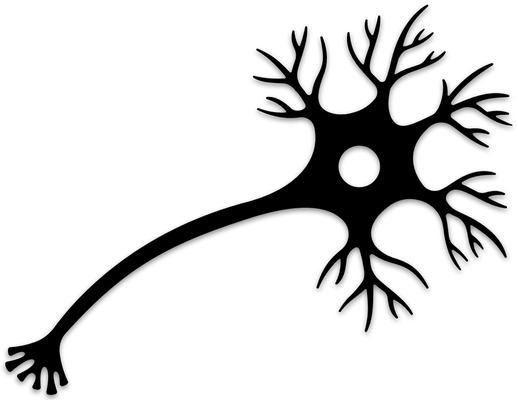
DASlab
@ Harvard SEAS

How do I make my **data system** run x times as fast?  (sql,nosql,bigdata, …)

How do I minimize my **bill** in the **cloud**?

How do I extend the **lifetime** of my hardware? 

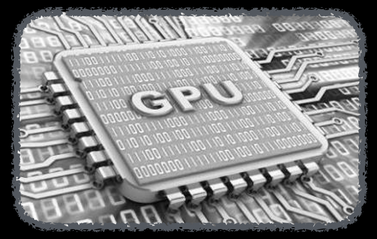
How to accelerate **statistics** computation for data science/ML? 

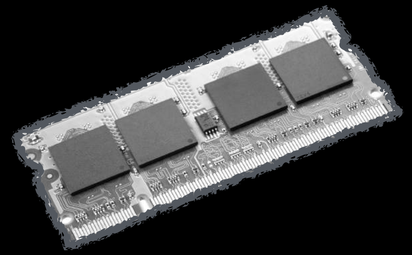


How do I train my **neural network** x times faster?

DASlab @ Harvard SEAS

Research/Systems Intro

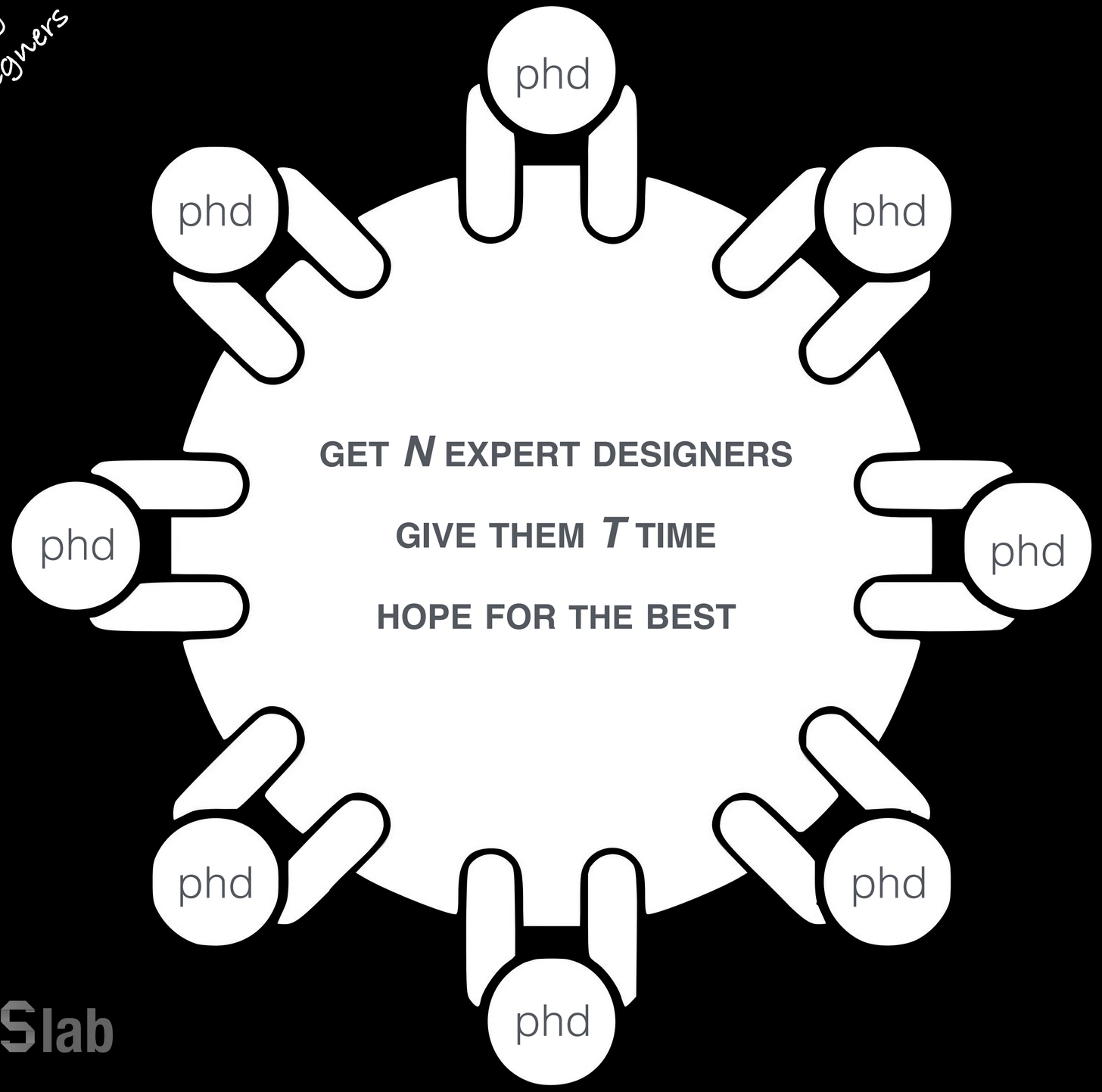




~7-10 lectures: Periodic Table, Calculator, NoSQL. Then Statistics, Neural Networks

DASlab @ Harvard SEAS

the dinning systems designers



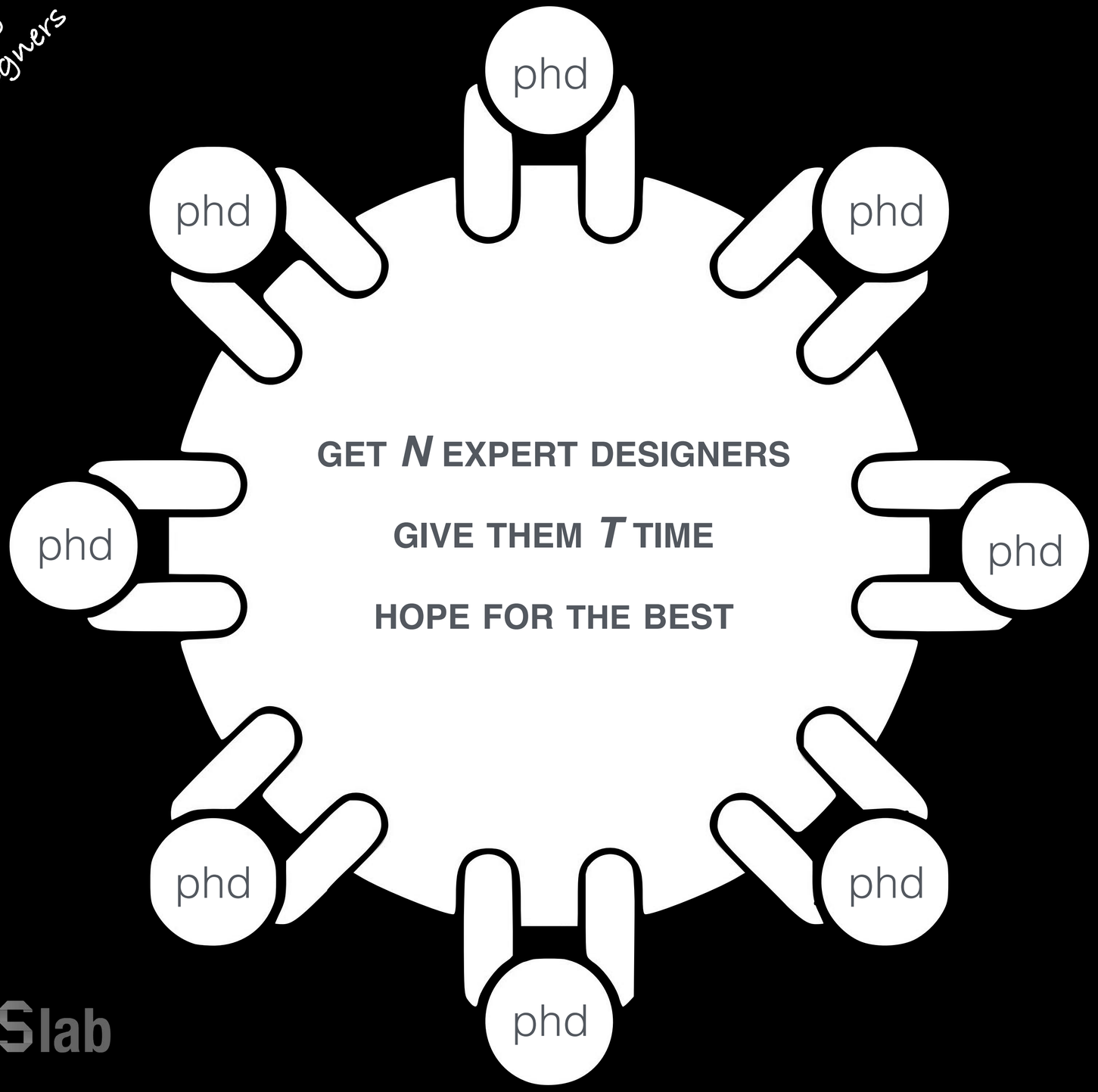
GET *N* EXPERT DESIGNERS

GIVE THEM *T* TIME

HOPE FOR THE BEST

DASlab

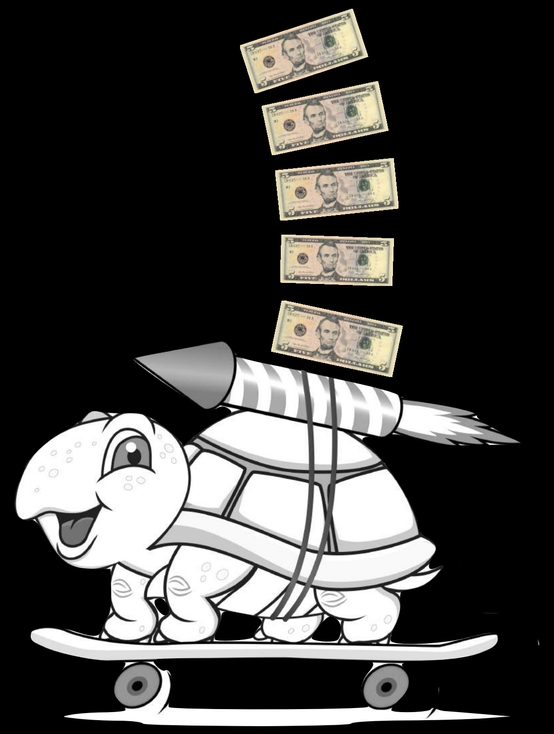
the dinning systems designers



GET *N* EXPERT DESIGNERS

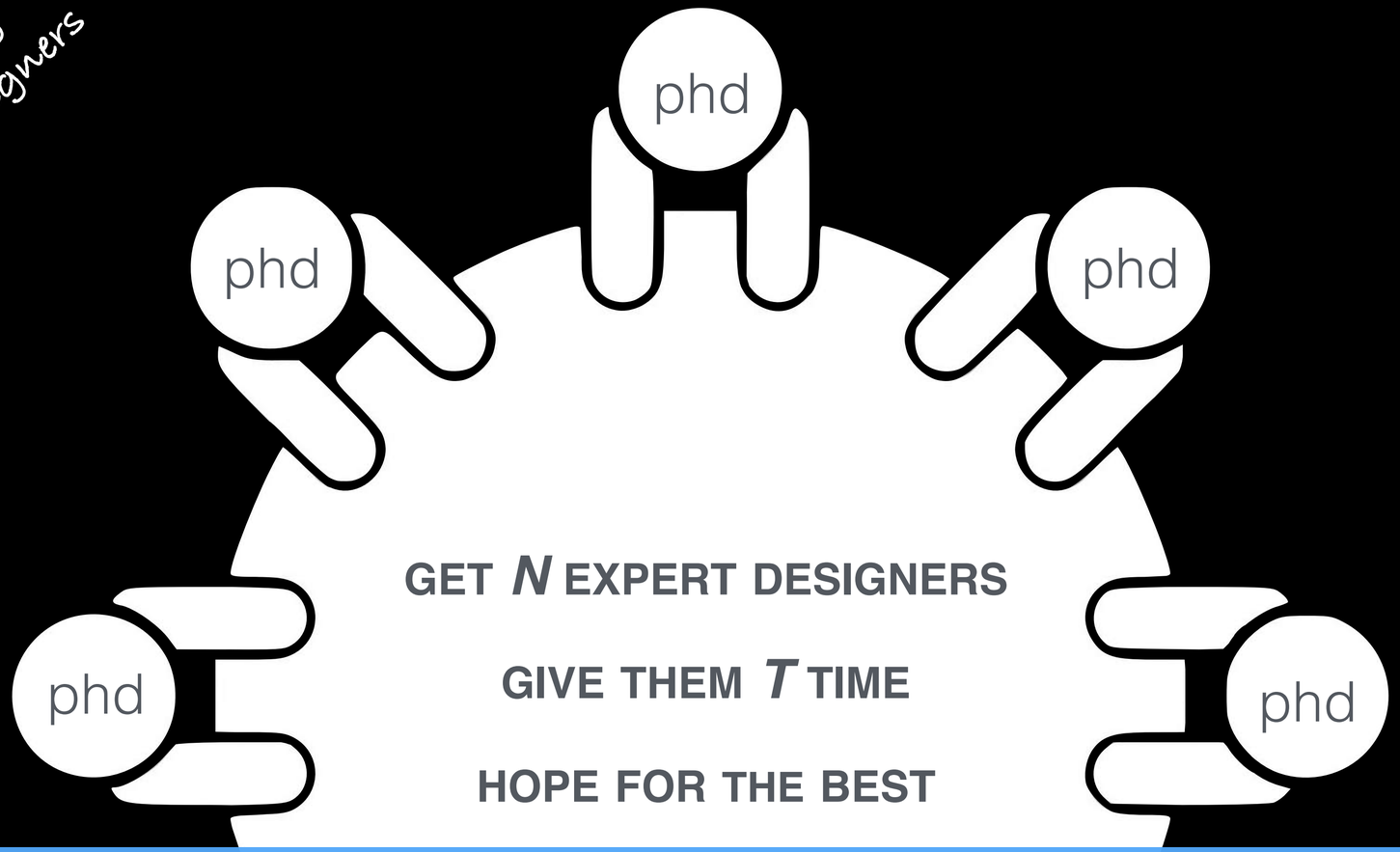
GIVE THEM *T* TIME

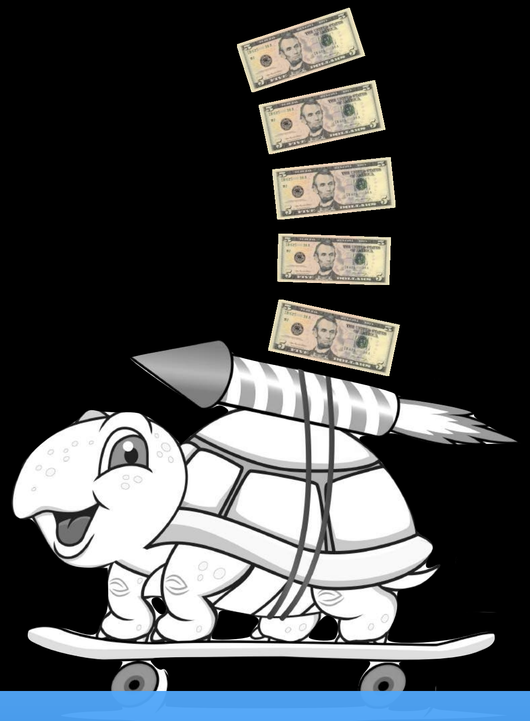
HOPE FOR THE BEST



DASlab

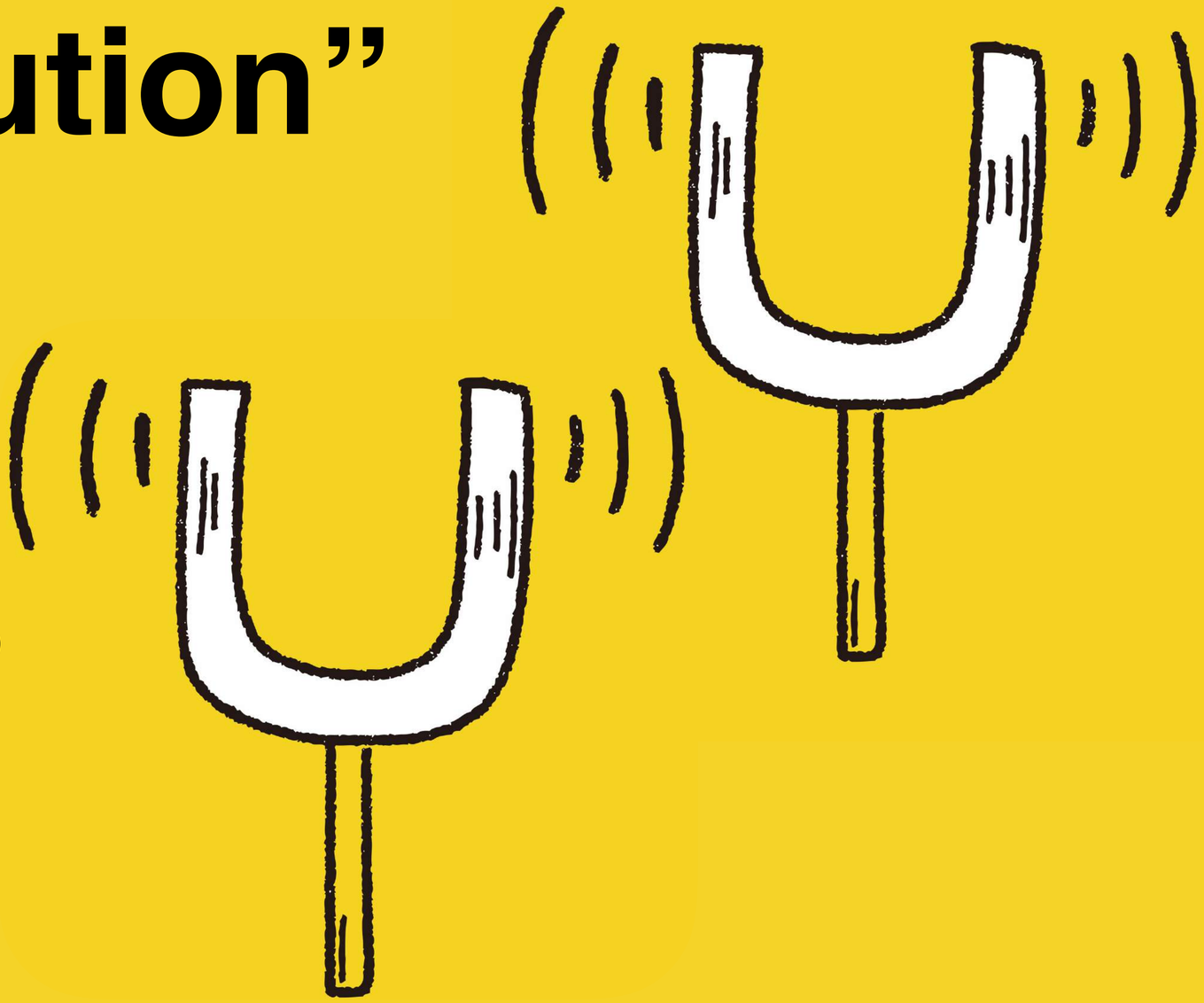
the dinning systems designers





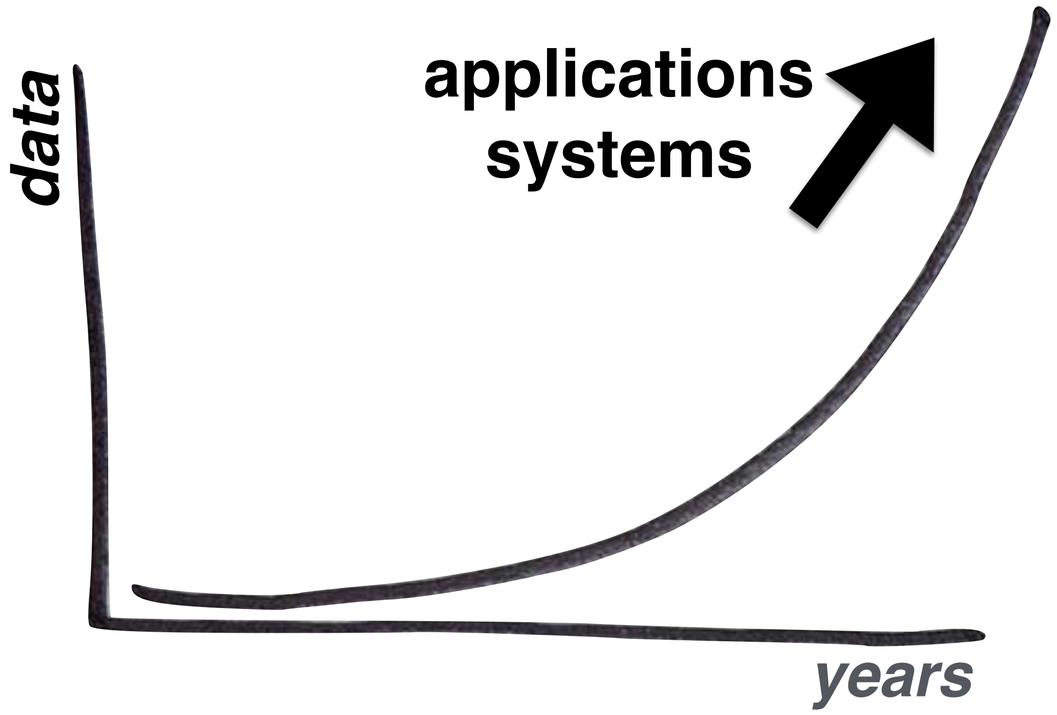
THE HIPPO METHOD
"HIGHEST PAID PERSON'S OPINION"

standard “solution”

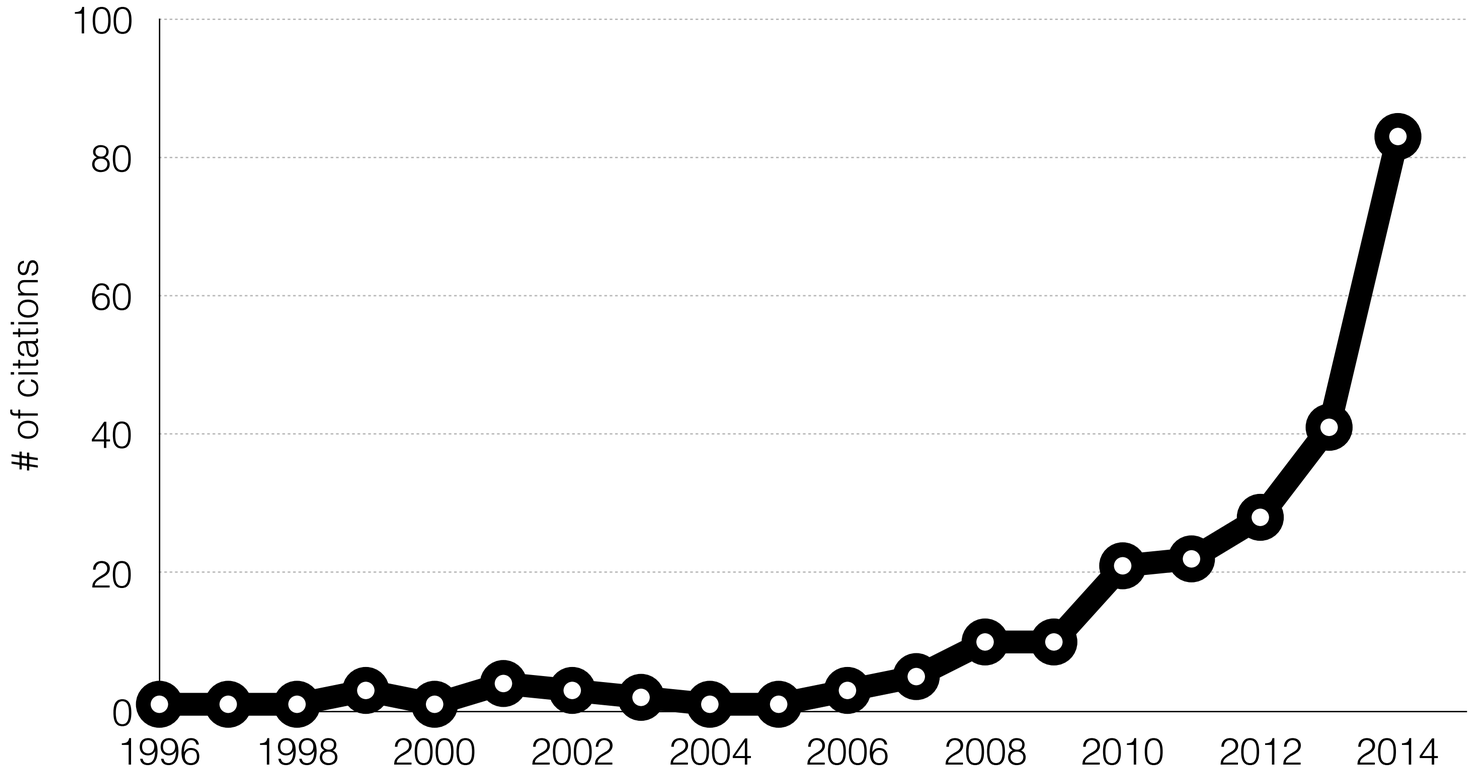


expose knobs

1 design/research skills do not scale



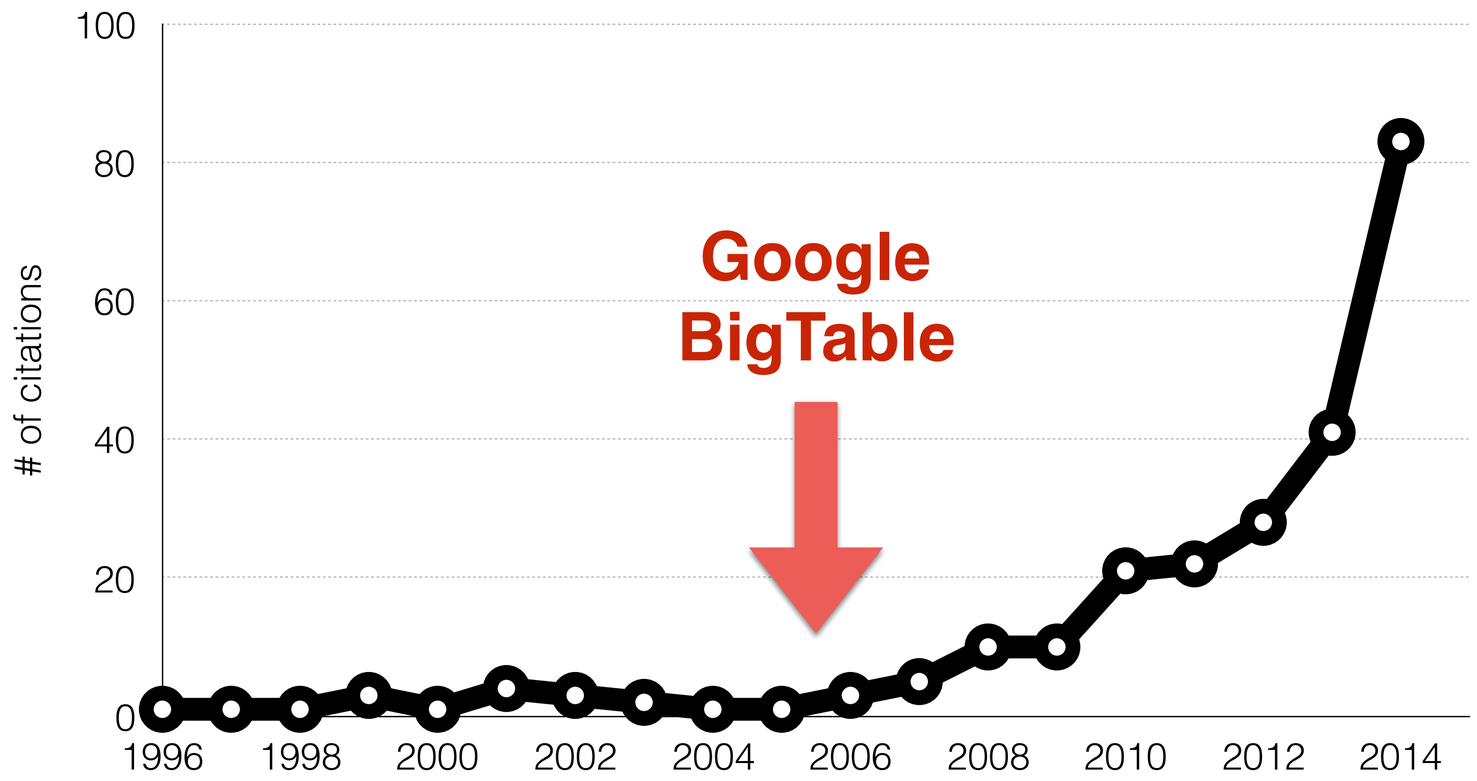
2 no one knows everything out there



NoSQL storage

P. O'Neil, E. Cheng, D. Gawlick, E, O'Neil
The log-structured merge-tree (LSM-tree)
Acta Informatica 33 (4): 351–385, 1996

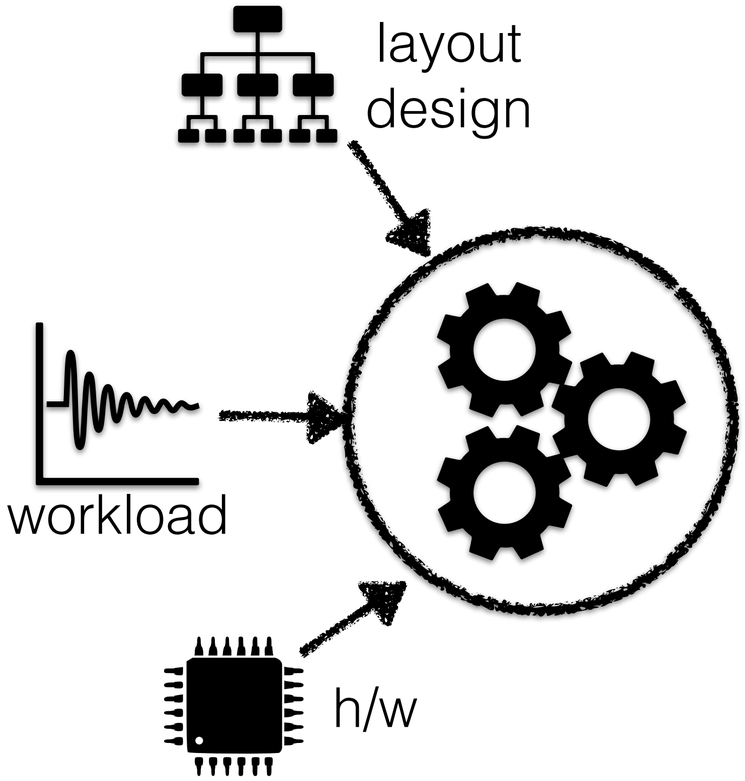
2 no one knows everything out there



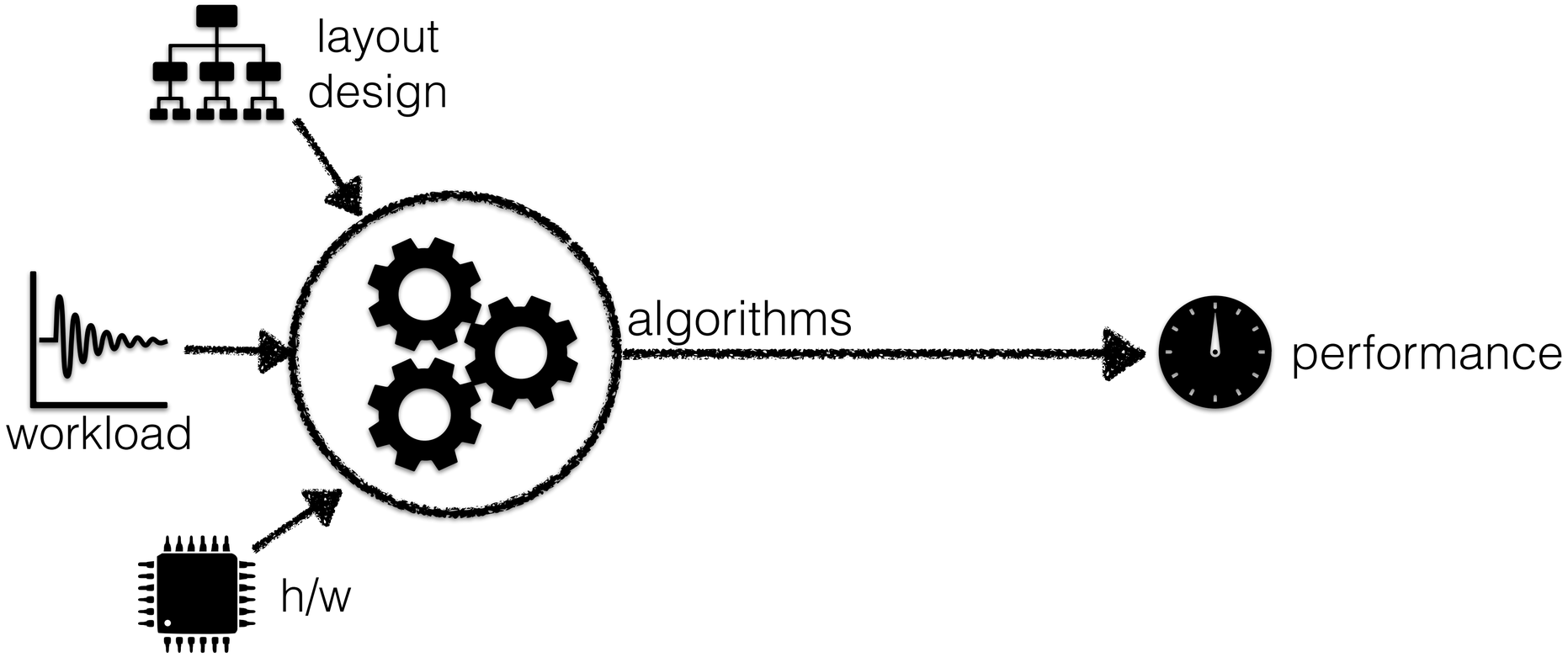
NoSQL storage

P. O'Neil, E. Cheng, D. Gawlick, E, O'Neil
The log-structured merge-tree (LSM-tree)
Acta Informatica 33 (4): 351–385, 1996

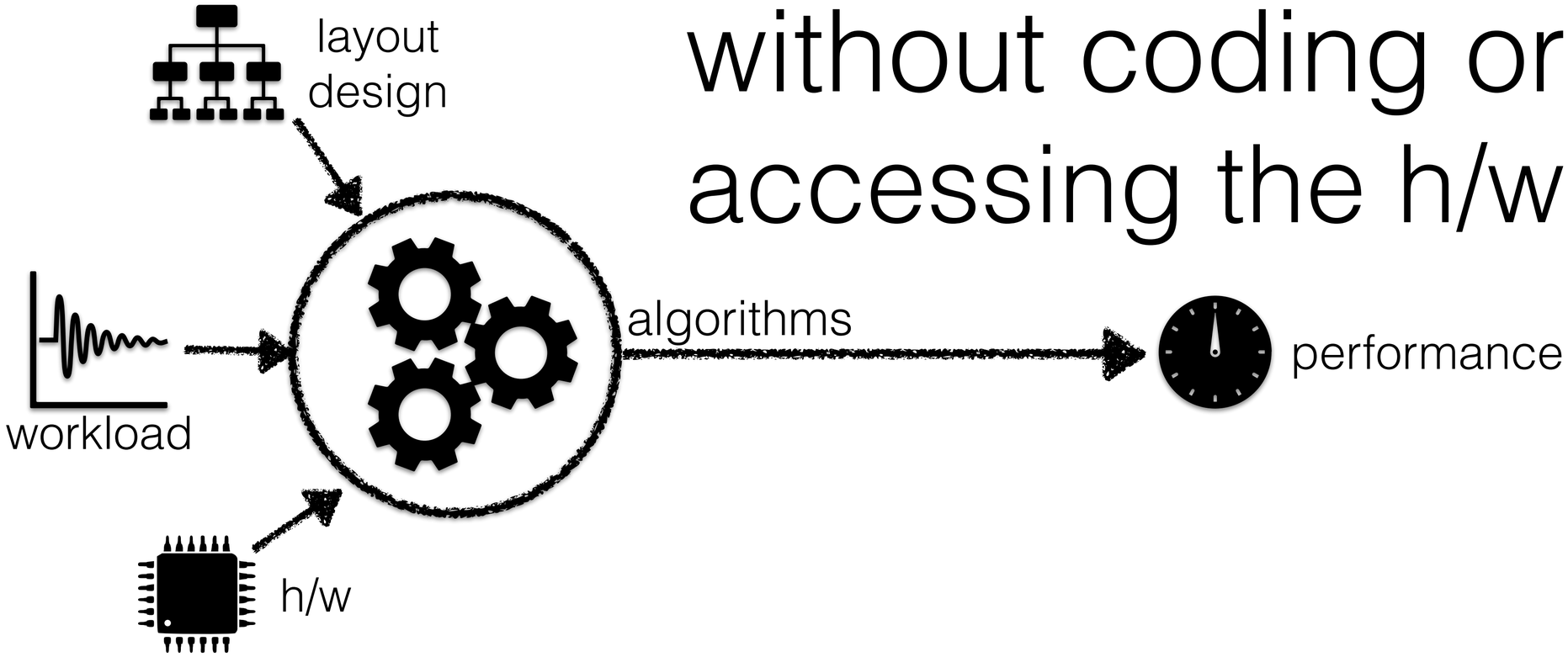
Data Calculator



Data Calculator

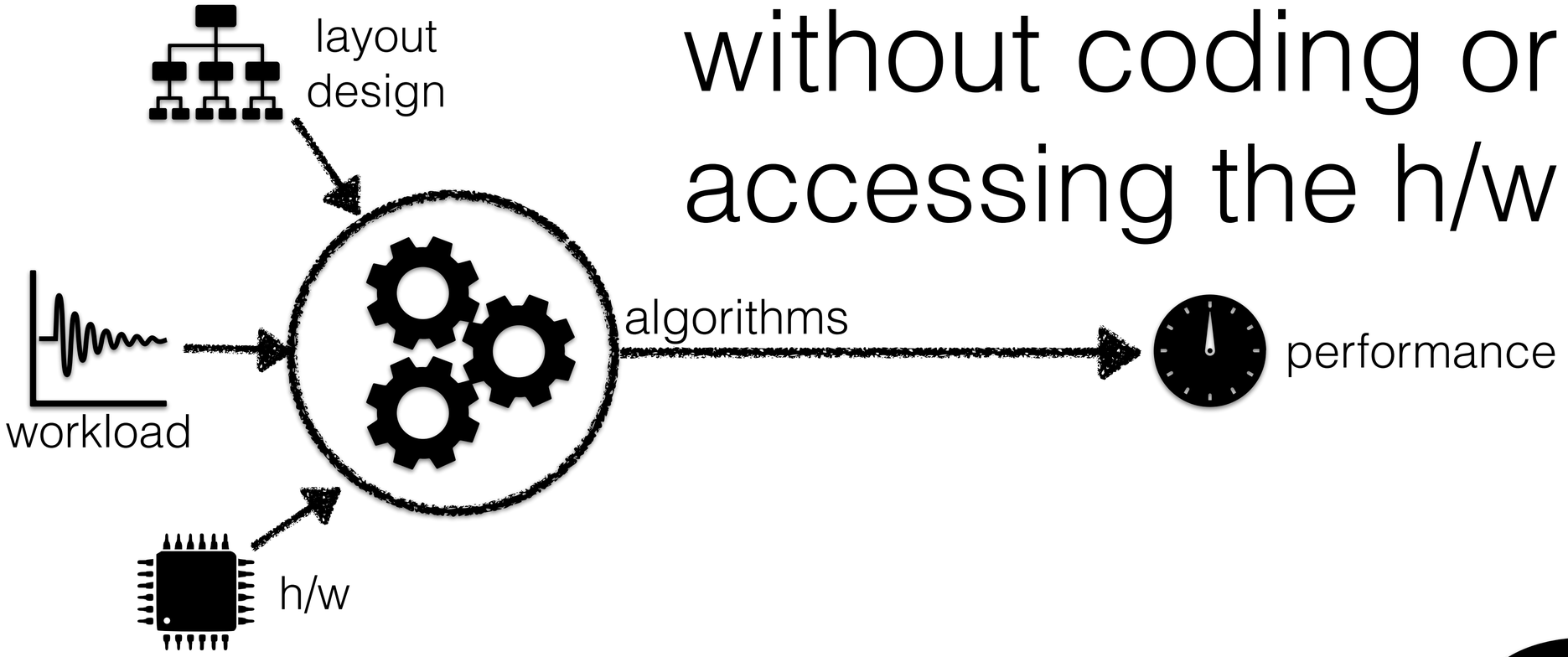


Data Calculator



without coding or accessing the h/w

Data Calculator



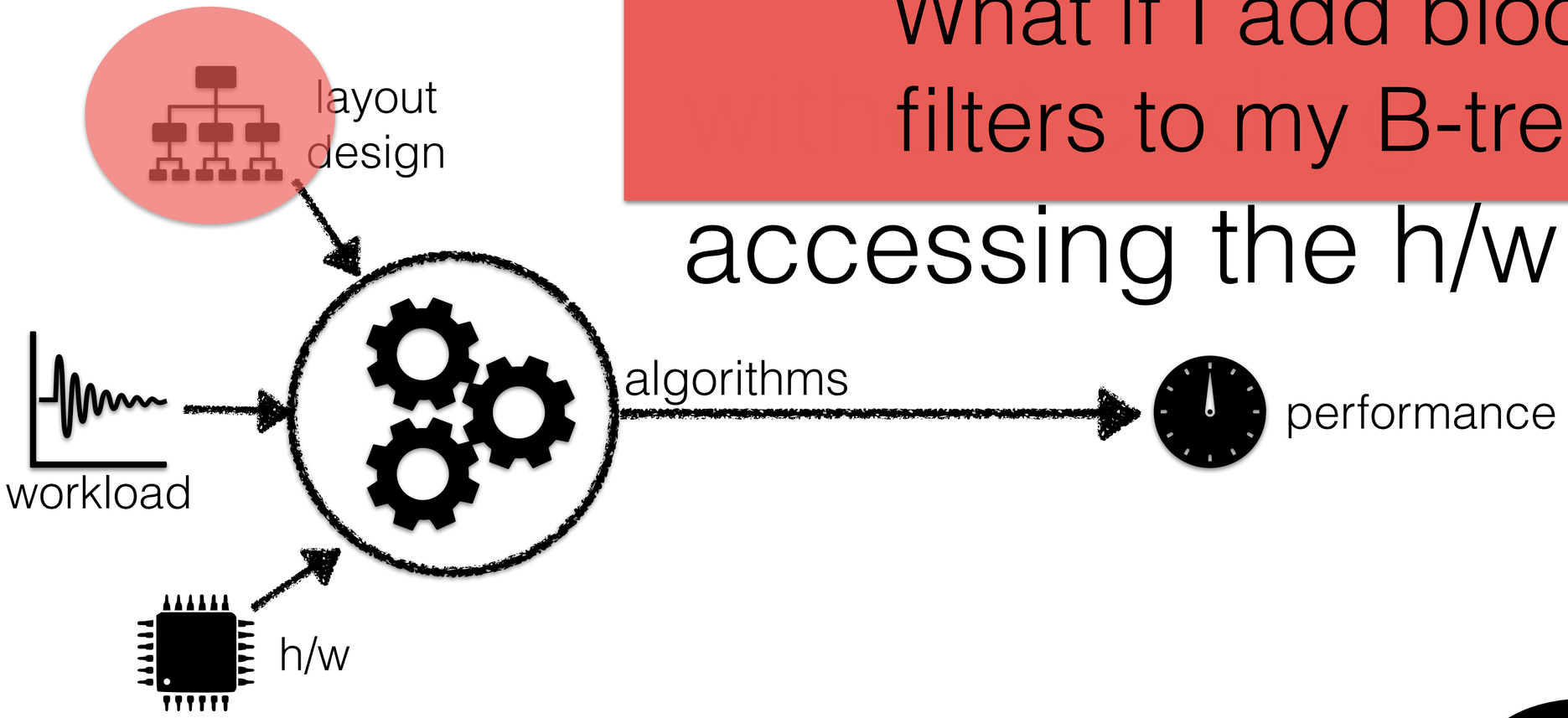
without coding or accessing the h/w

what-if design

Data Calculator

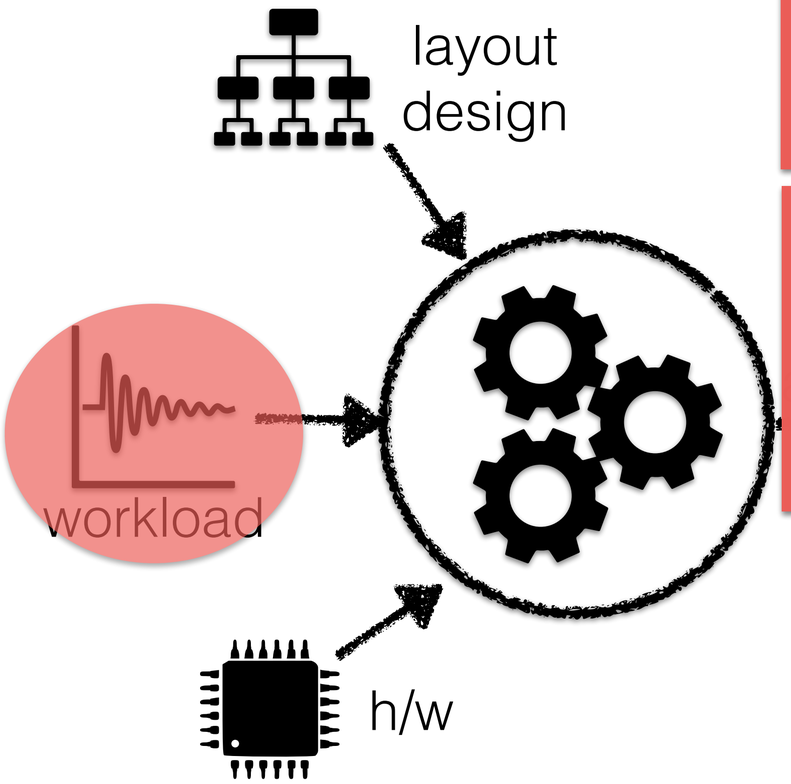
What if I add bloom filters to my B-tree?

accessing the h/w



what-if design

Data Calculator

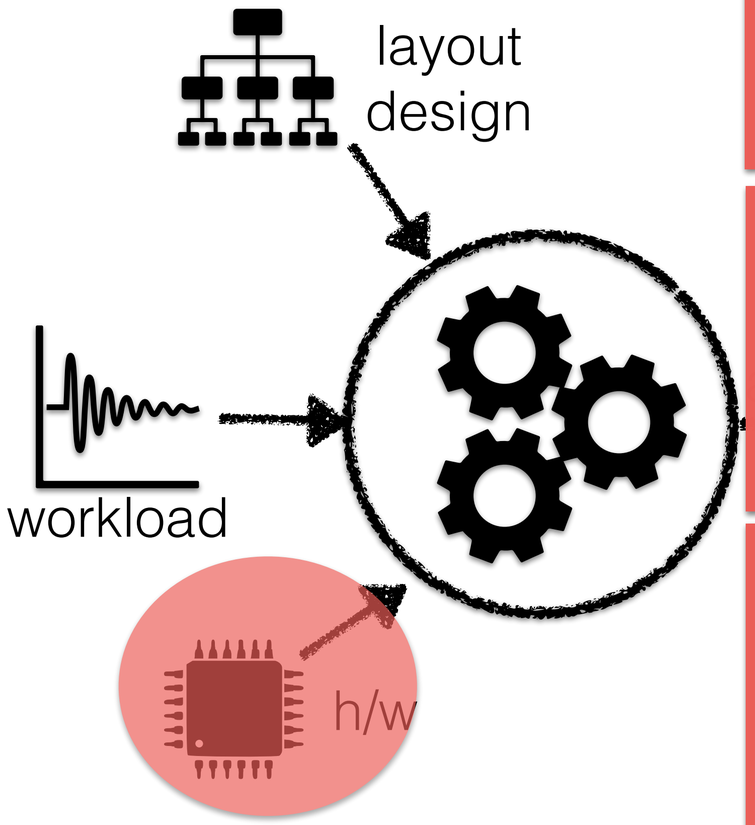


What if I add bloom filters to my B-tree?

What if add feature X that brings 60% more writes?

what-if design

Data Calculator



What if I add bloom filters to my B-tree?

What if add feature X that brings 60% more writes?

What if I need to reduce memory by 50%?

what-if design

Data Calculator

Cost in Amazon Cloud?

What if I add bloom filters to my B-tree?

Which workload breaks my system?

What if add feature X that brings 60% more writes?

Should I buy new hardware X?

What if I need to reduce memory by 50%?

what-if design

Rob Tarjan, Turing Award 1986

"IS THERE A CALCULUS OF DATA STRUCTURES

by which one can choose the appropriate representation and techniques for a given problem?" (SIAM,1978)

[*P* vs *NP, average case, constant factors vs asymptotic, low bounds*]

How many and which structures are possible?

DESIGN SPACE

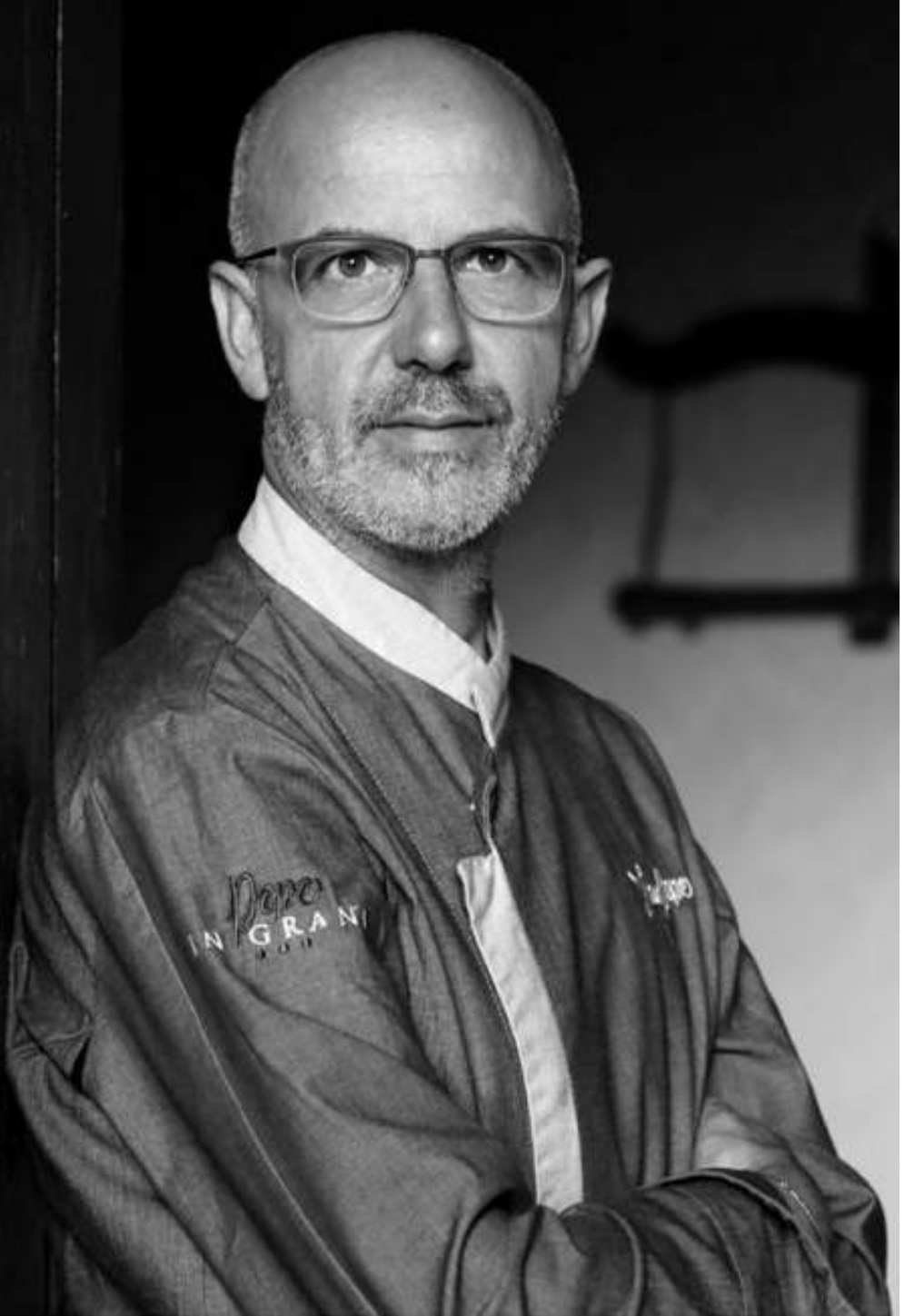
How many and which structures are possible?

DESIGN SPACE

Can we compute performance w/o coding?

COST SYNTHESIS

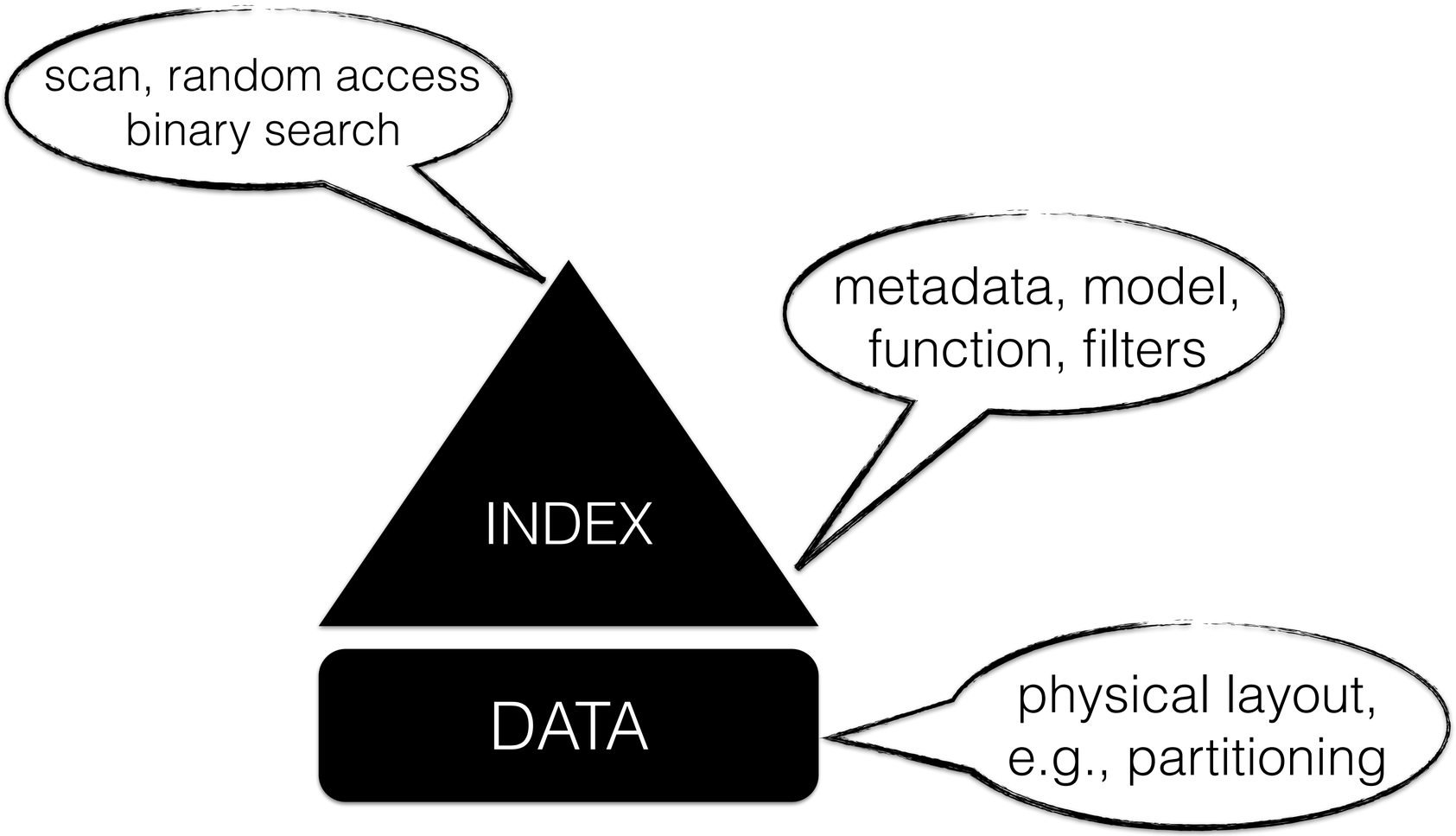

researcher
in action



Franco Pepe

“there is no pizza recipe
he who thinks there is a pizza recipe is no pizza maker”

EVERY DESIGN: **1** A **SET** OF CONCEPTS **2** **EXISTING** OR **NEW** CONCEPTS **3** ALL GOOD IDEAS IN THE 60s?

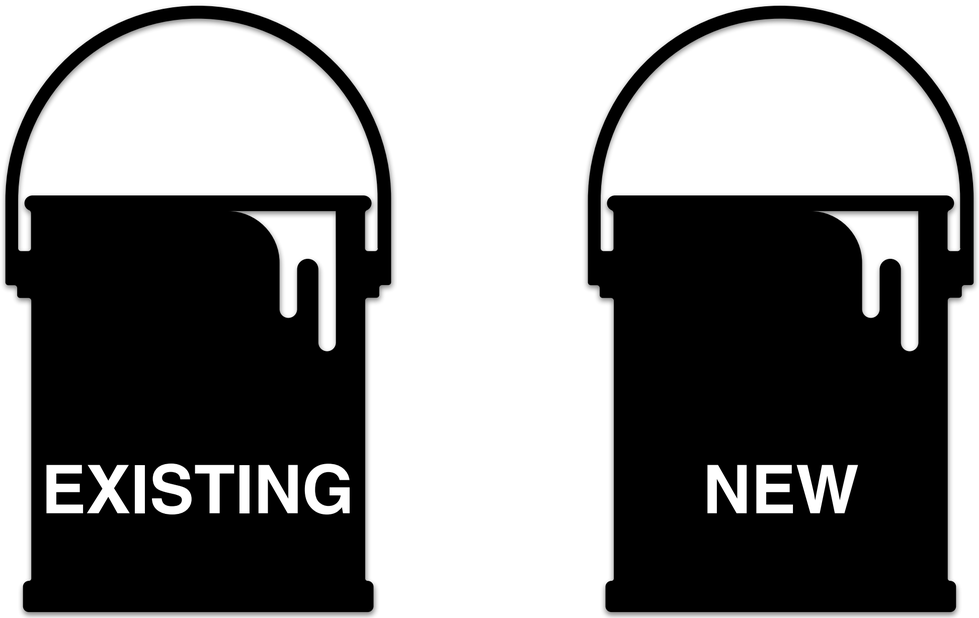


EVERY DESIGN:

1 A **SET** OF CONCEPTS

2 **EXISTING** OR **NEW** CONCEPTS

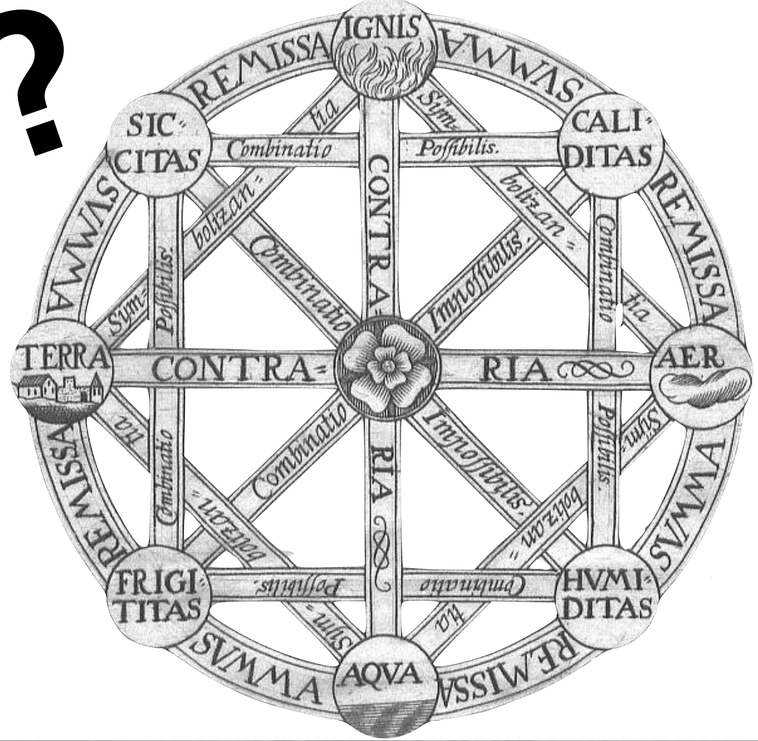
3 ALL GOOD IDEAS IN THE 60s?

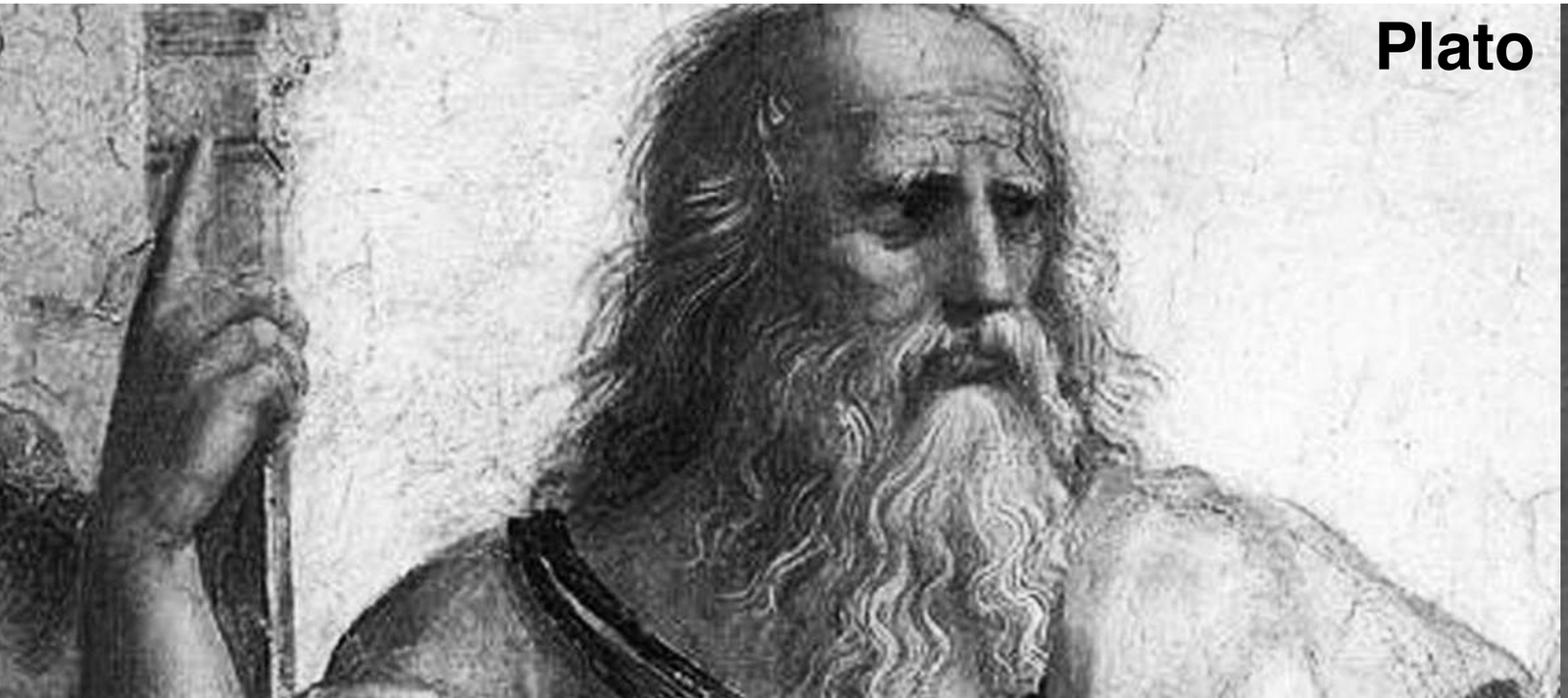


EVERY DESIGN: **1** A **SET** OF CONCEPTS **2** **EXISTING** OR **NEW** CONCEPTS **3** ALL GOOD IDEAS IN THE 60s?

(ALMOST) ALL DESIGNS ARE A COMBINATION/ TUNING OF **EXISTING CONCEPTS**

what is creativity?





Nikos Kazantzakis, philosopher

action is for nothing
hope the most holy of
fear form
am free
ultimate I theory

Nikos Kazantzakis, philosopher

action is the most holy form of ultimate theory

I hope for nothing
I fear nothing
I am free

Nikos Kazantzakis, philosopher

action is the most holy form of ultimate theory

NEW

I hope for nothing
I fear nothing
I am free

Nikos Kazantzakis, philosopher

action is the most holy form of ultimate theory

NEW & BRILLIANT

I hope for nothing

I fear nothing

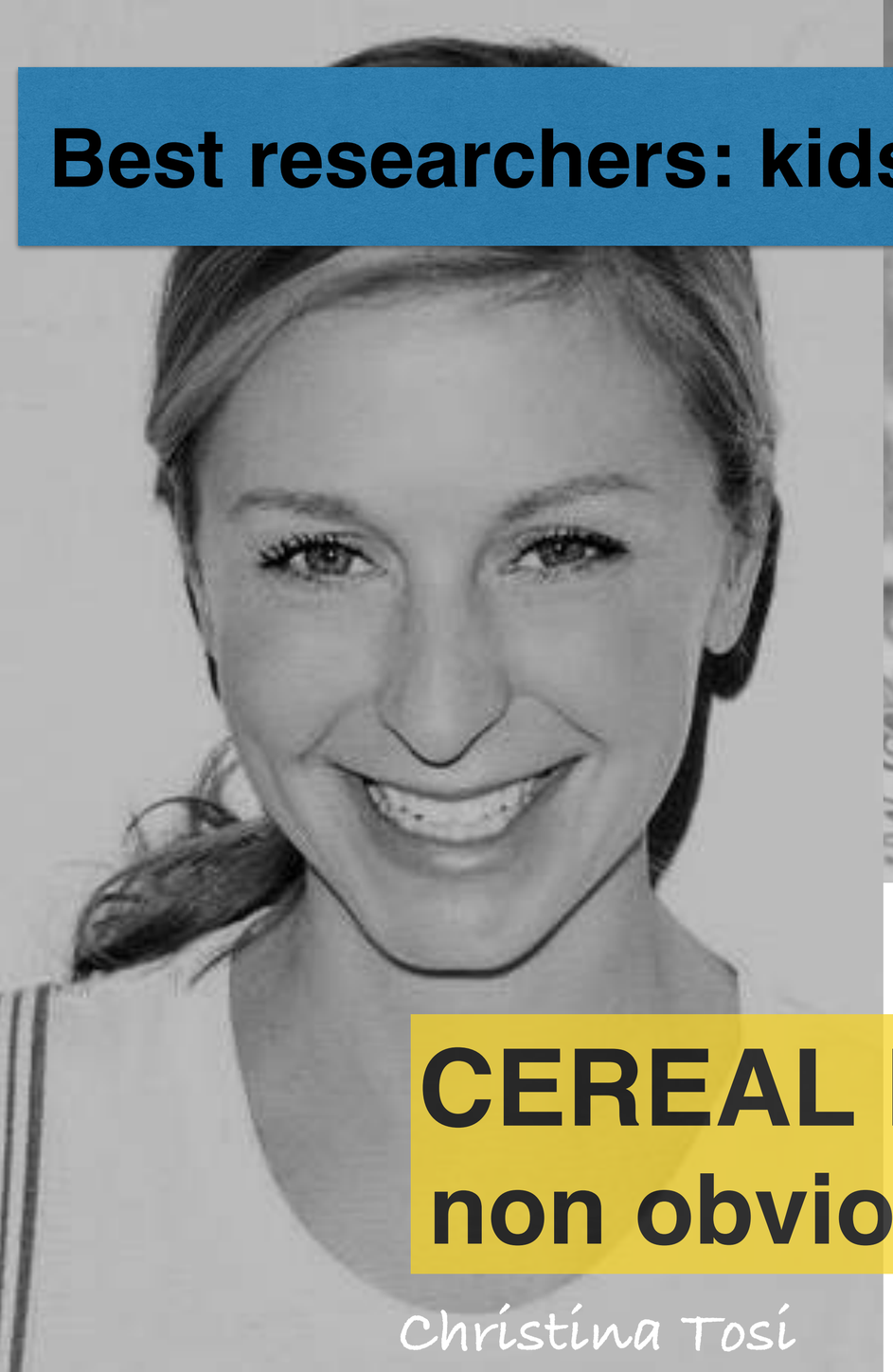
I am free

milk + cream + sugar + vanilla/lemon

CEREAL MILK PANNA COTTA
non obvious valid combinations

Christina Tosi

Best researchers: kids, young students, adults that stay kids



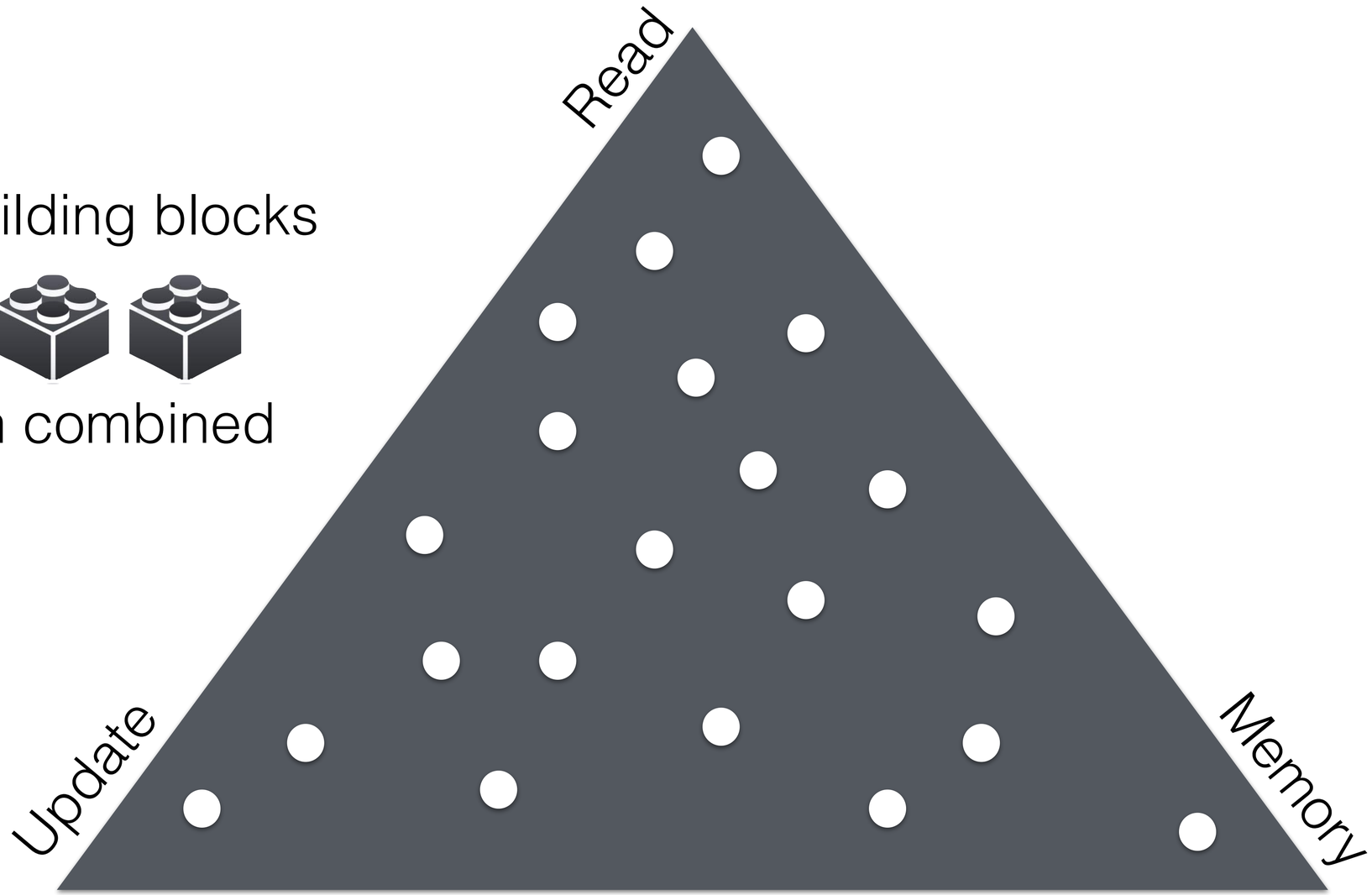
milk + cream + sugar + vanilla/lemon

CEREAL MILK PANNA COTTA
non obvious valid combinations

Christina Tosi

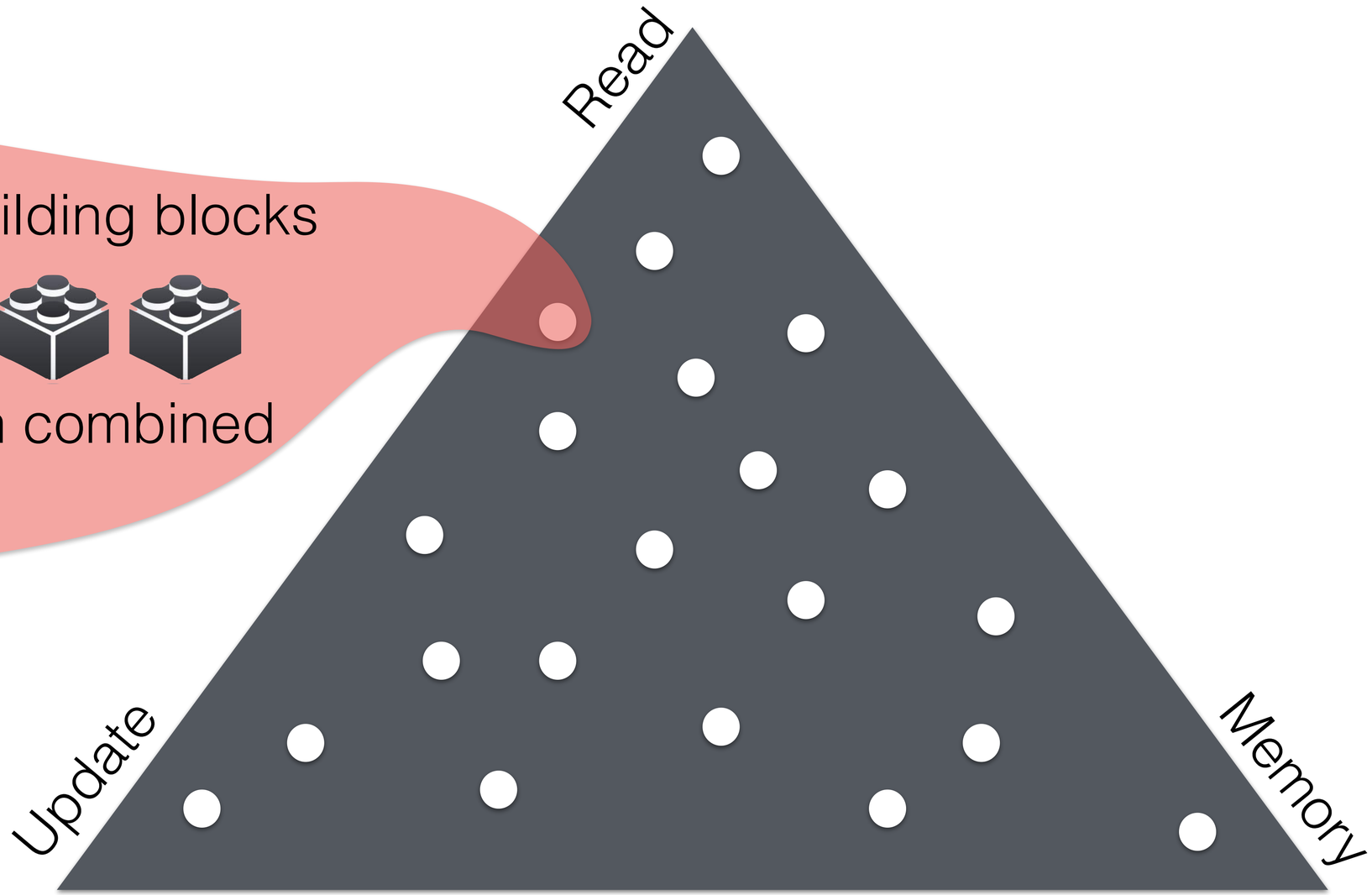
fundamental building blocks

properties when combined



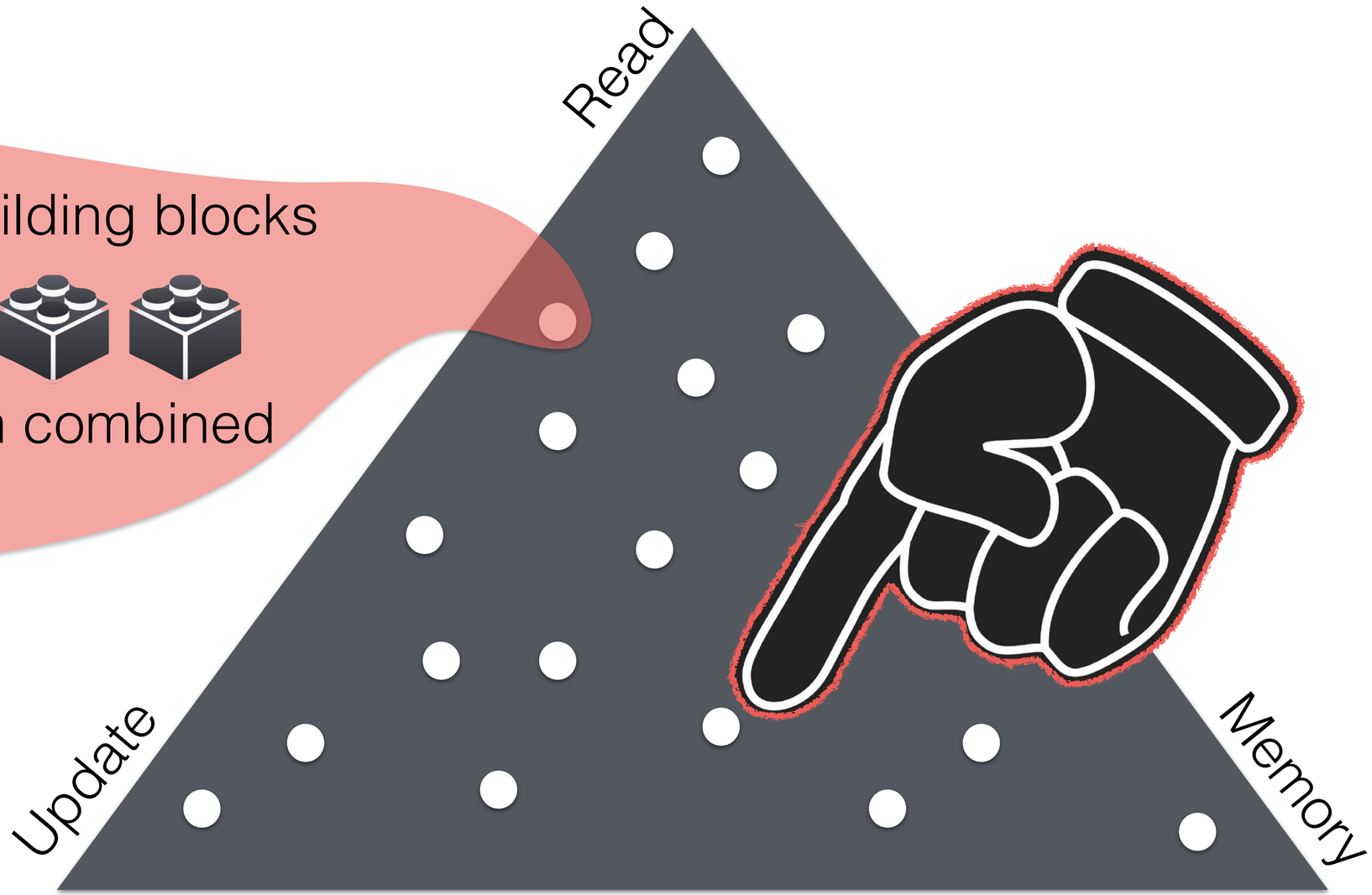
fundamental building blocks

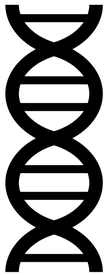
properties when combined



fundamental building blocks

properties when combined

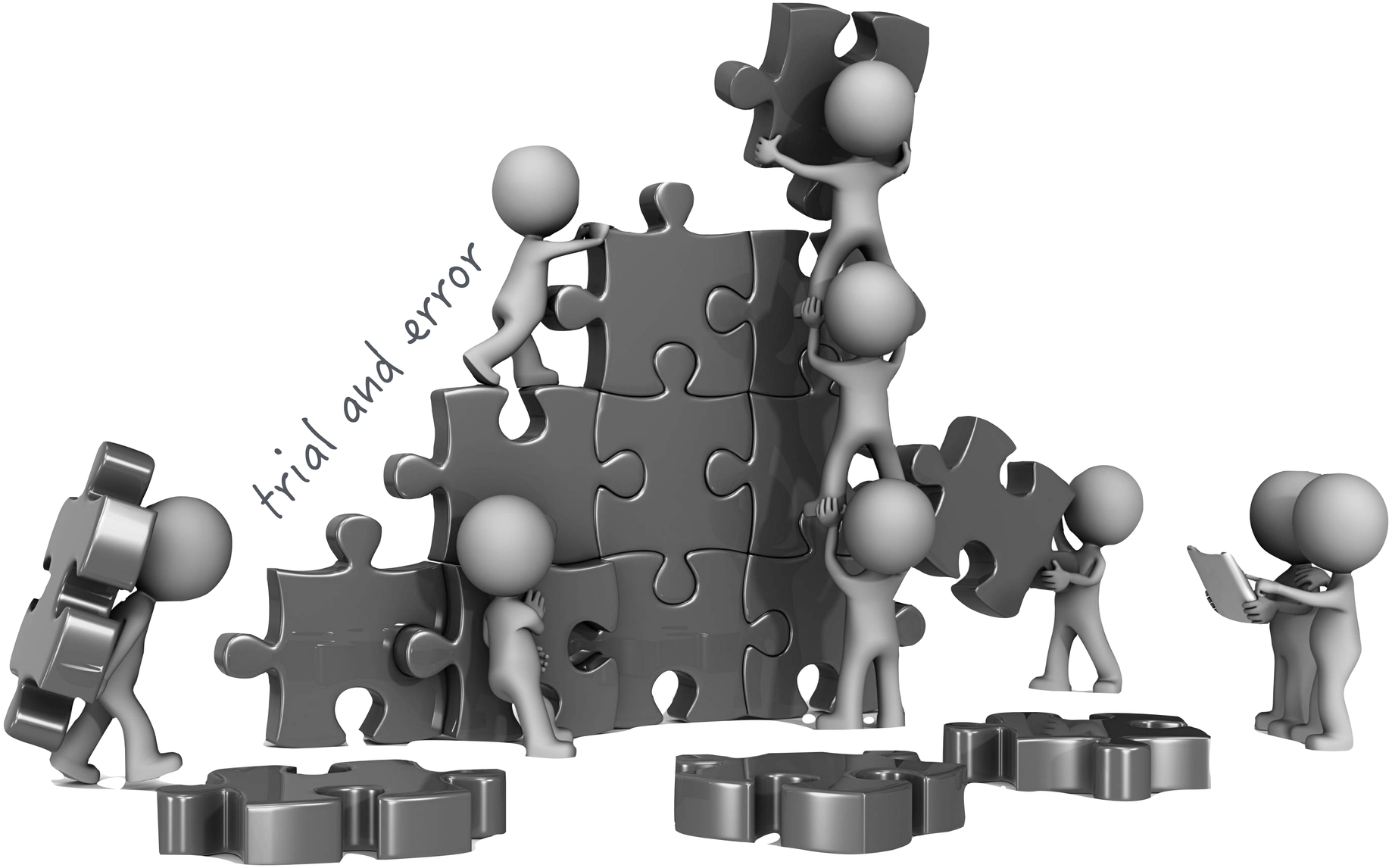




DESIGN SPACE

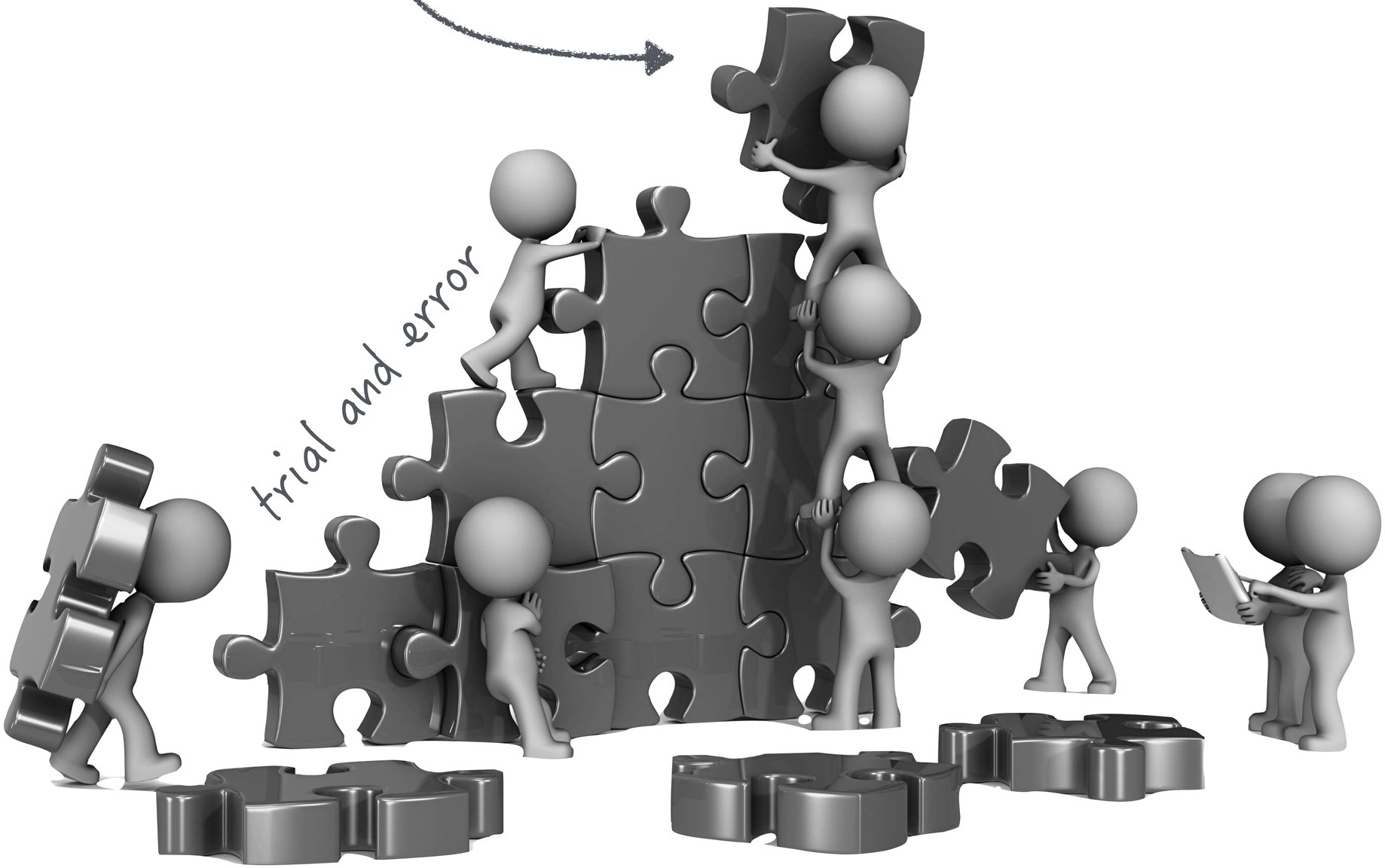
COST SYNTHESIS

WHAT-IF



DASlab
@ Harvard SEAS

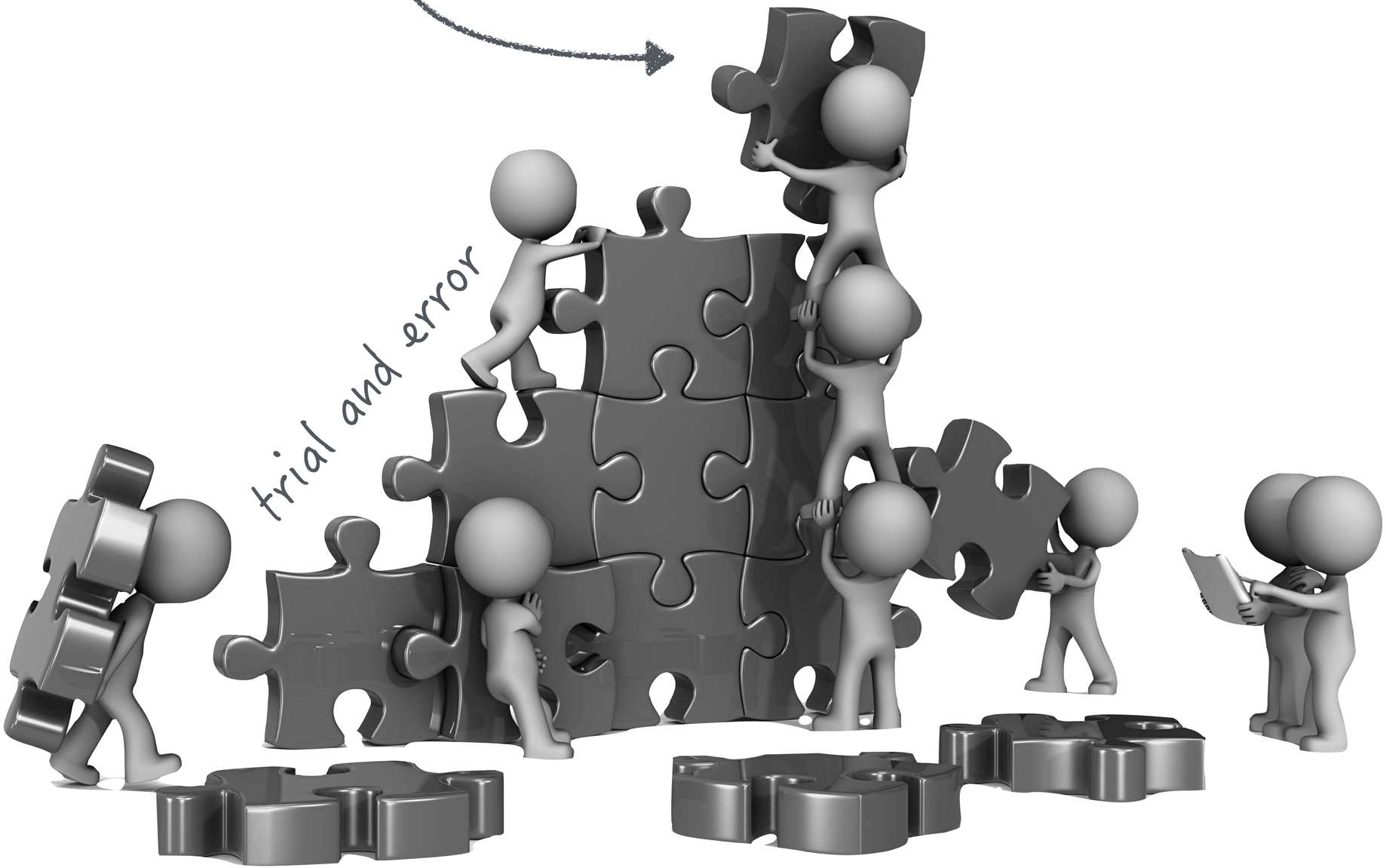
FIRST PRINCIPLE: design concept that does not break further



DASlab
@ Harvard SEAS

FIRST PRINCIPLE: design concept that does not break further

MAP LAYOUT FIRST

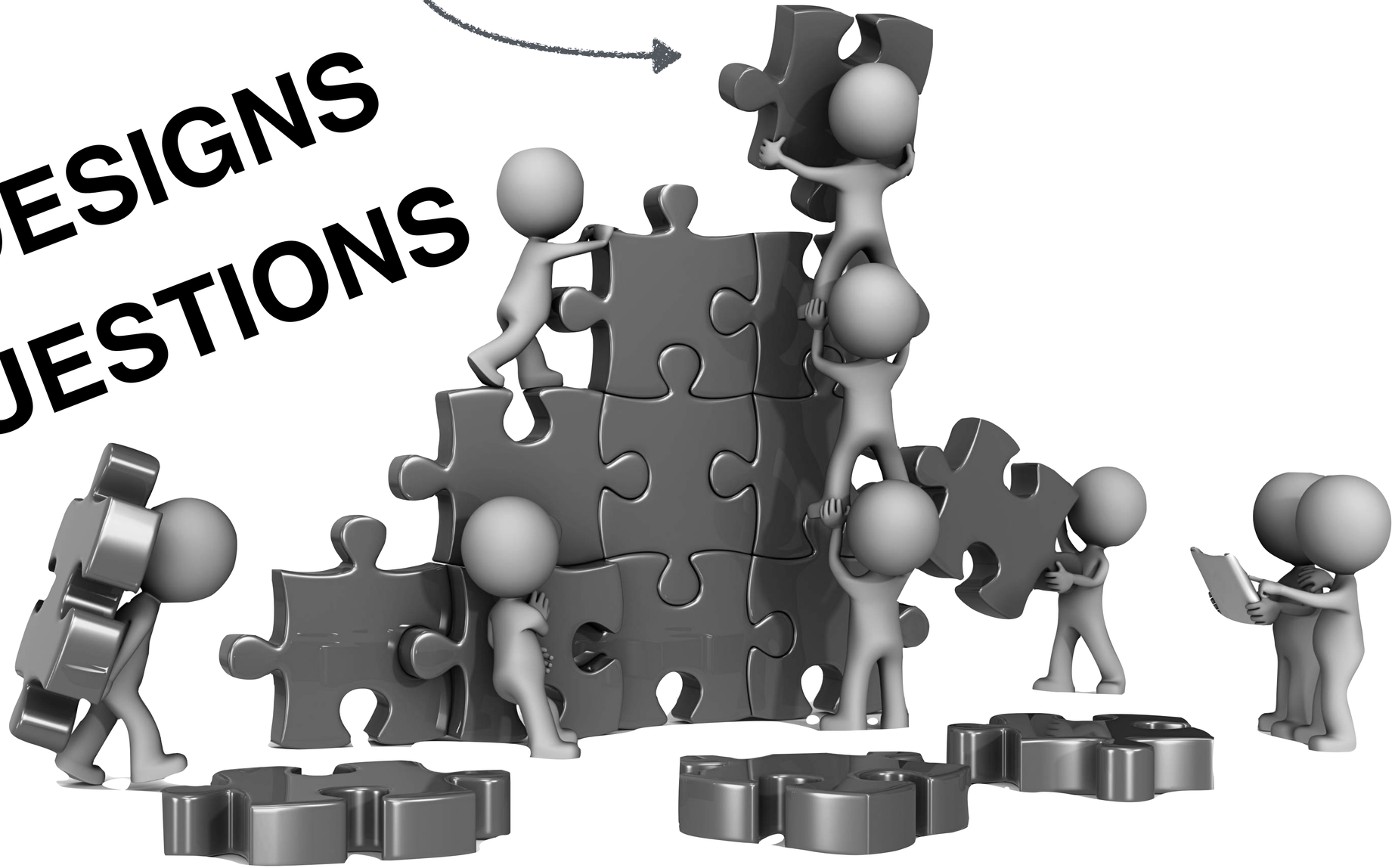


DASlab
@ Harvard SEAS

FIRST PRINCIPLE: design concept that does not break further

MAP LAYOUT FIRST

KNOWN DESIGNS
OPEN QUESTIONS



DASlab
@ Harvard SEAS

NoSQL Key-value Stores

RocksDB Google BigTable mongoDB SQLite in

amazon DynamoDB cassandra APACHE HBASE

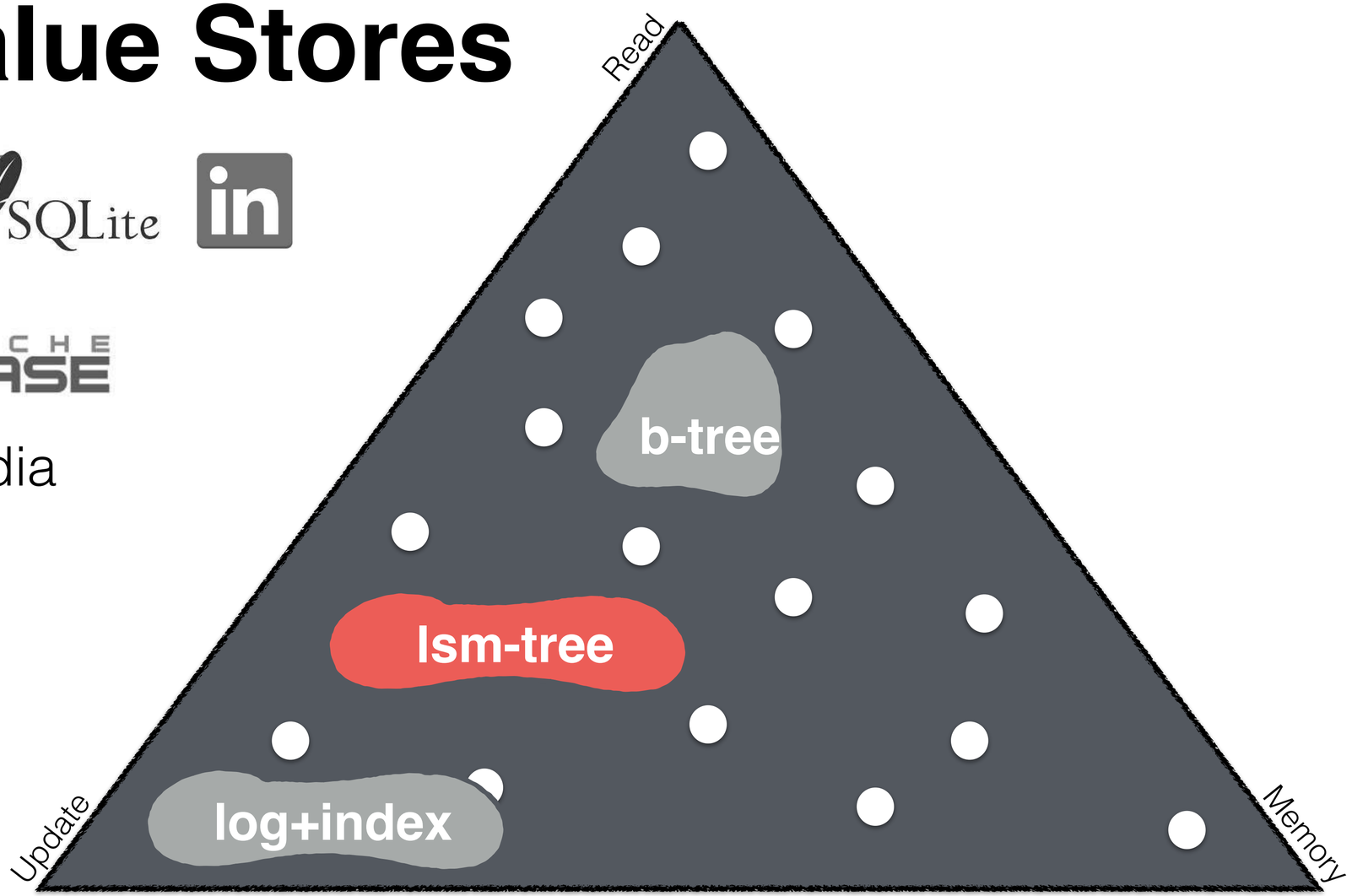
machine learning social media

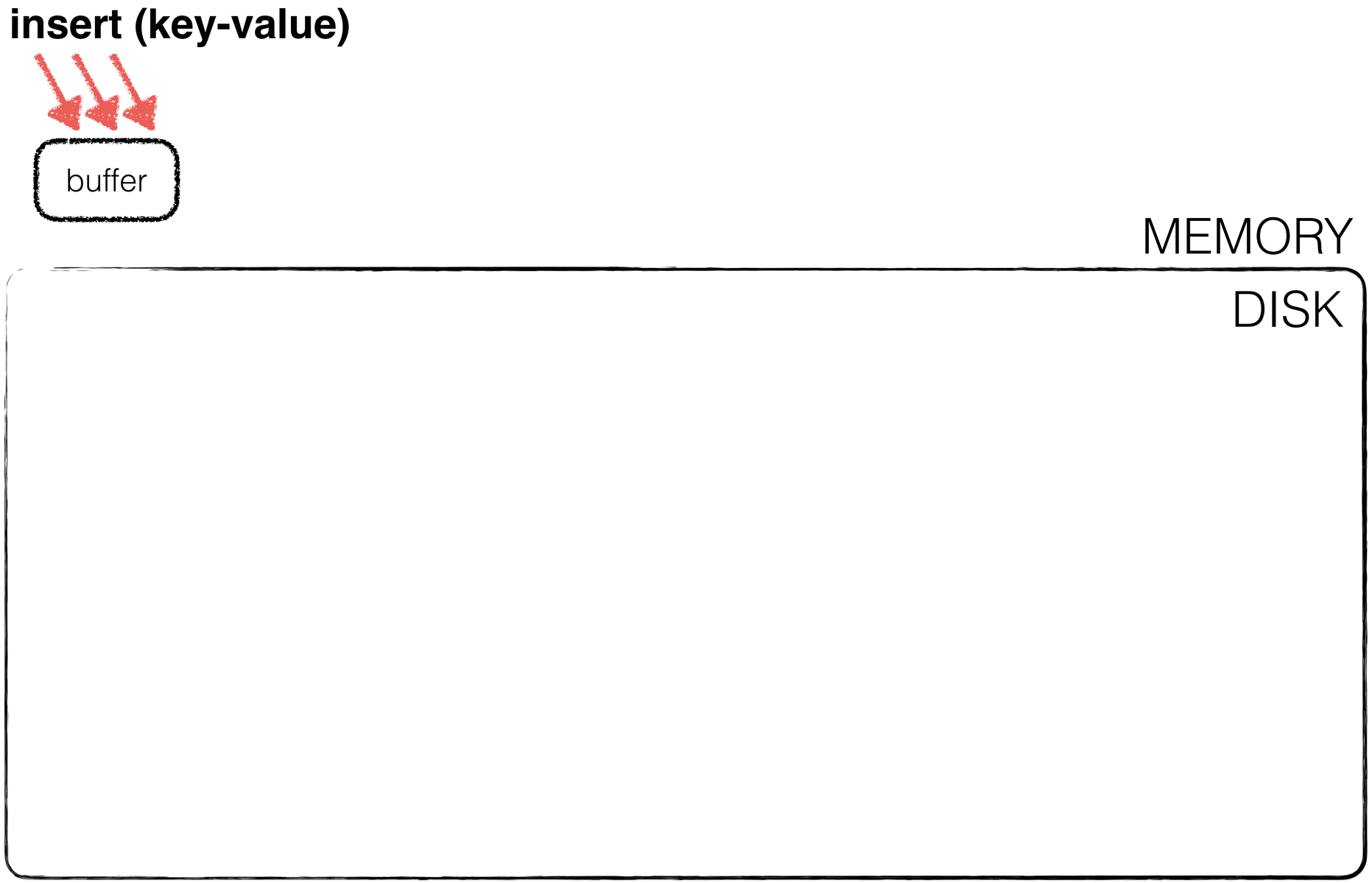
smart homes web browsers

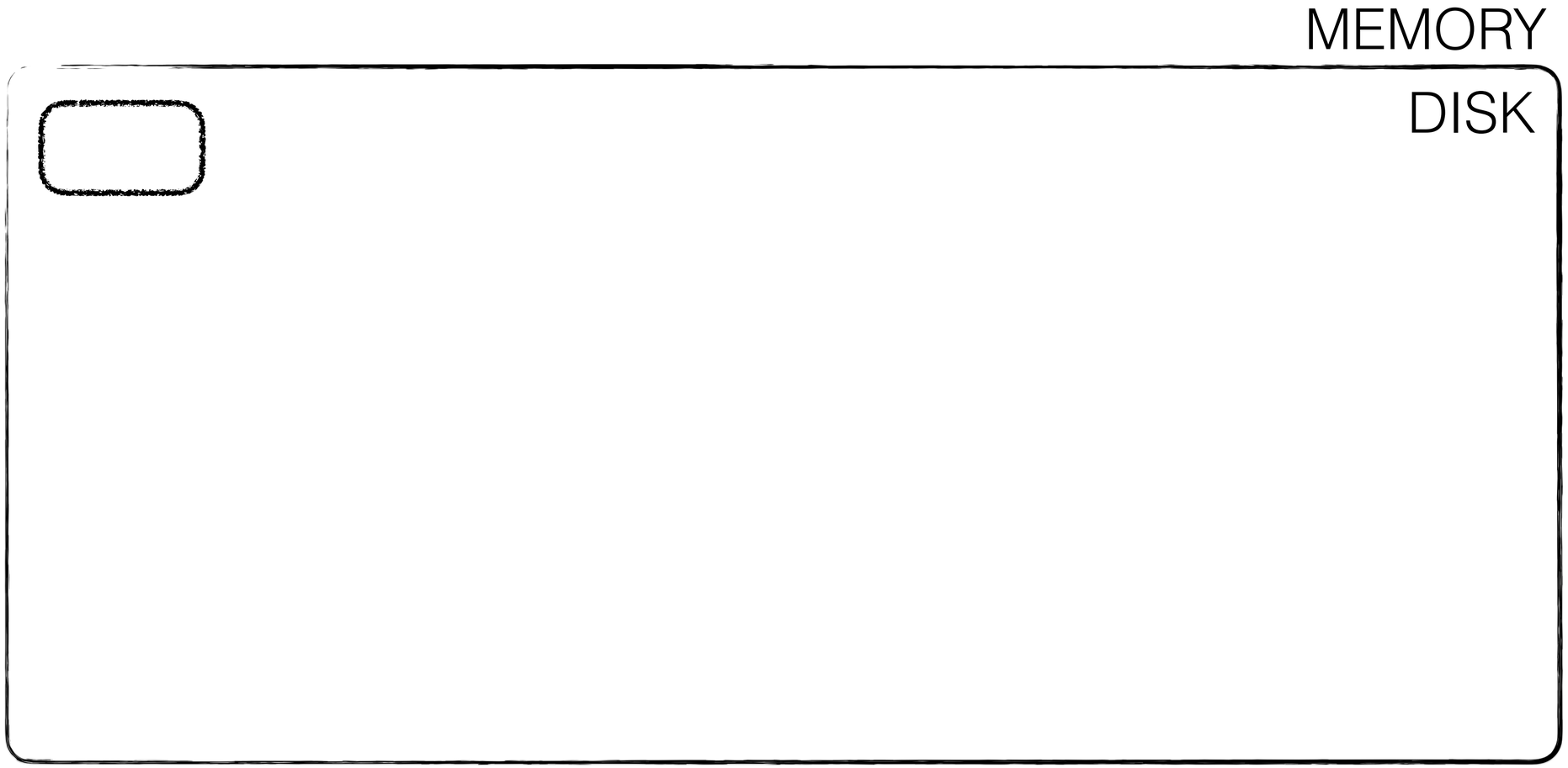
phones web-based apps

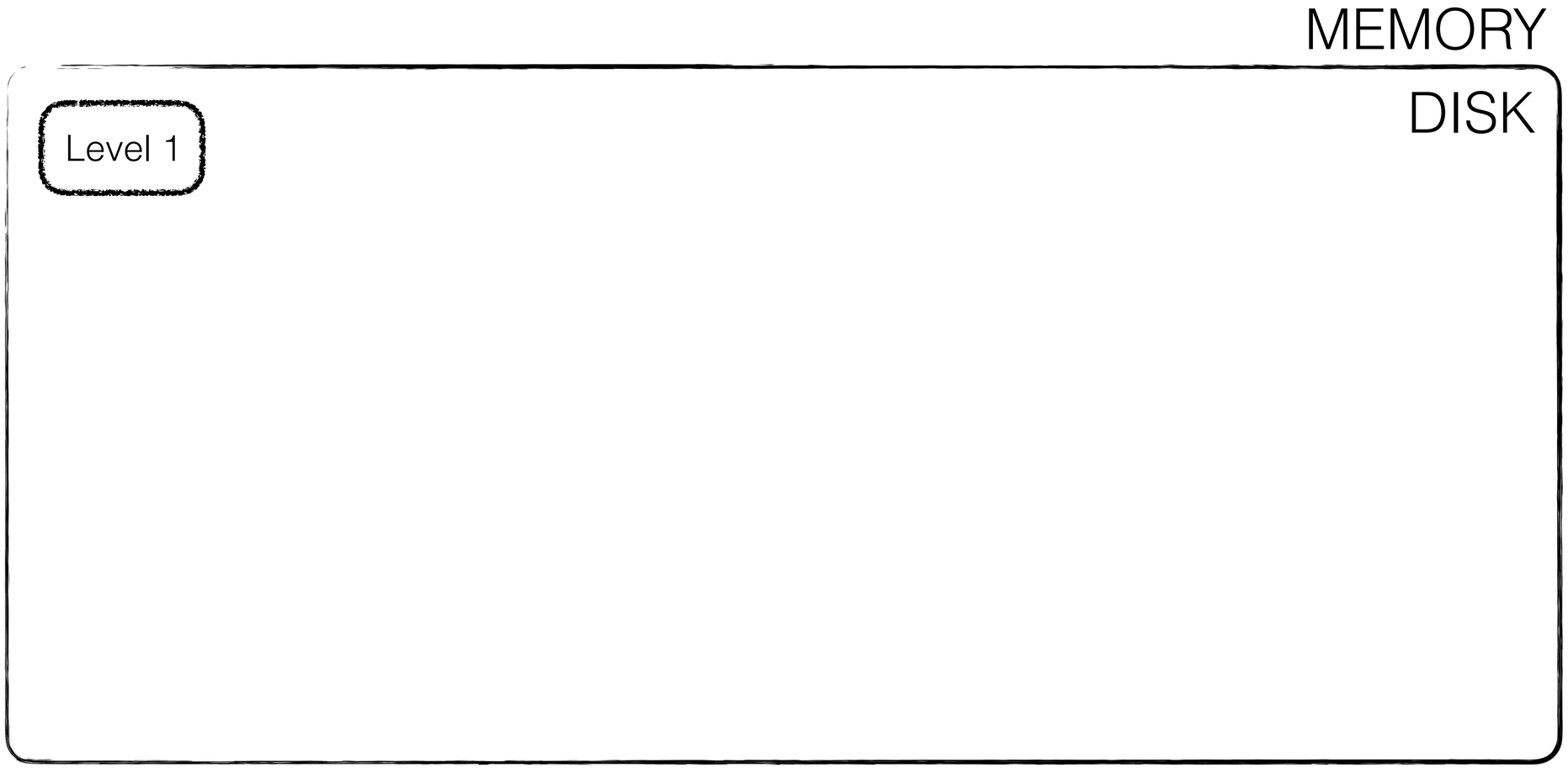
security health devices

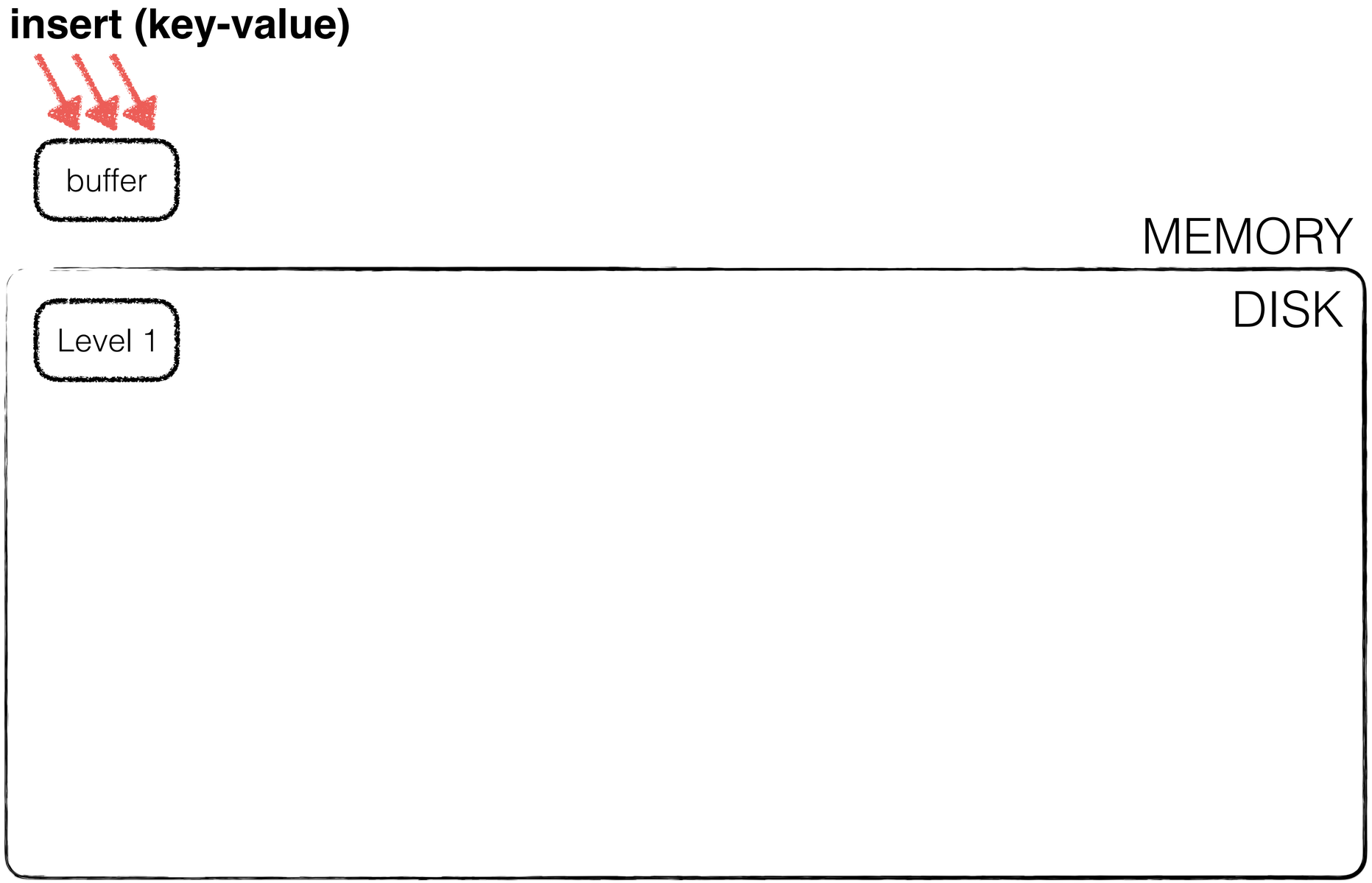
graphs analytics

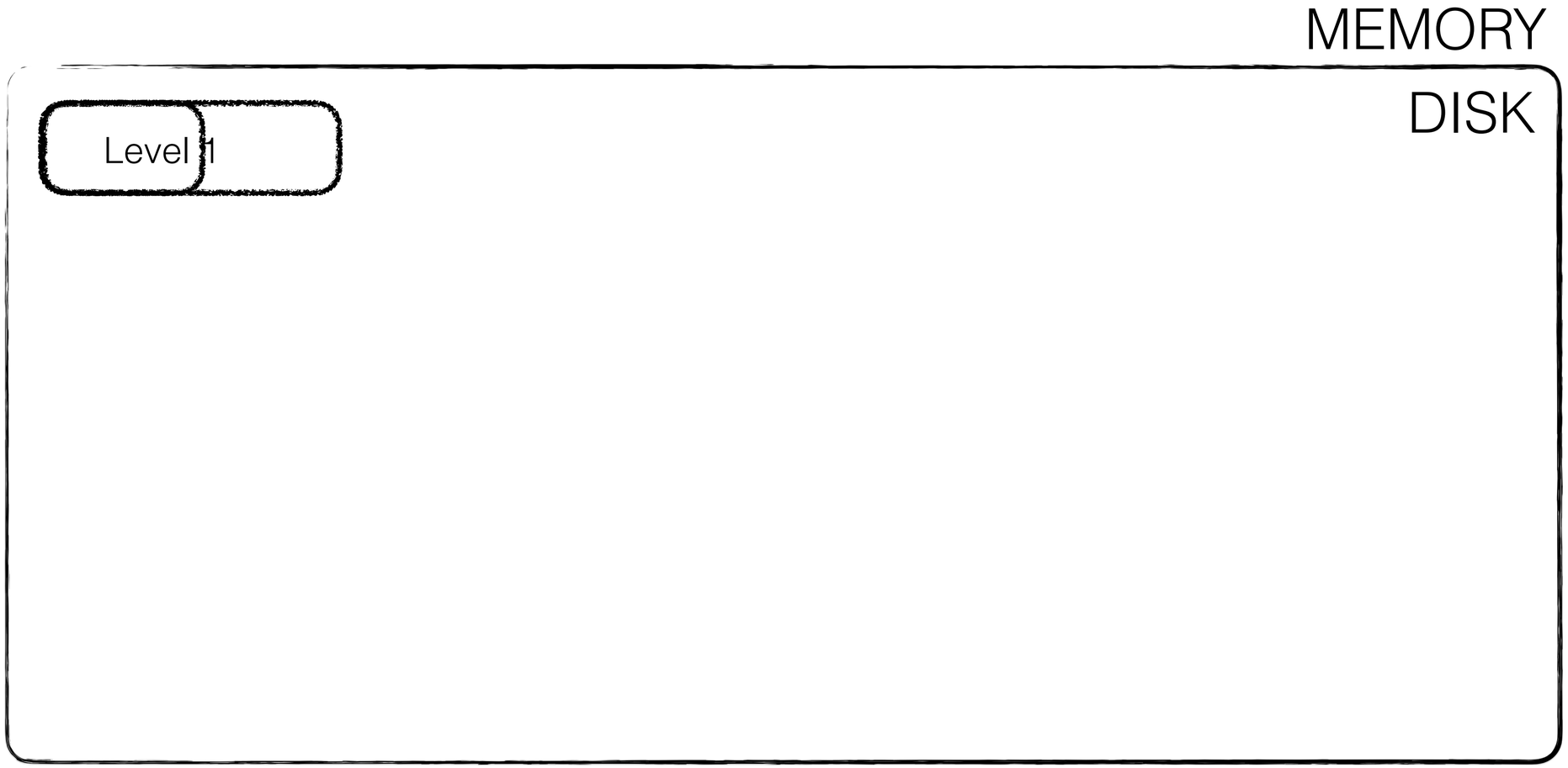


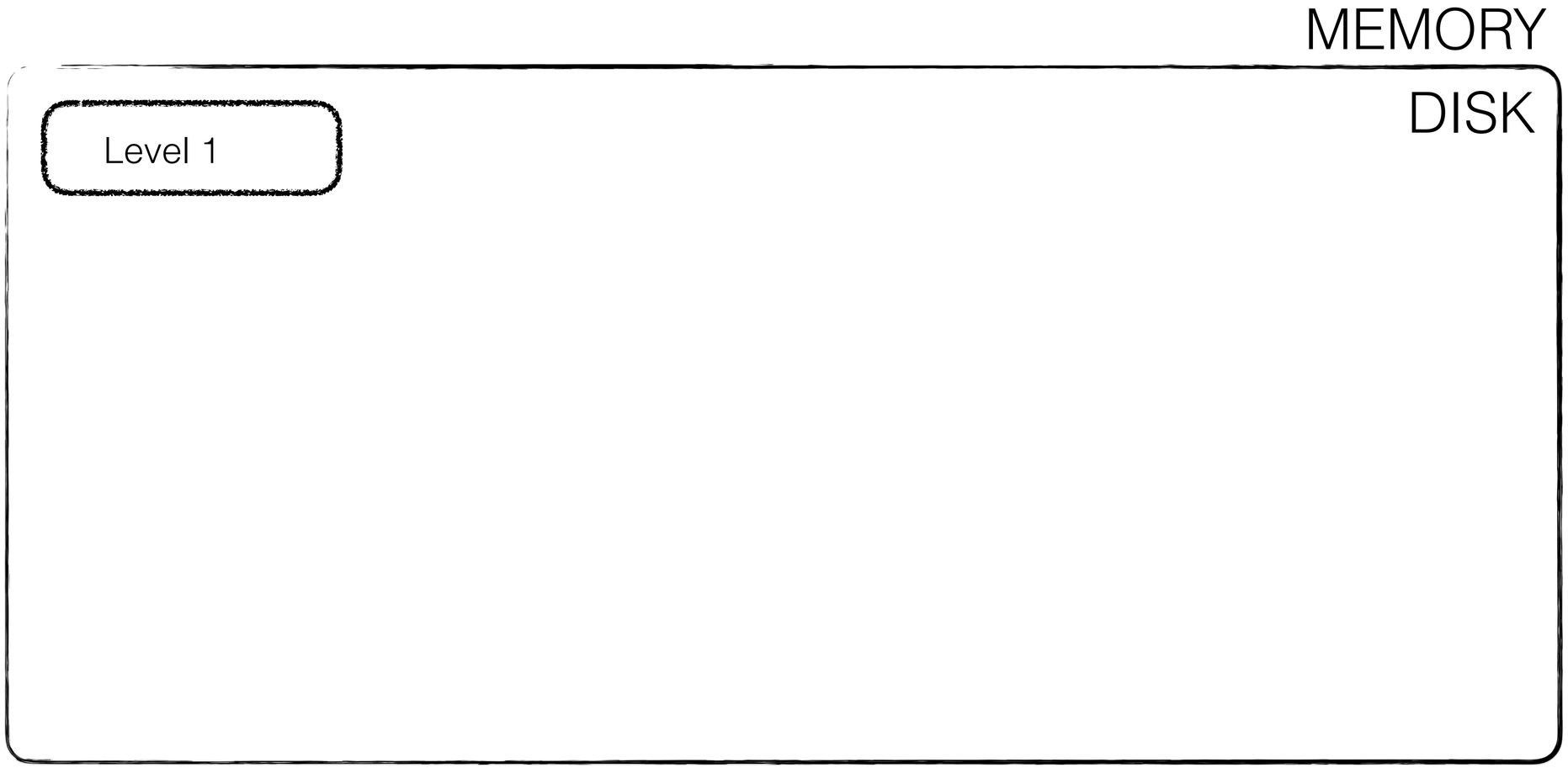

insert (key-value)
buffer
MEMORY
DISK

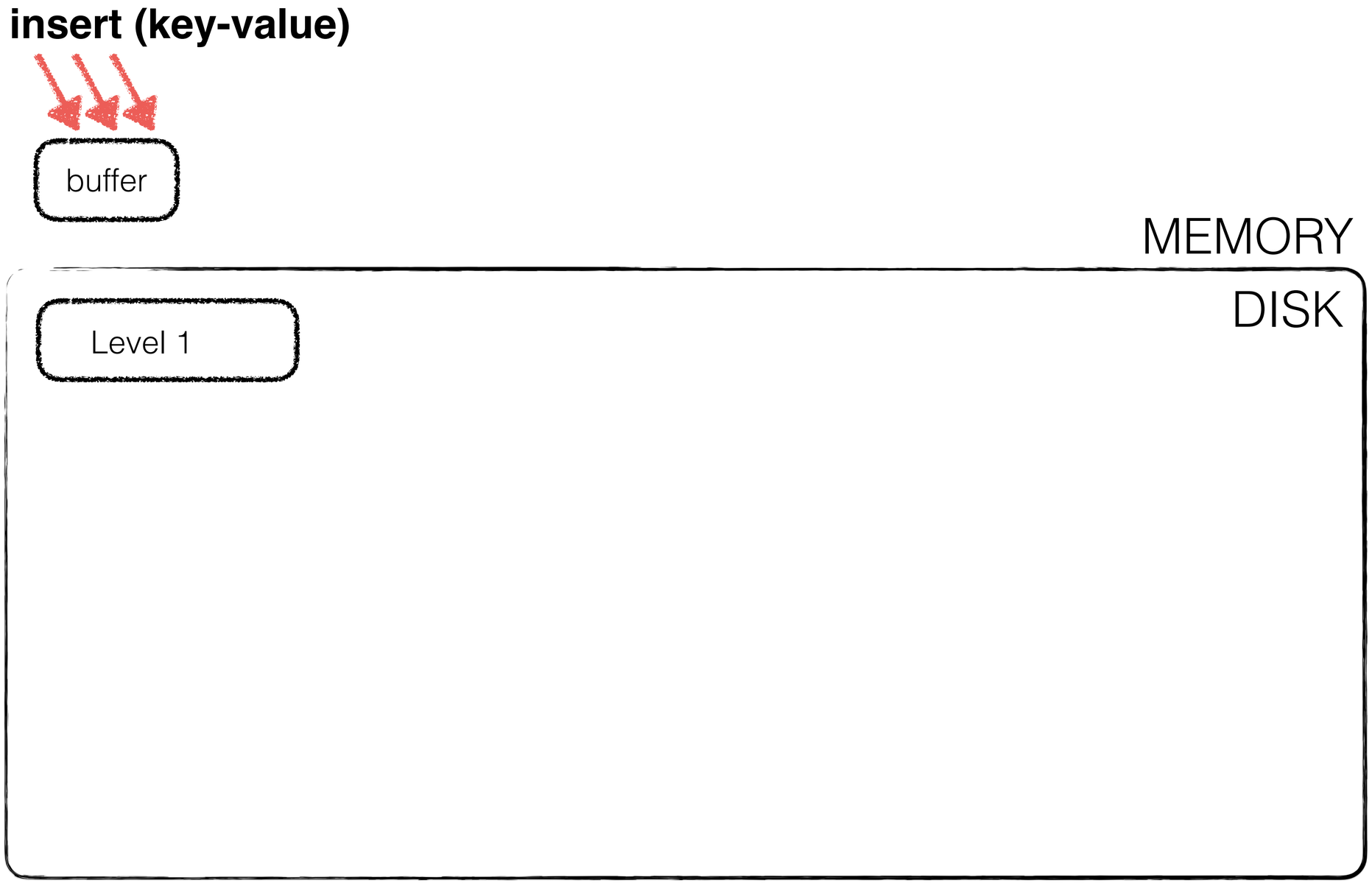

MEMORY
DISK

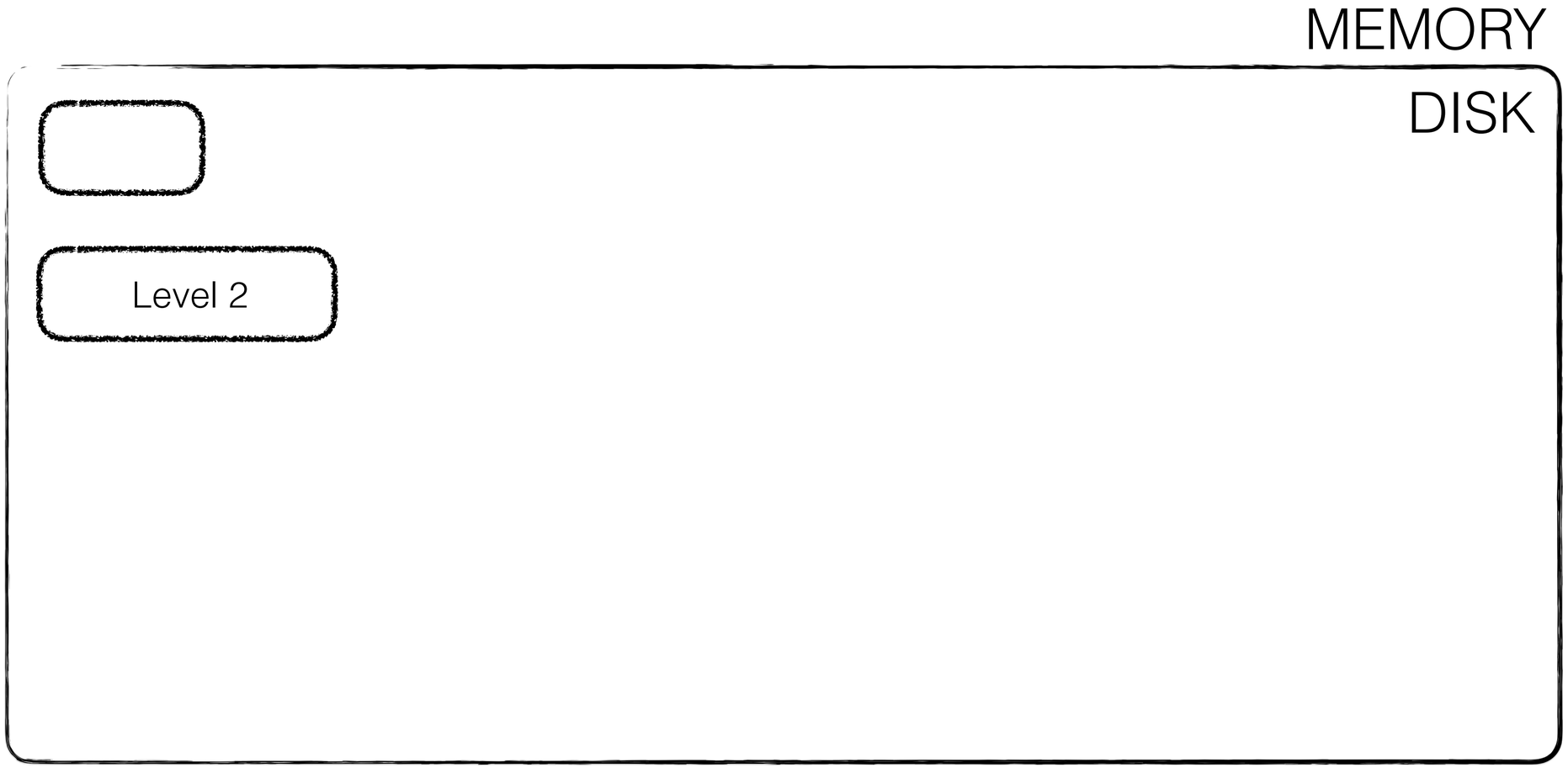

MEMORY
DISK
Level 1

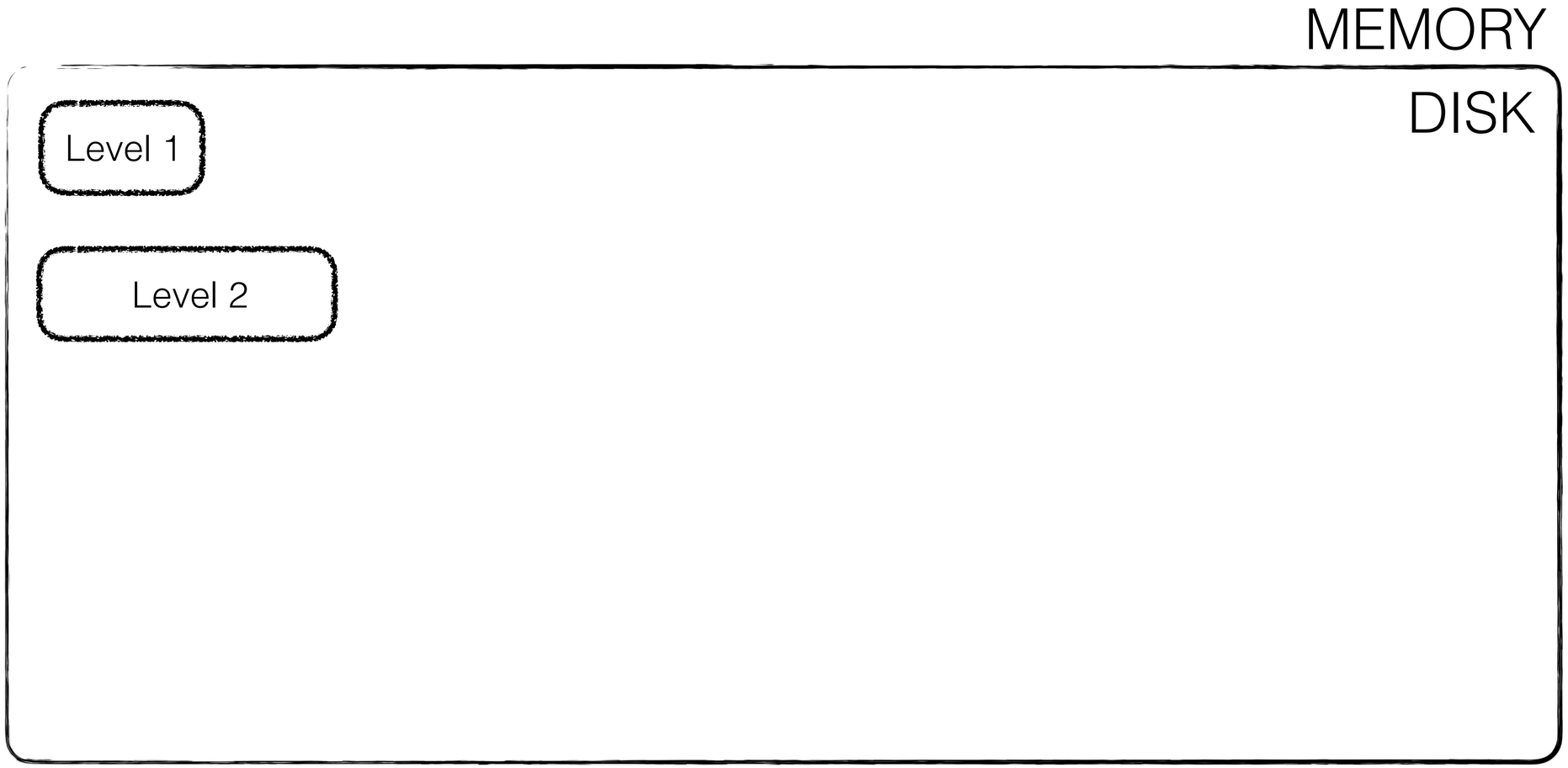

insert (key-value)
buffer
MEMORY
DISK
Level 1

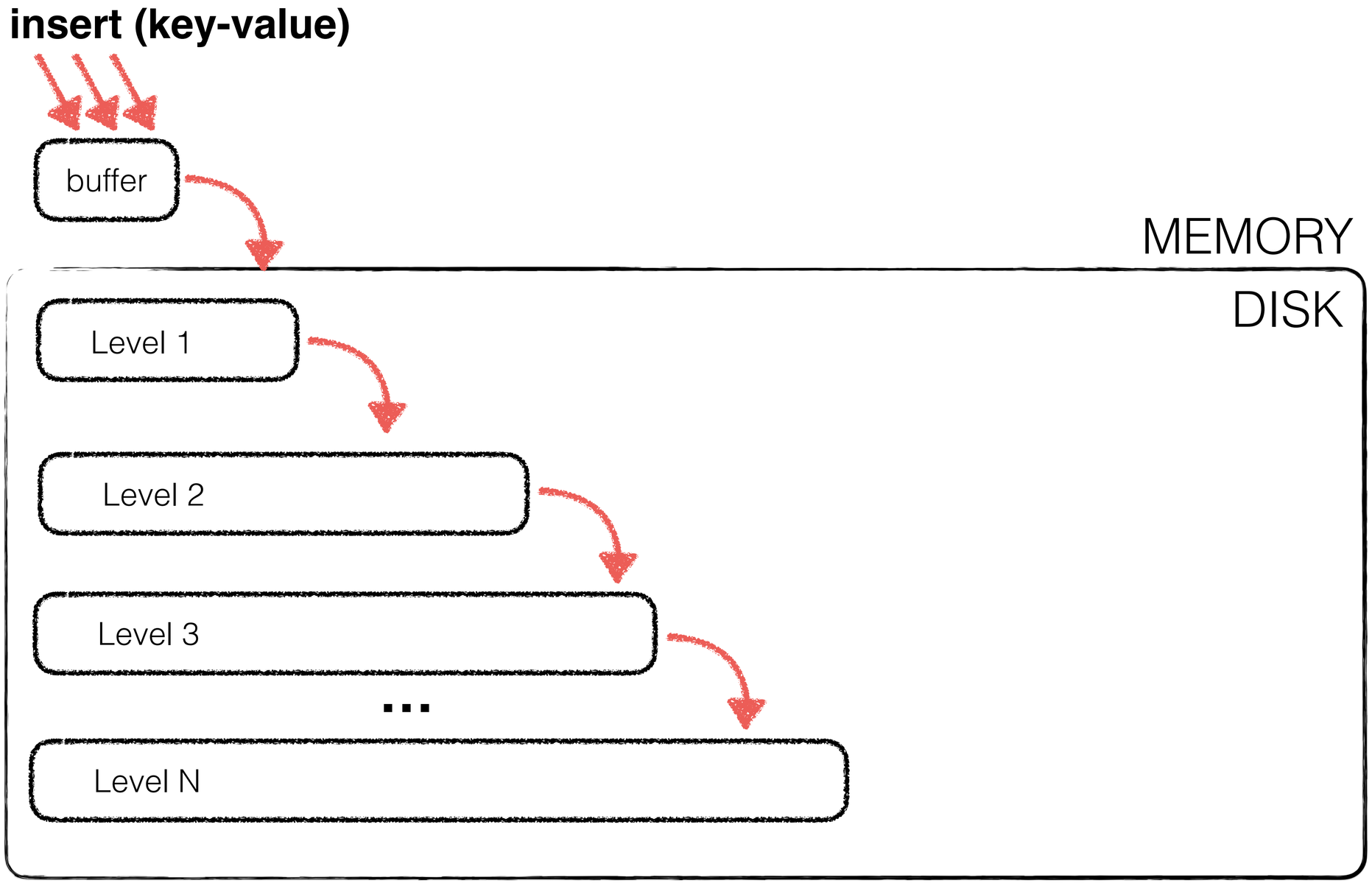

MEMORY
DISK
Level 1

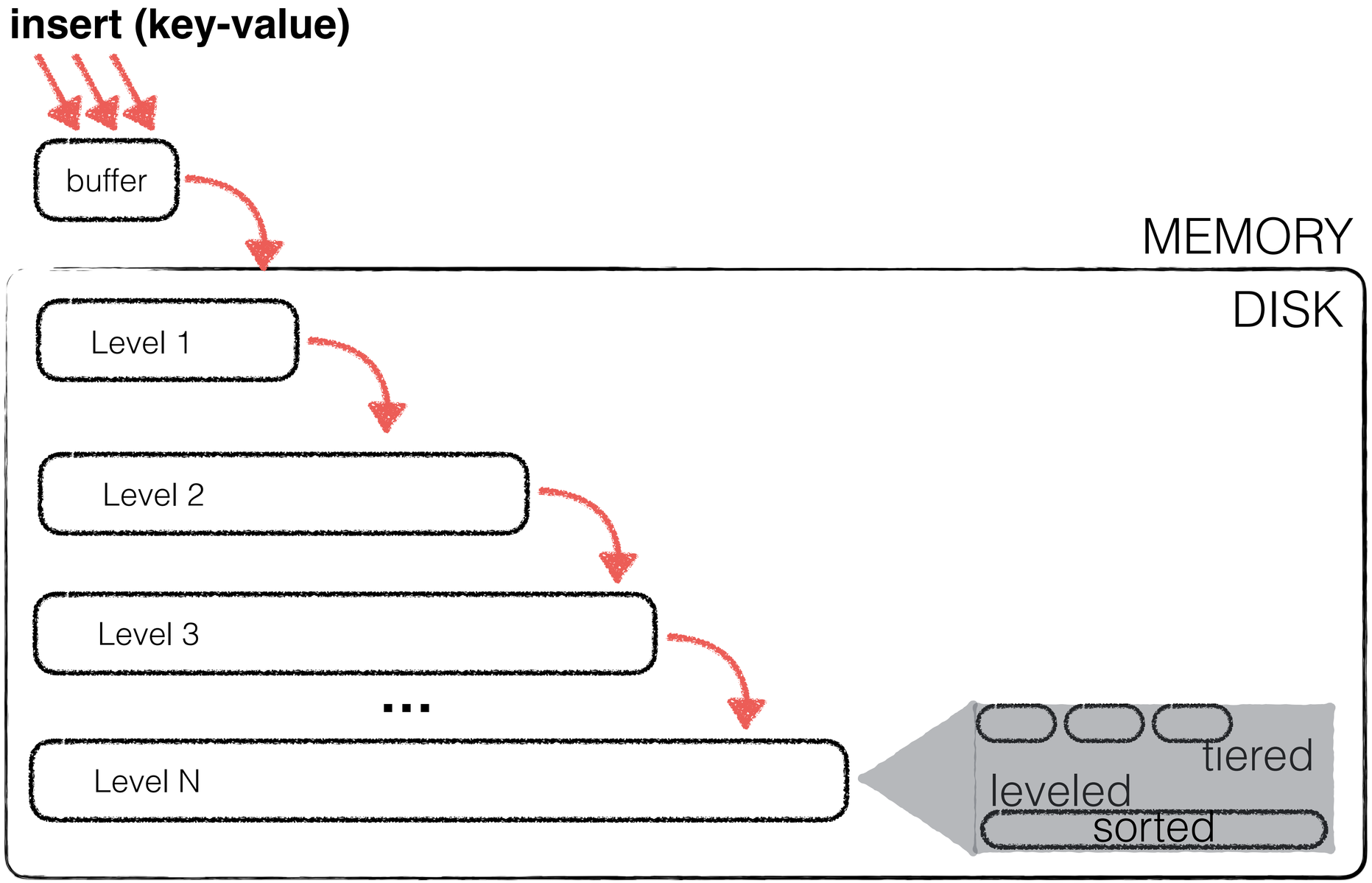

MEMORY
DISK
Level 1

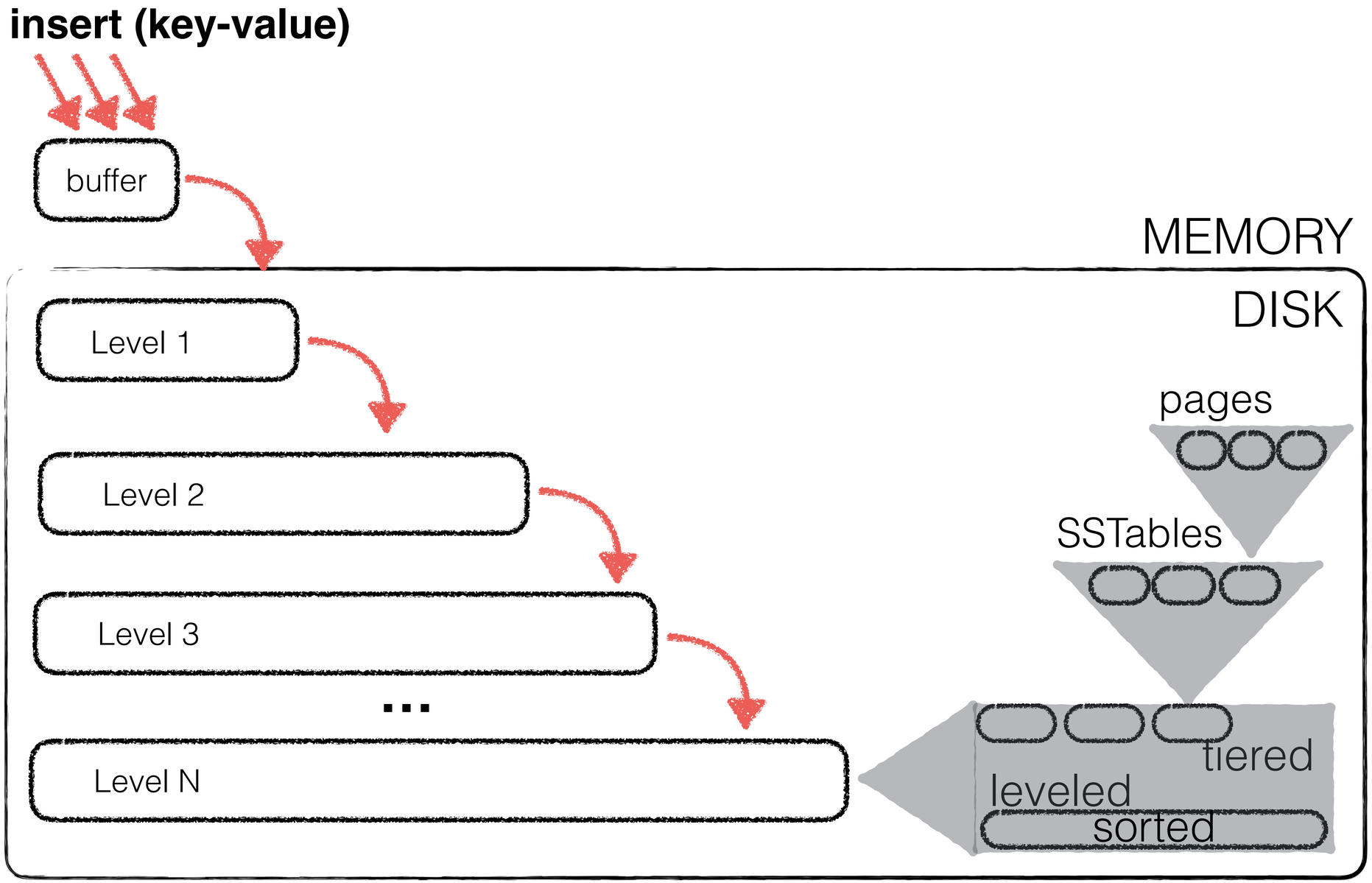

insert (key-value)
buffer
MEMORY
DISK
Level 1

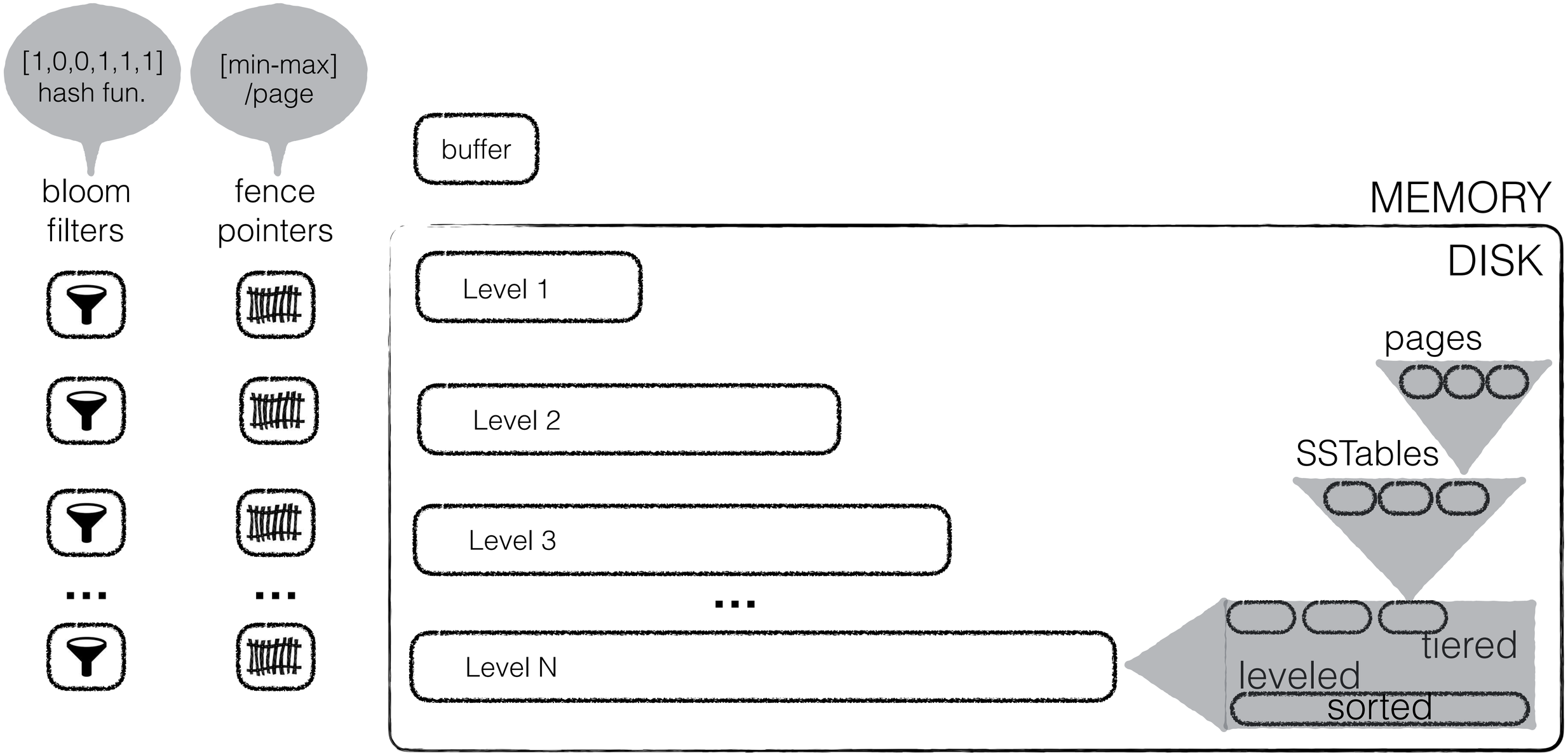

MEMORY
DISK
Level 2

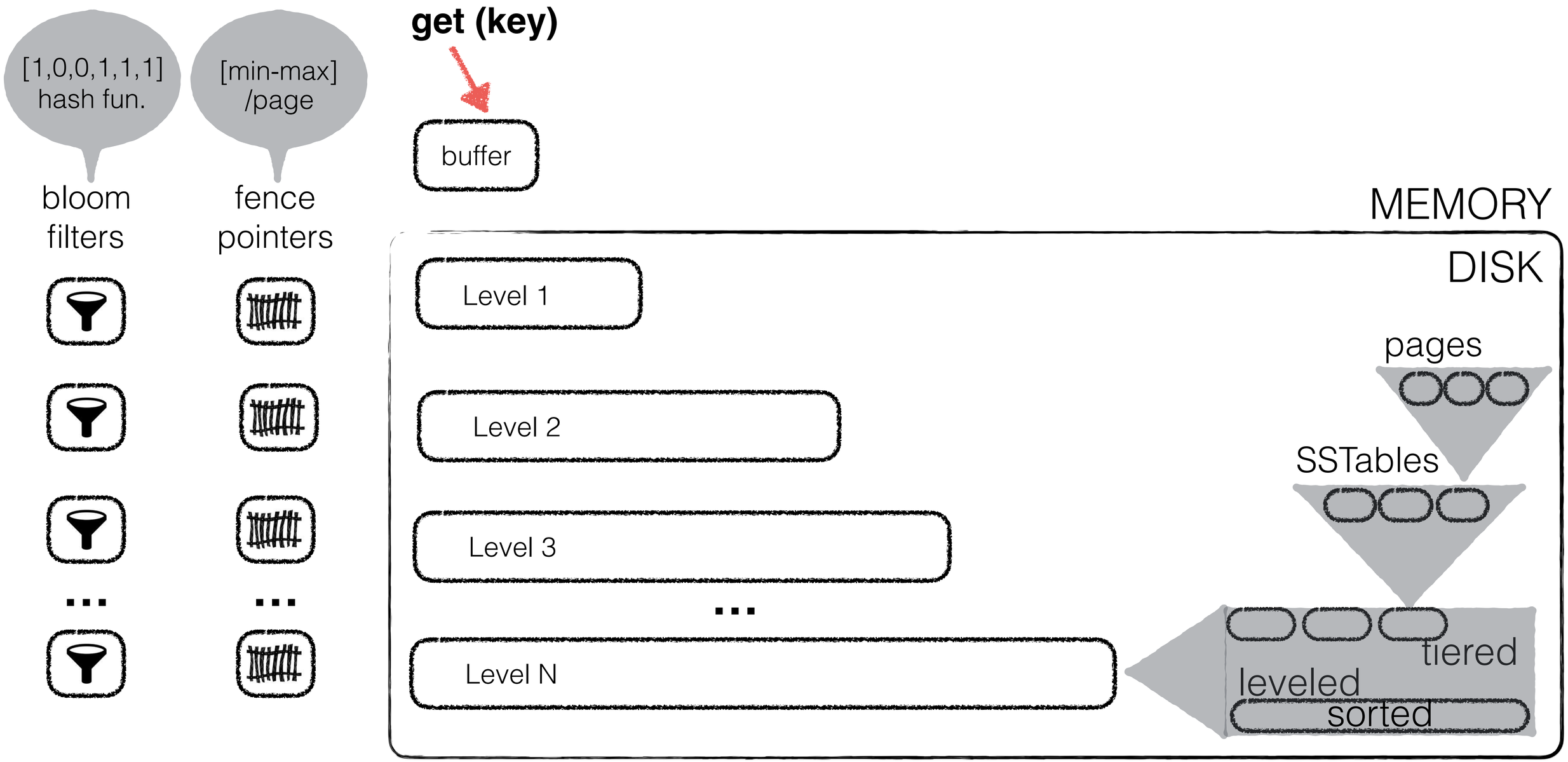

MEMORY
DISK
Level 1
Level 2

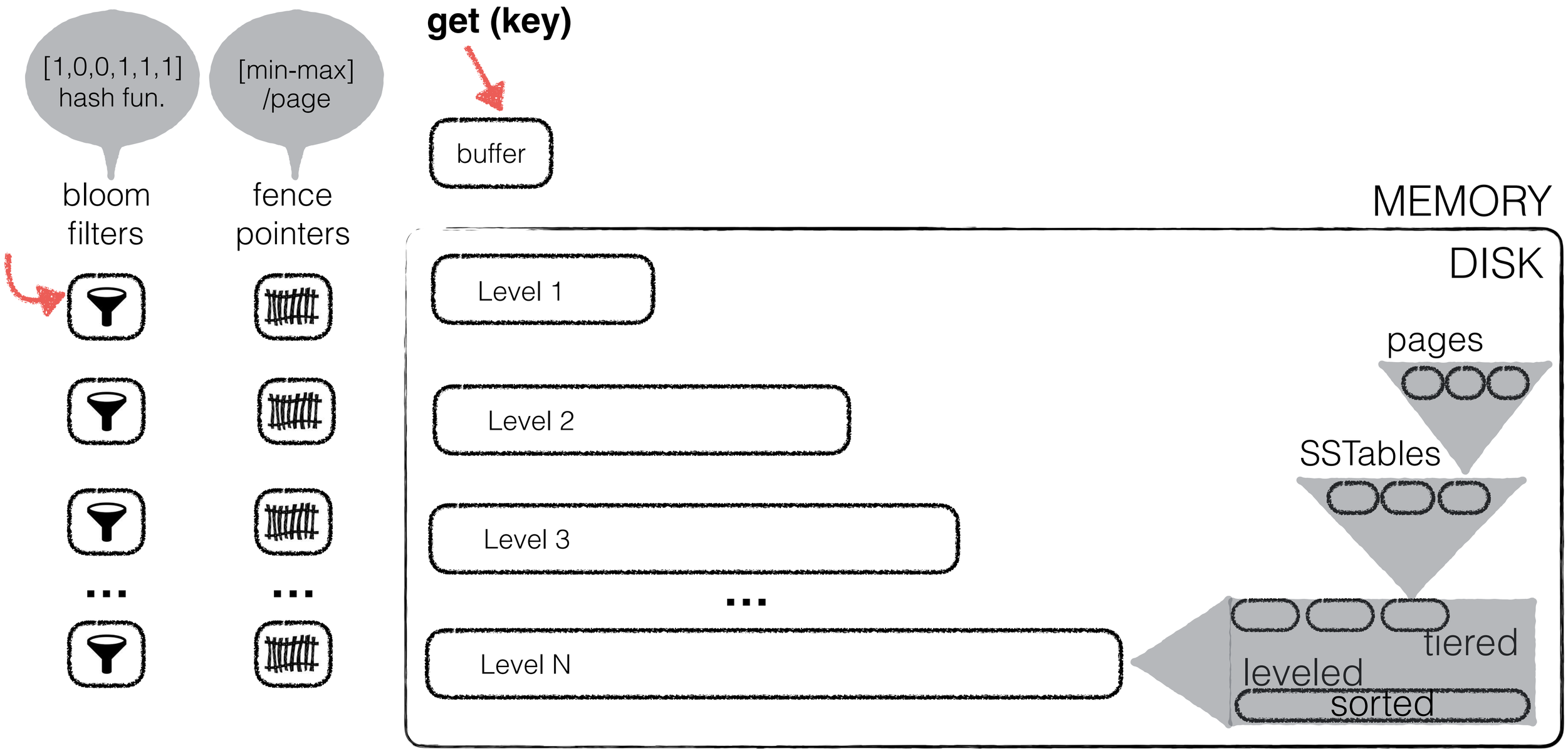

insert (key-value)
buffer
MEMORY
DISK
Level 1
Level 2
Level 3
...
Level N

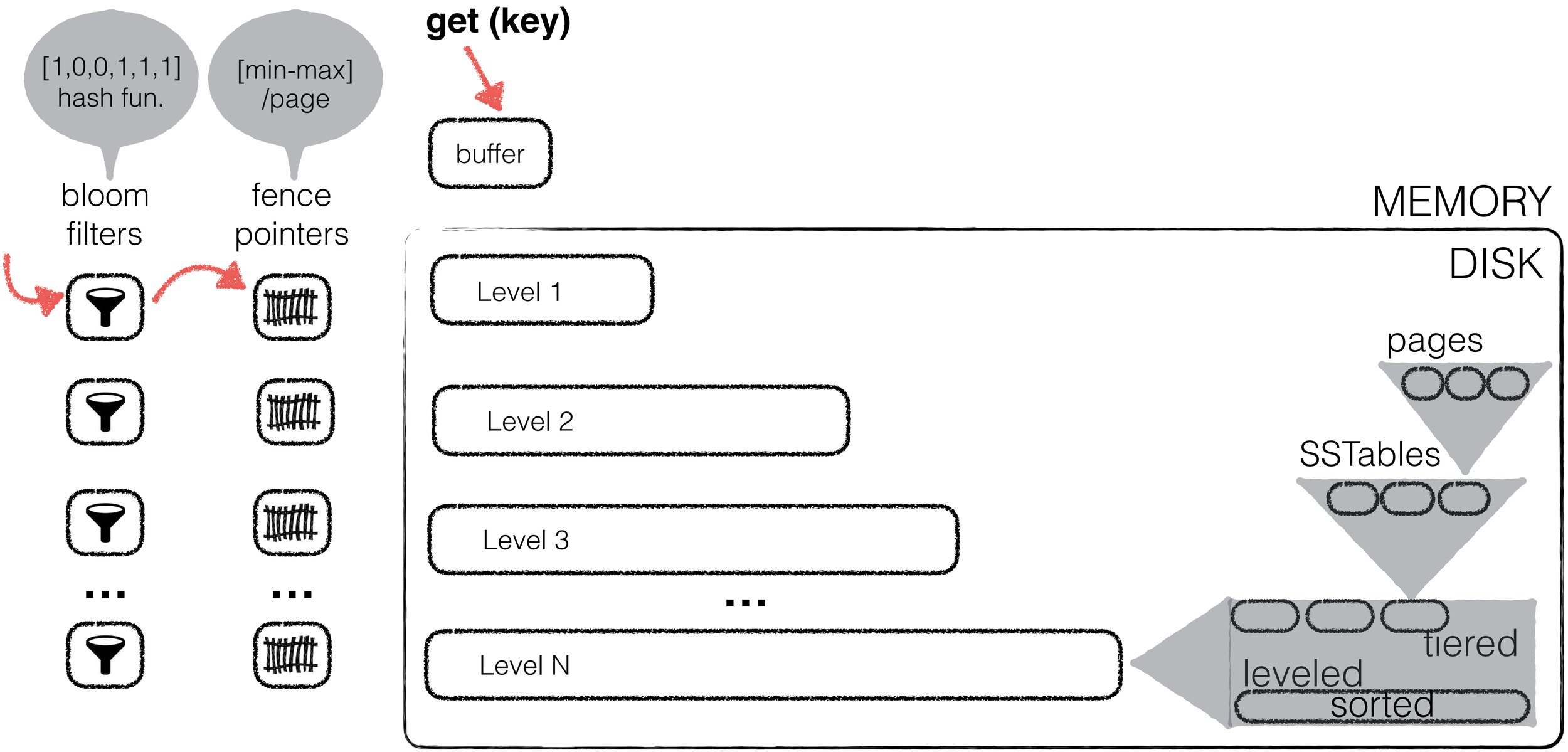

insert (key-value)
buffer
MEMORY
DISK
Level 1
Level 2
Level 3
...
Level N
tiered
leveled
sorted

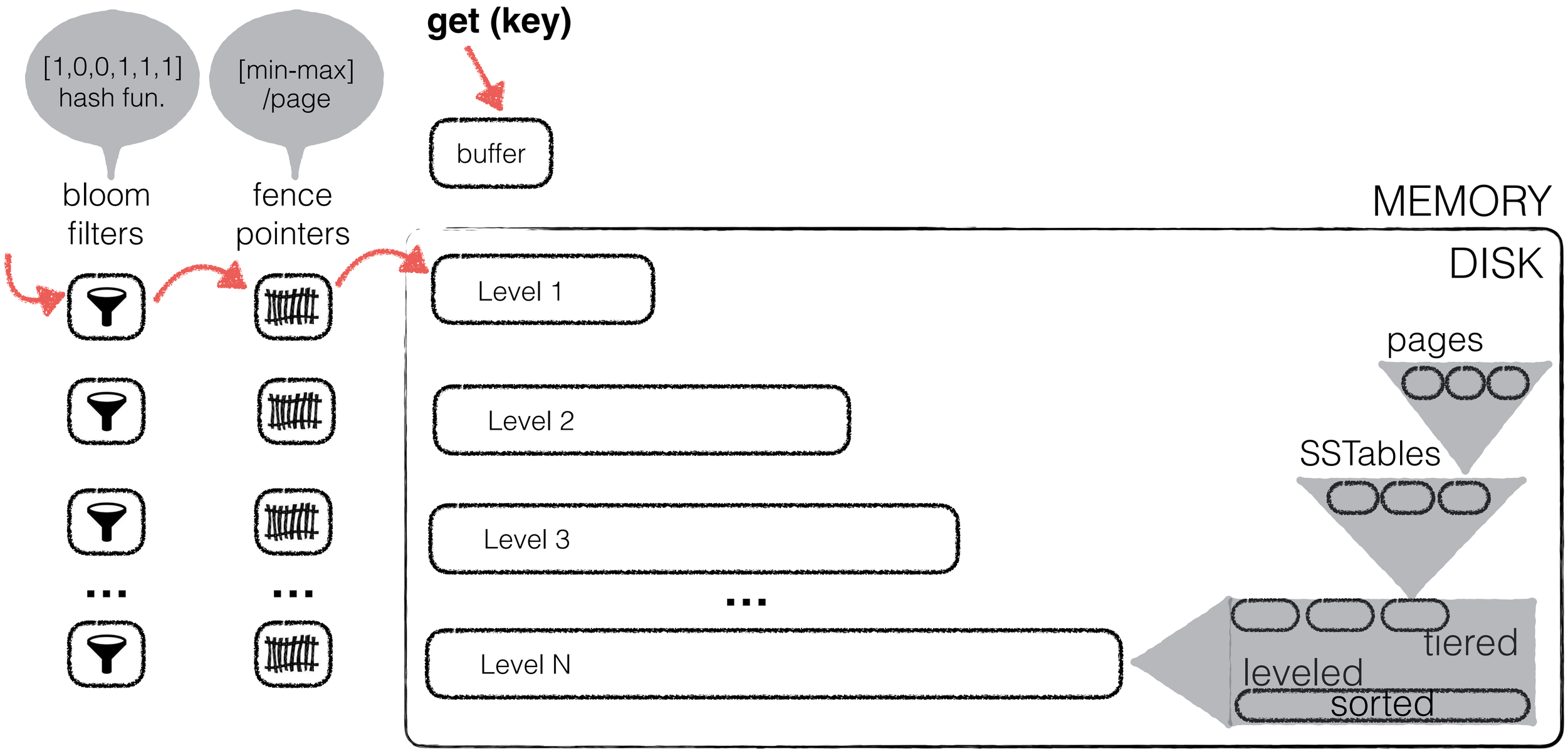

insert (key-value)
buffer
MEMORY
DISK
Level 1
Level 2
Level 3
...
Level N
pages
SSTables
tiered
leveled
sorted

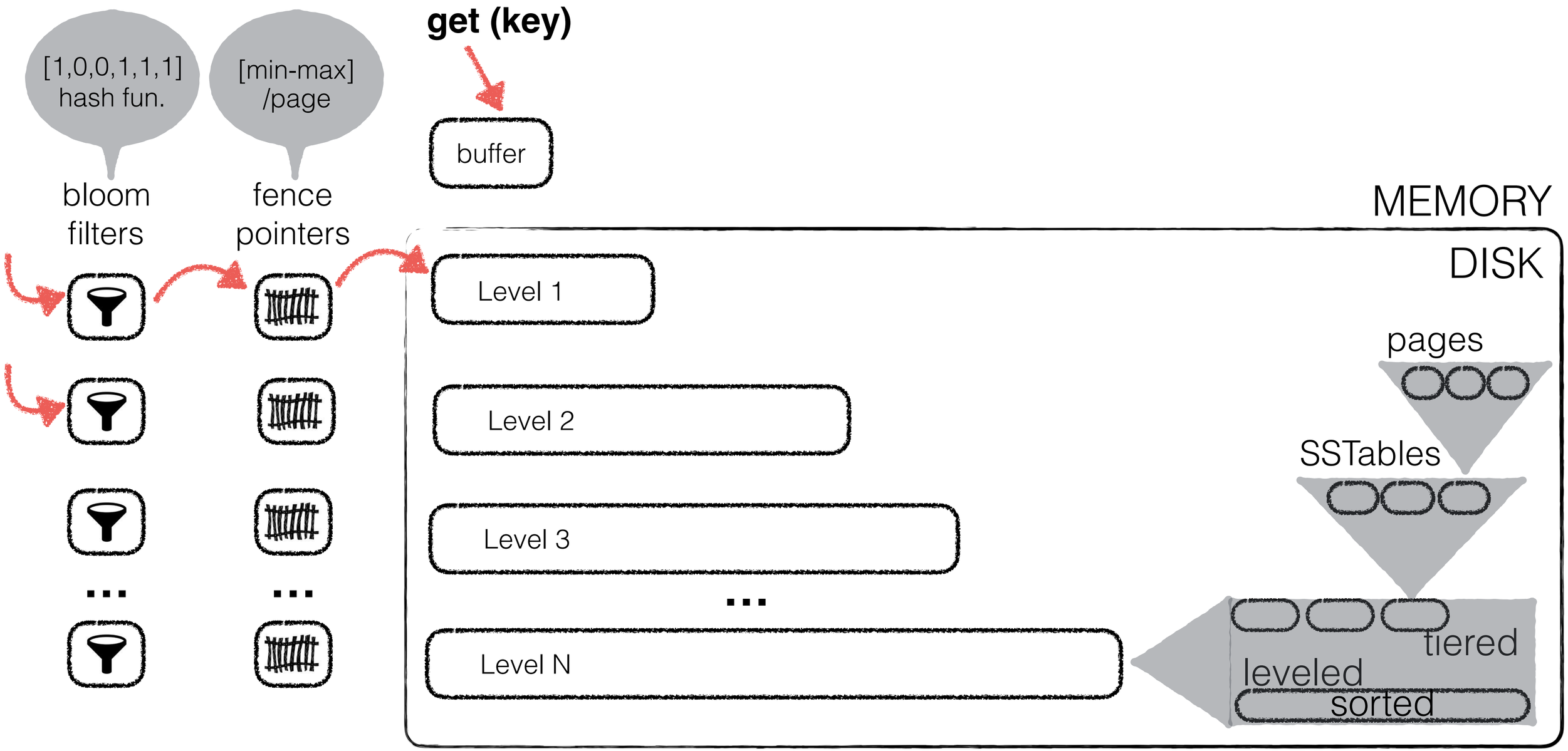

[1,0,0,1,1,1]
hash fun.
[min-max]
/page
bloom
filters
fence
pointers
buffer
MEMORY
DISK
Level 1
Level 2
Level 3
...
Level N
pages
SSTables
tiered
leveled
sorted

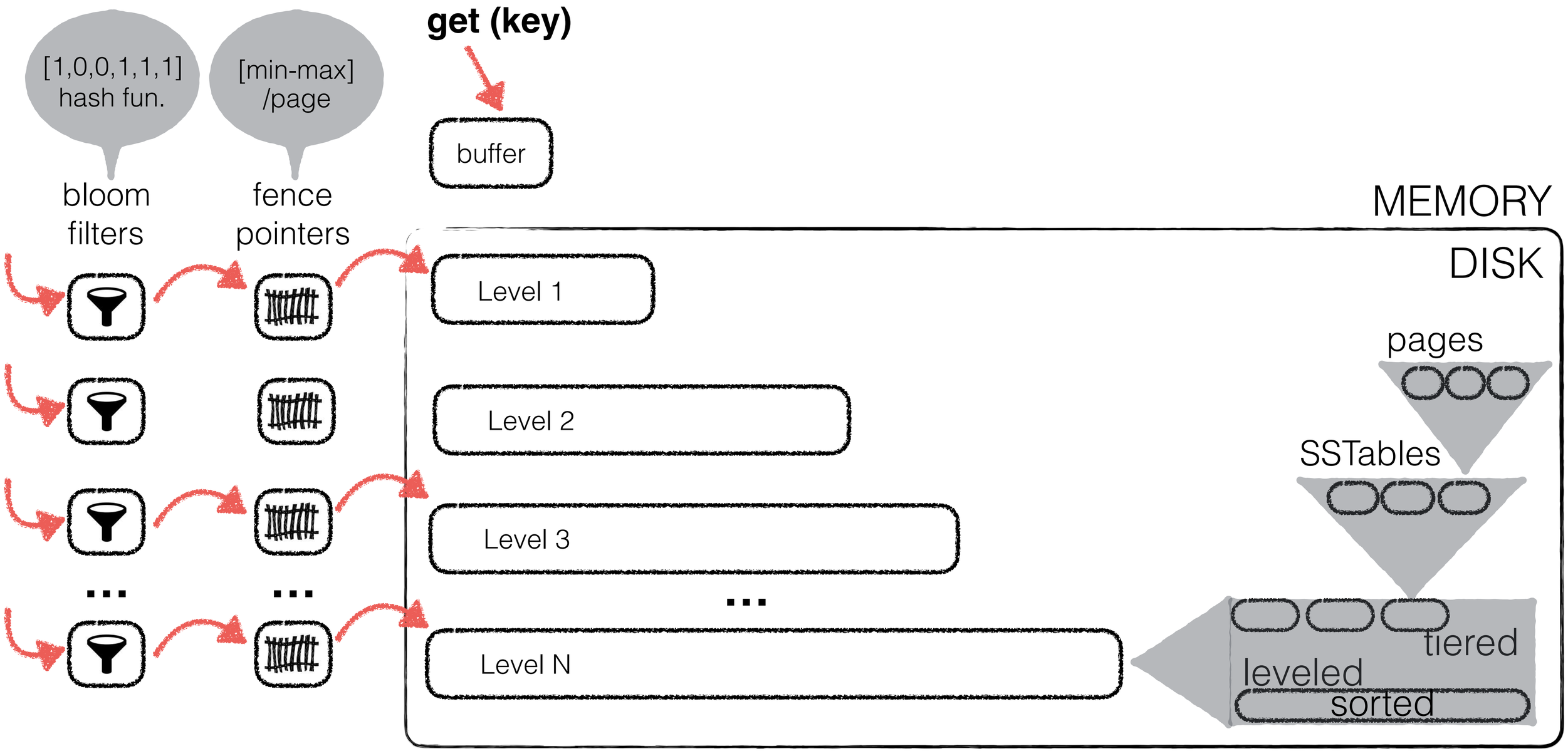

[1,0,0,1,1,1]
hash fun.
[min-max]
/page
bloom
filters
fence
pointers
get (key)
buffer
MEMORY
DISK
Level 1
Level 2
Level 3
...
Level N
pages
SSTables
tiered
leveled
sorted

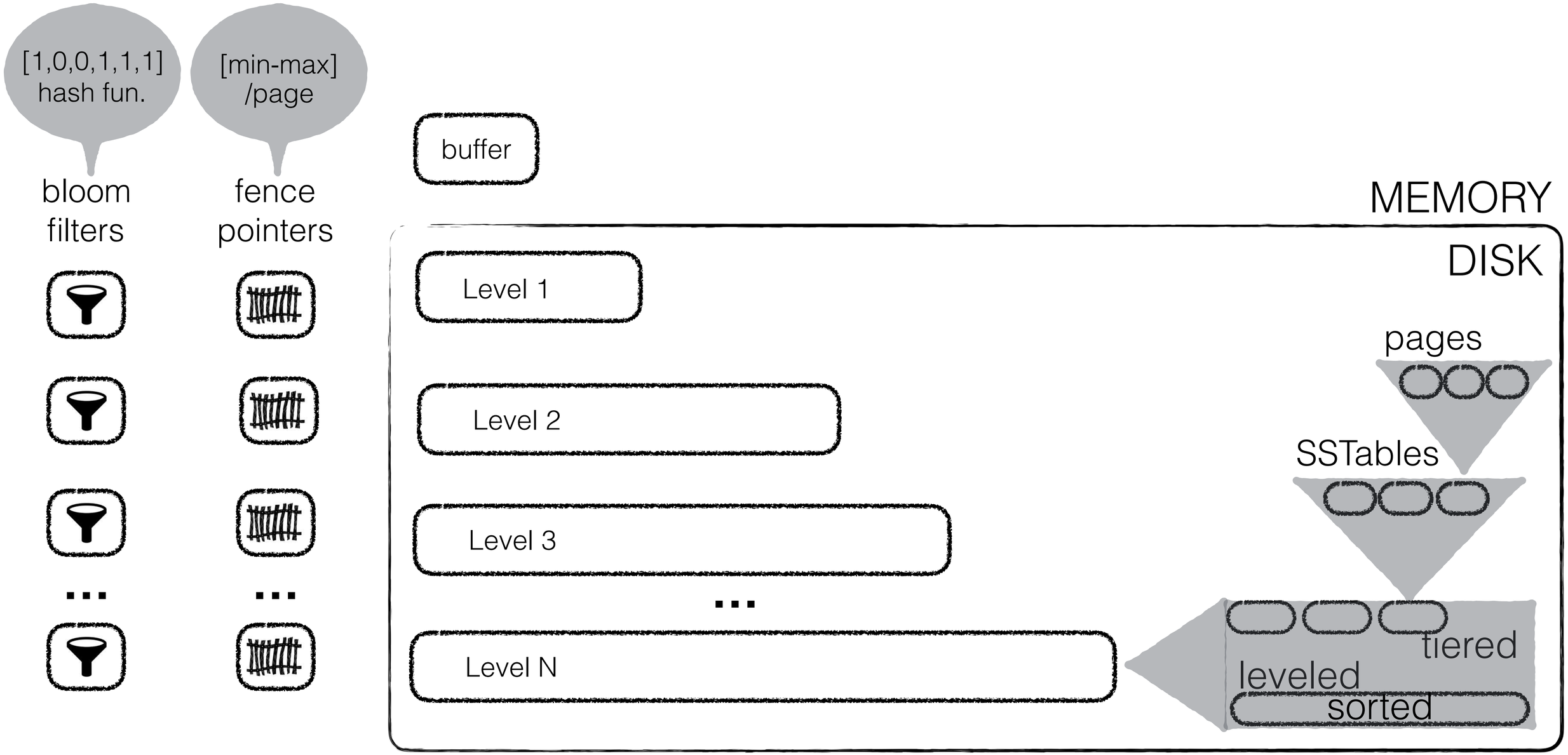

[1,0,0,1,1,1]
hash fun.
[min-max]
/page
bloom
filters
fence
pointers
get (key)
buffer
MEMORY
DISK
Level 1
Level 2
Level 3
...
Level N
pages
SSTables
tiered
leveled
sorted

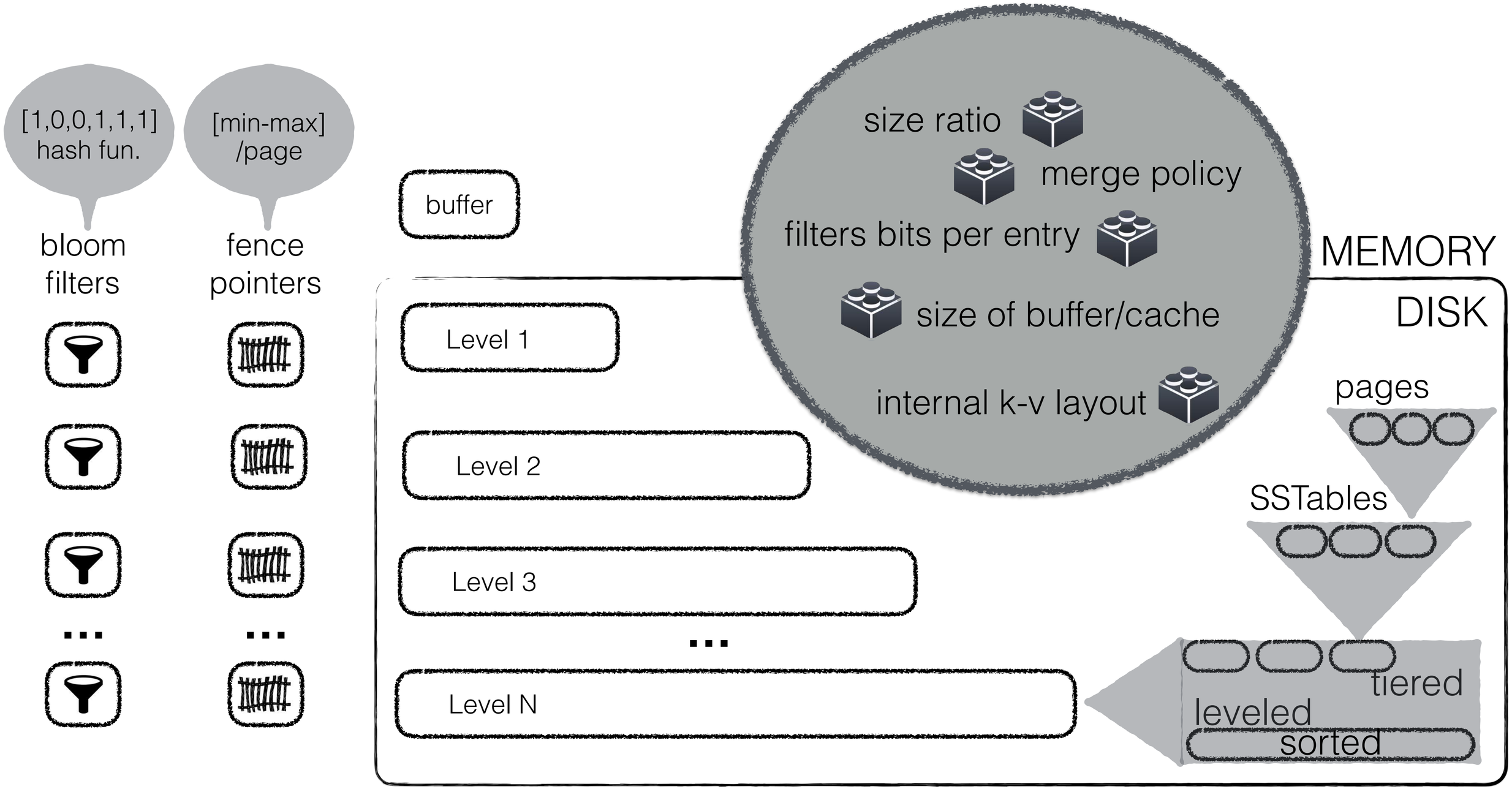

[1,0,0,1,1,1]
hash fun.
[min-max]
/page
bloom
filters
fence
pointers
...
...
get (key)
buffer
MEMORY
DISK
Level 1
Level 2
Level 3
...
Level N
pages
SSTables
tiered
leveled
sorted

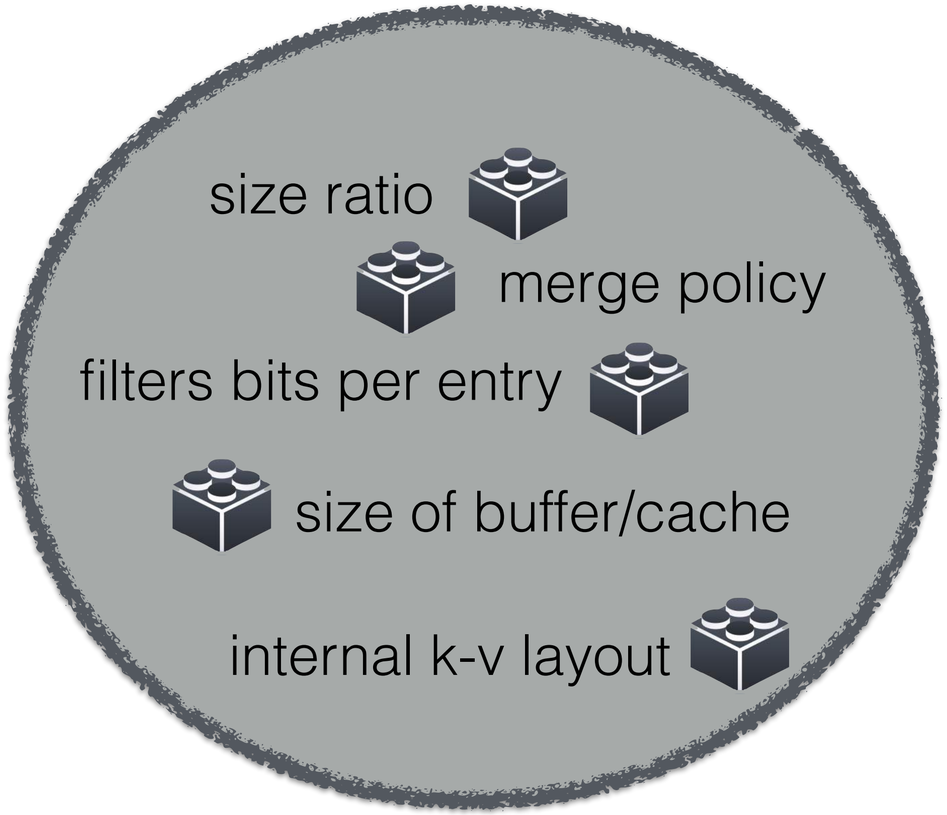



get (key)
[1,0,0,1,1,1]
hash fun.
[min-max]
/page
buffer
bloom
filters
fence
pointers
MEMORY
DISK
Level 1
Level 2
Level 3
...
Level N
pages
SSTables
tiered
leveled
sorted

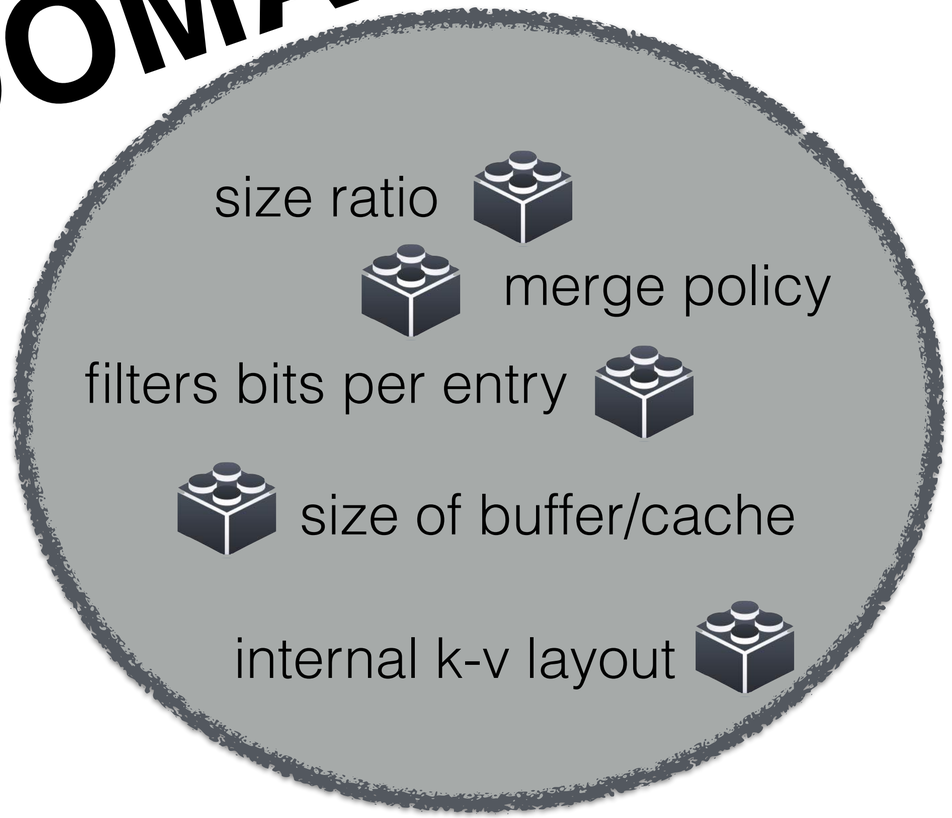

get (key)
[1,0,0,1,1,1]
hash fun.
[min-max]
/page
buffer
bloom
filters
fence
pointers
MEMORY
DISK
Level 1
Level 2
Level 3
...
Level N
pages
SSTables
tiered
leveled
sorted


[1,0,0,1,1,1]
hash fun.
[min-max]
/page
bloom
filters
fence
pointers
buffer
MEMORY
DISK
Level 1
Level 2
Level 3
...
Level N
pages
SSTables
tiered
leveled
sorted

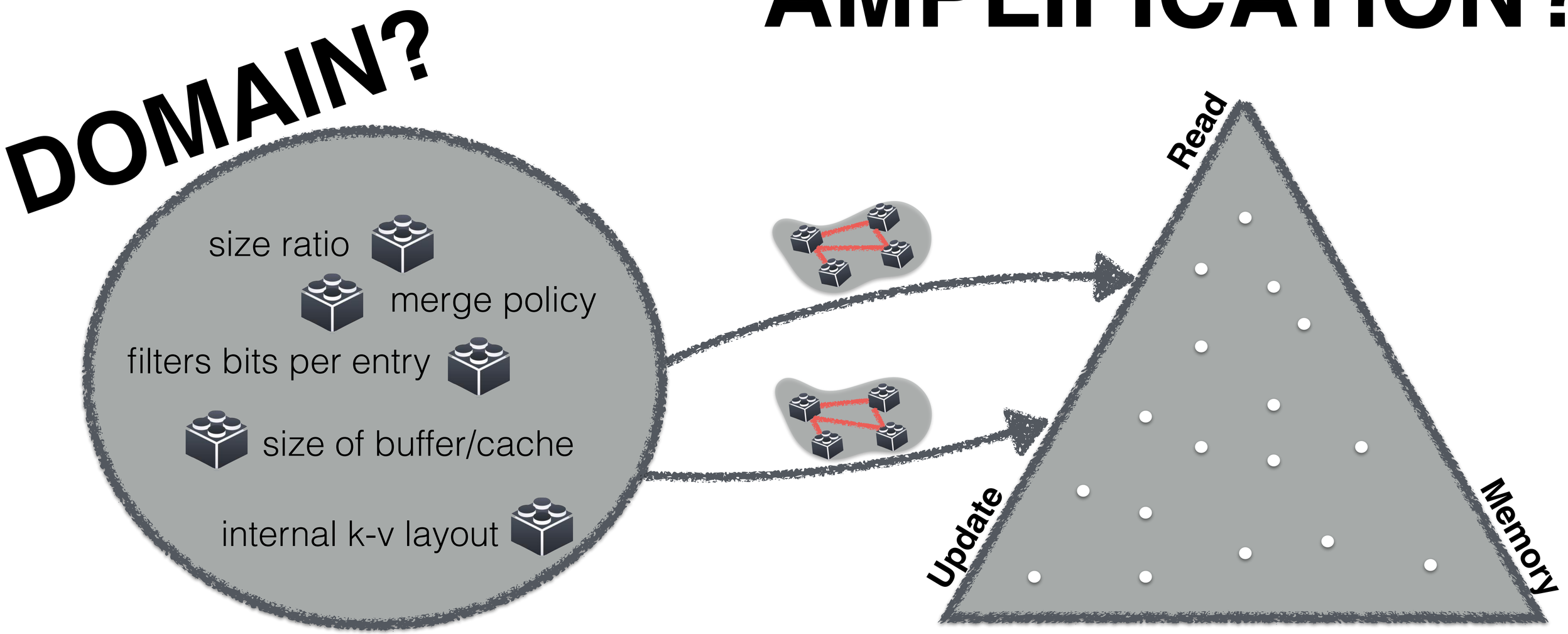



size ratio
merge policy
filters bits per entry
size of buffer/cache
internal k-v layout

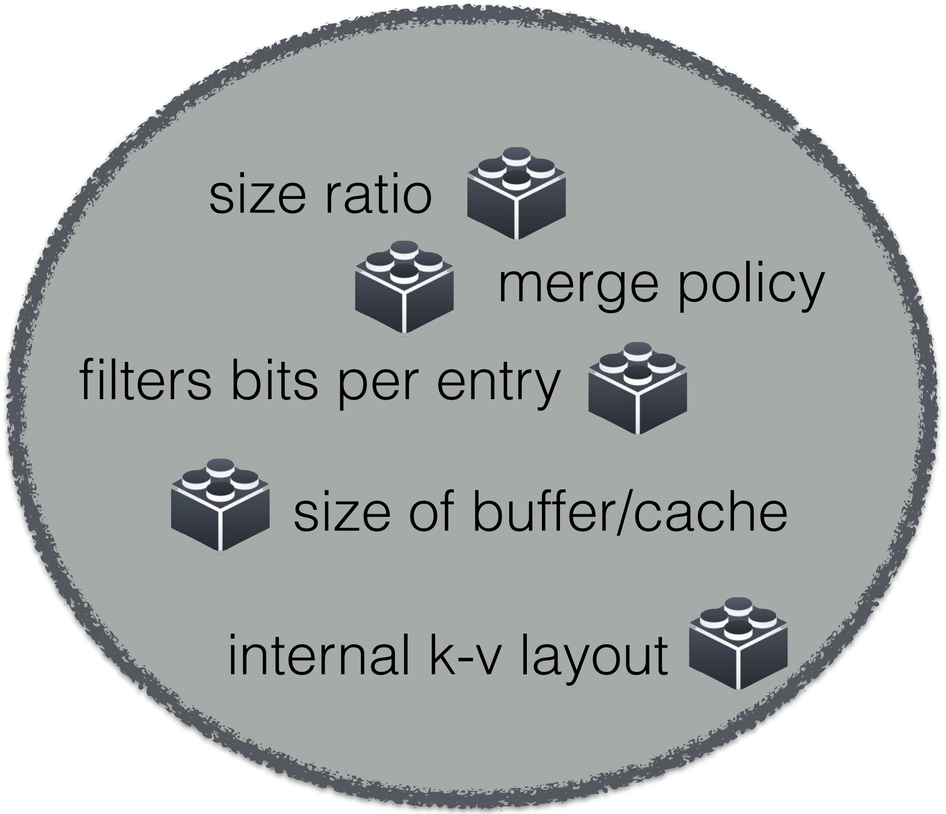
DOMAIN?


size ratio
merge policy
filters bits per entry
size of buffer/cache
internal k-v layout

AMPLIFICATION?



LSM-trees


size ratio
merge policy
filters bits per entry
size of buffer/cache
internal k-v layout

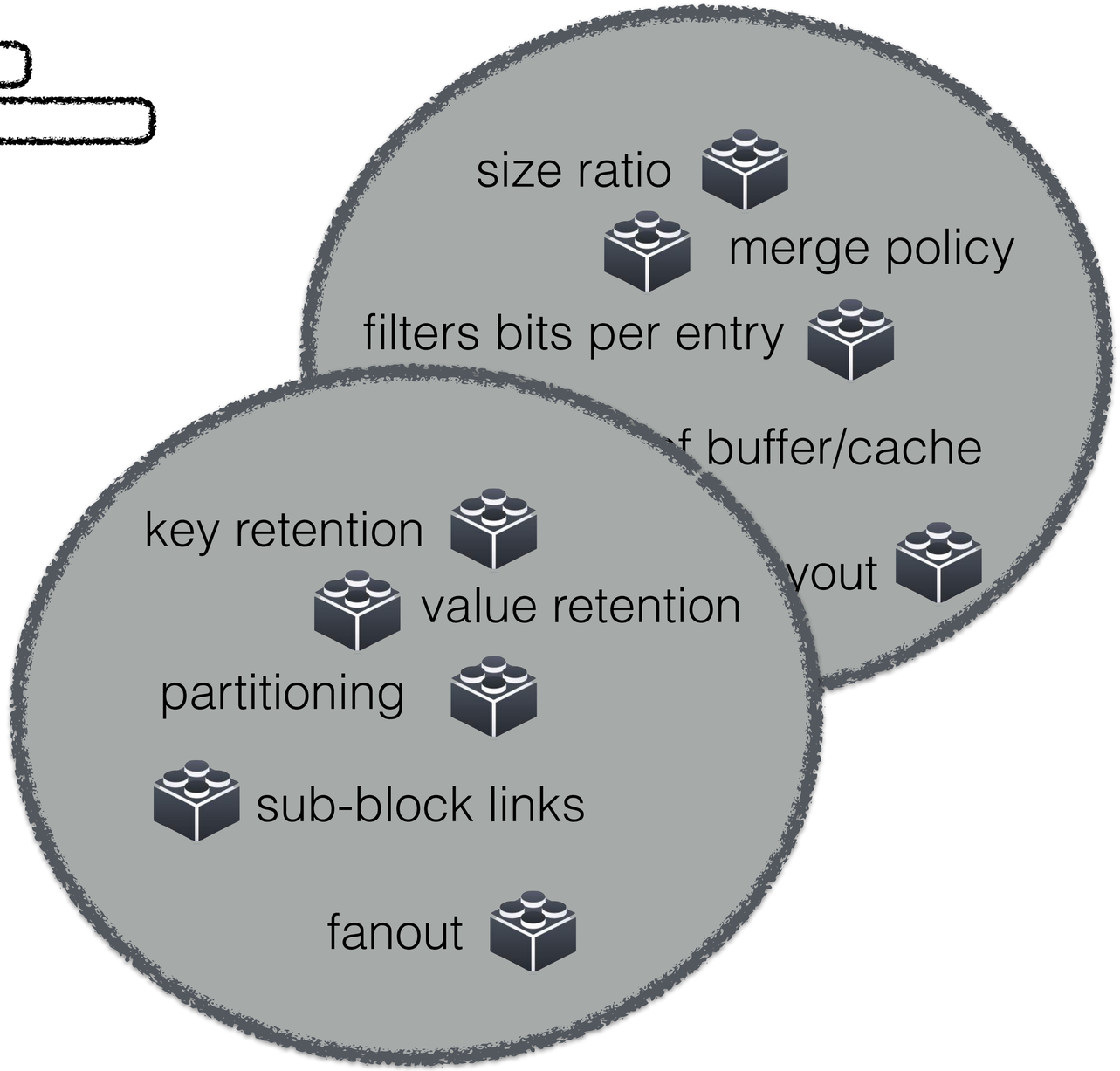
LSM-trees

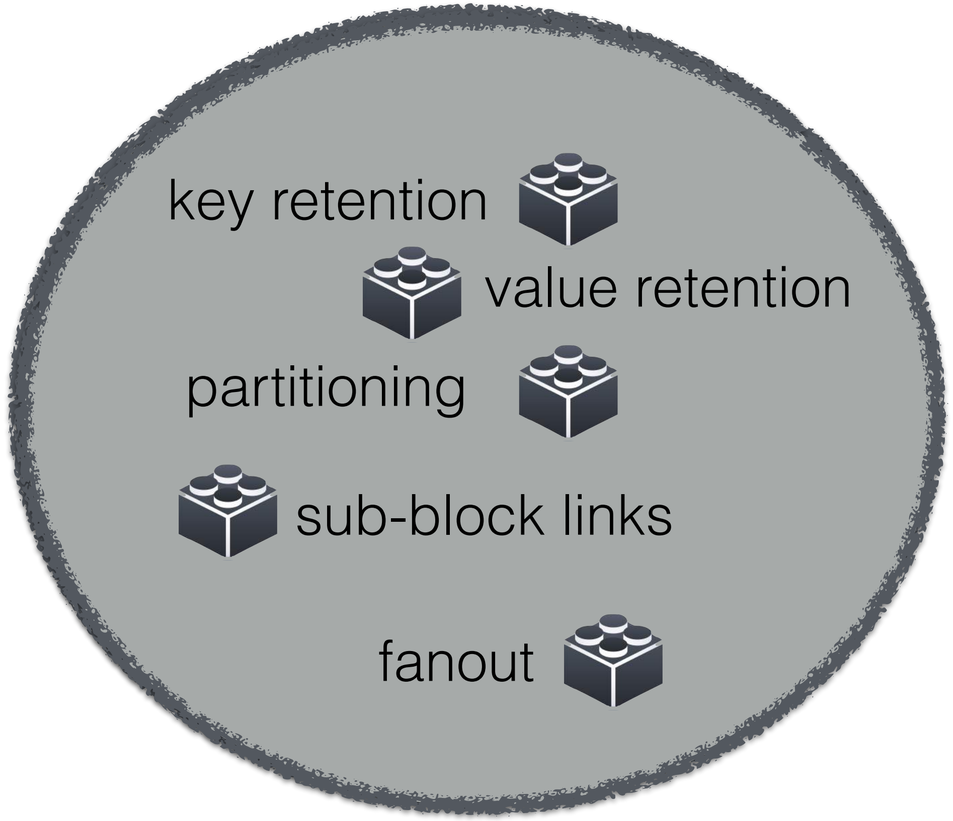
B-trees

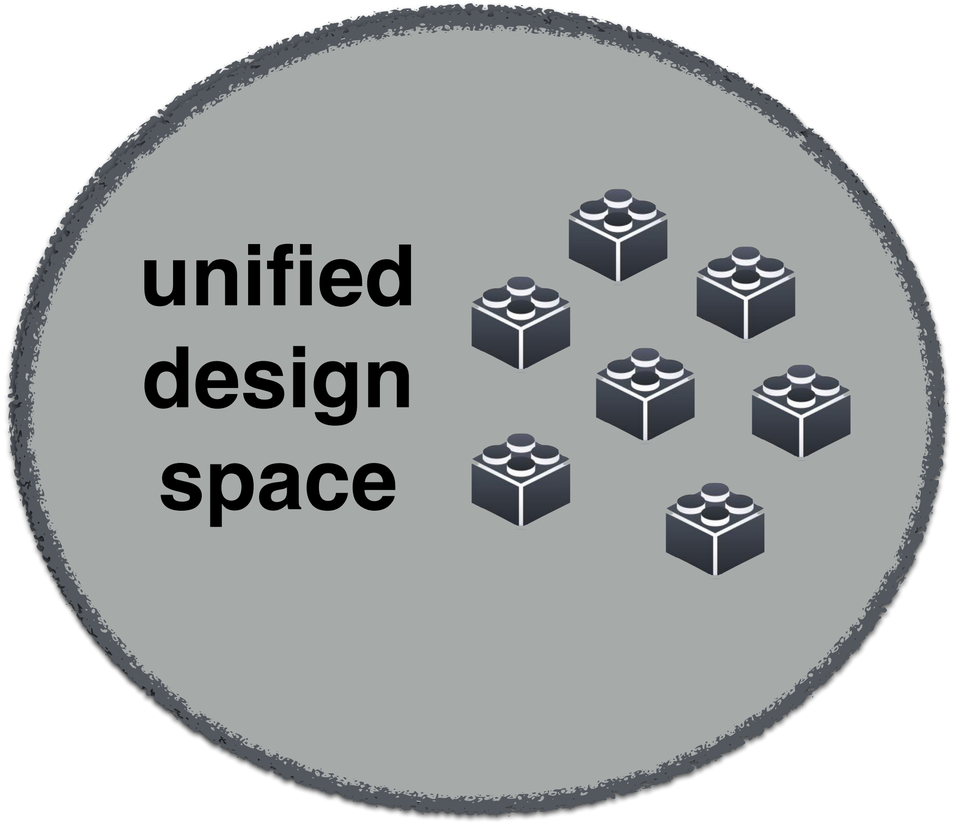
Logs

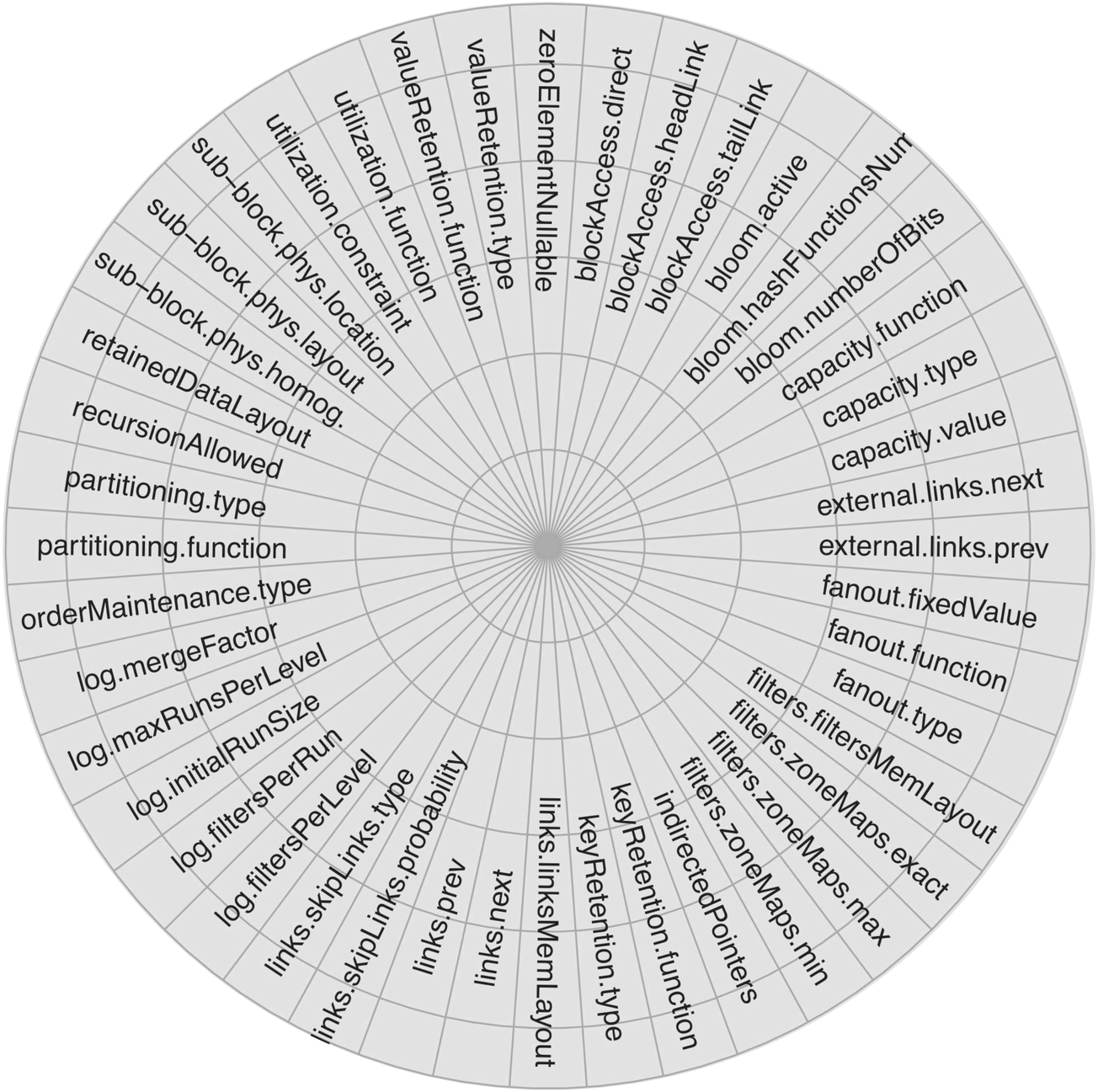
Arrays

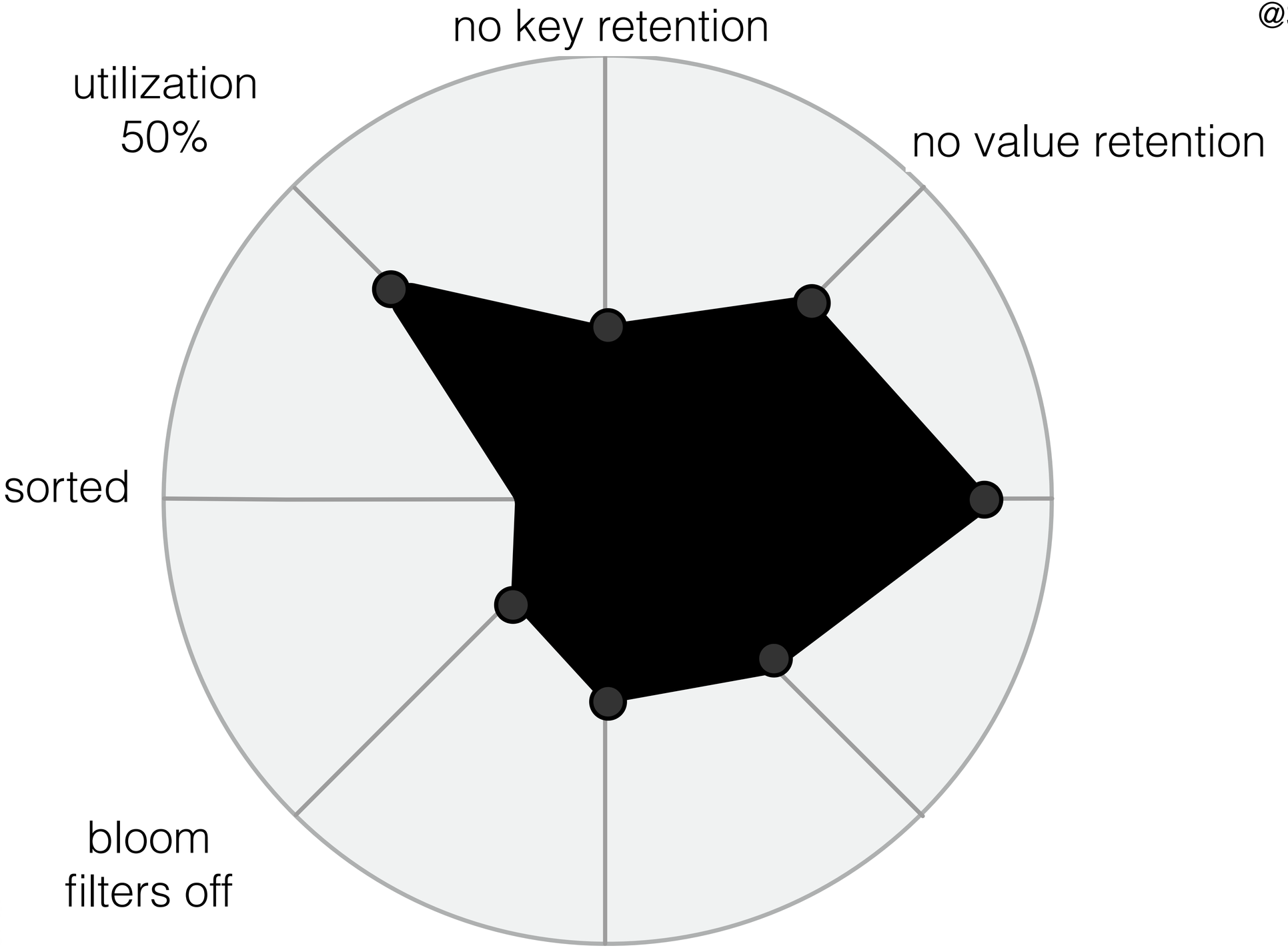
Bitmaps

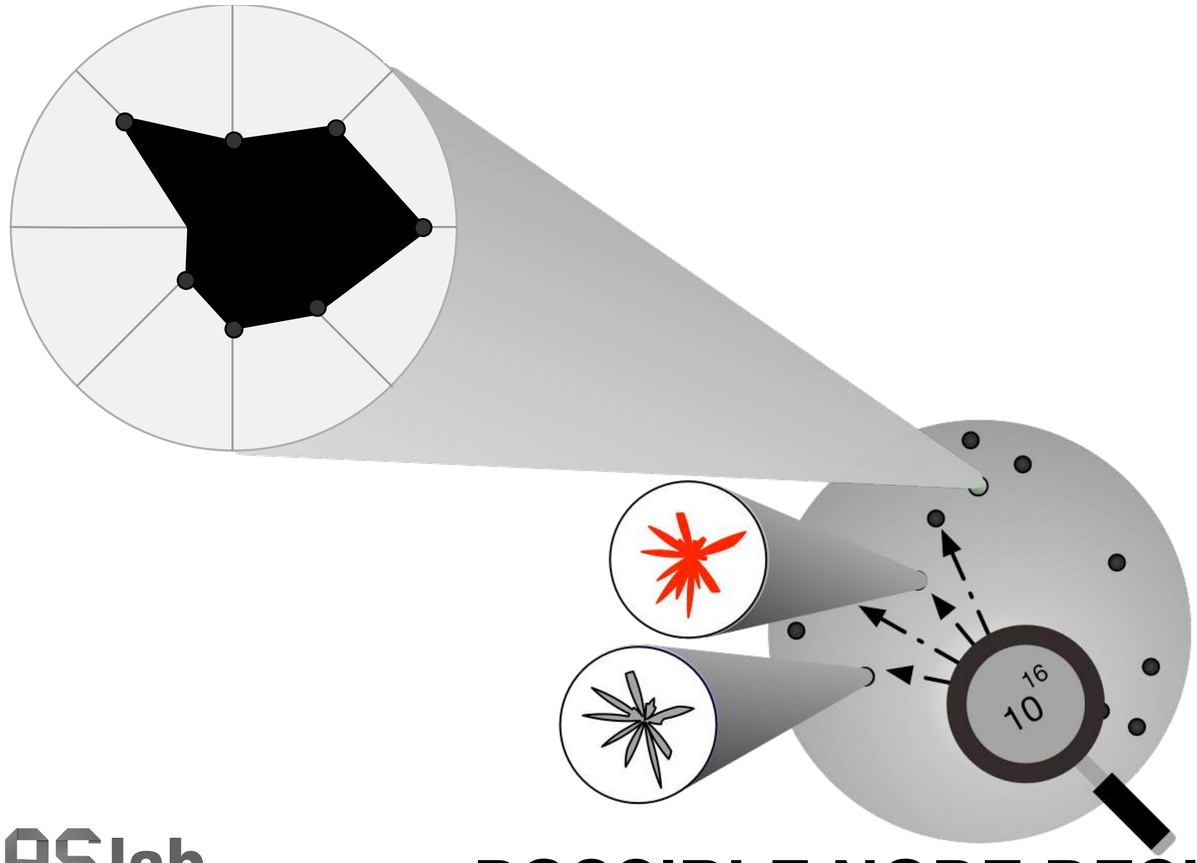



key retention
value retention
partitioning
sub-block links
fanout

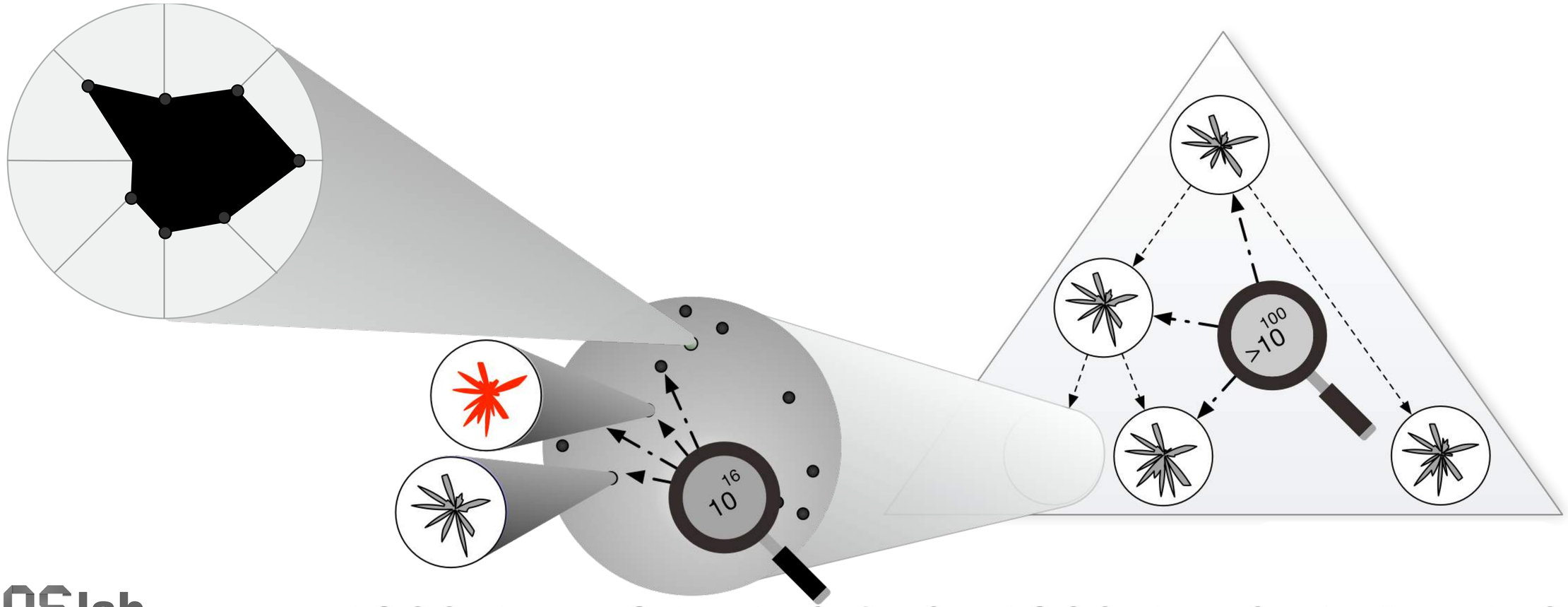

unified
design
space


zeroElementNullable
blockAccess.direct
blockAccess.headLink
blockAccess.tailLink
bloom.active
bloom.hashFunctionsNum
bloom.numberOfBits
capacity.function
capacity.type
capacity.value
external.links.next
external.links.prev
fanout.fixedValue
fanout.function
fanout.type
filters.filtersMemLayout
filters.zoneMaps.exact
filters.zoneMaps.max
filters.zoneMaps.min
indirectedPointers
keyRetention.function
keyRetention.type
links.linksMemLayout
links.next
links.prev
links.skipLinks.probability
links.skipLinks.type
log.filtersPerLevel
log.filtersPerRun
log.initialRunSize
log.maxRunsPerLevel
log.mergeFactor
orderMaintenance.type
partitioning.function
partitioning.type
recursionAllowed
retainedDataLayout
sub–block.phys.homog.
sub–block.phys.layout
sub–block.phys.location
utilization.constraint
utilization.function
valueRetention.function
valueRetention.type

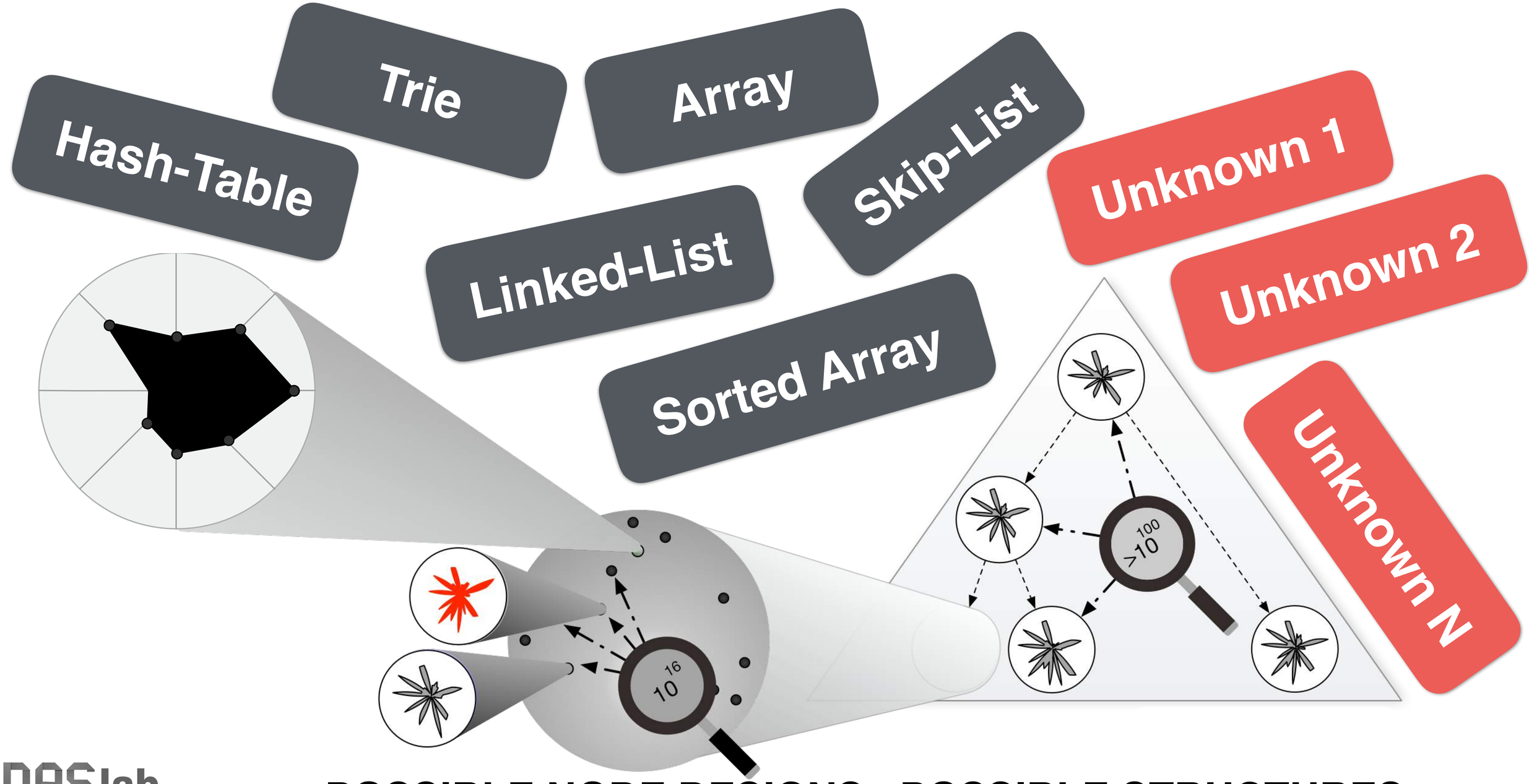

no key retention
no value retention
bloom
filters off
sorted
utilization
50%



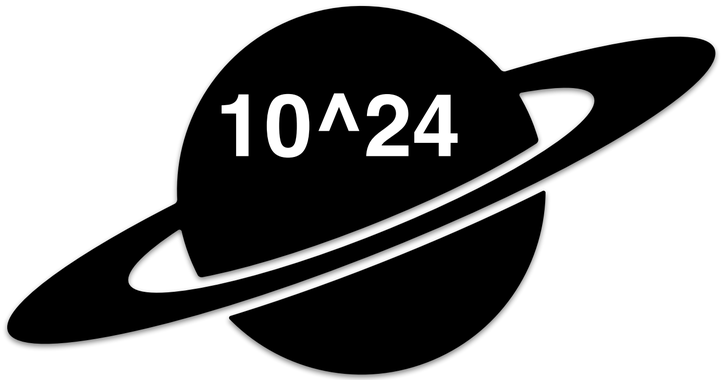
POSSIBLE NODE DESIGNS



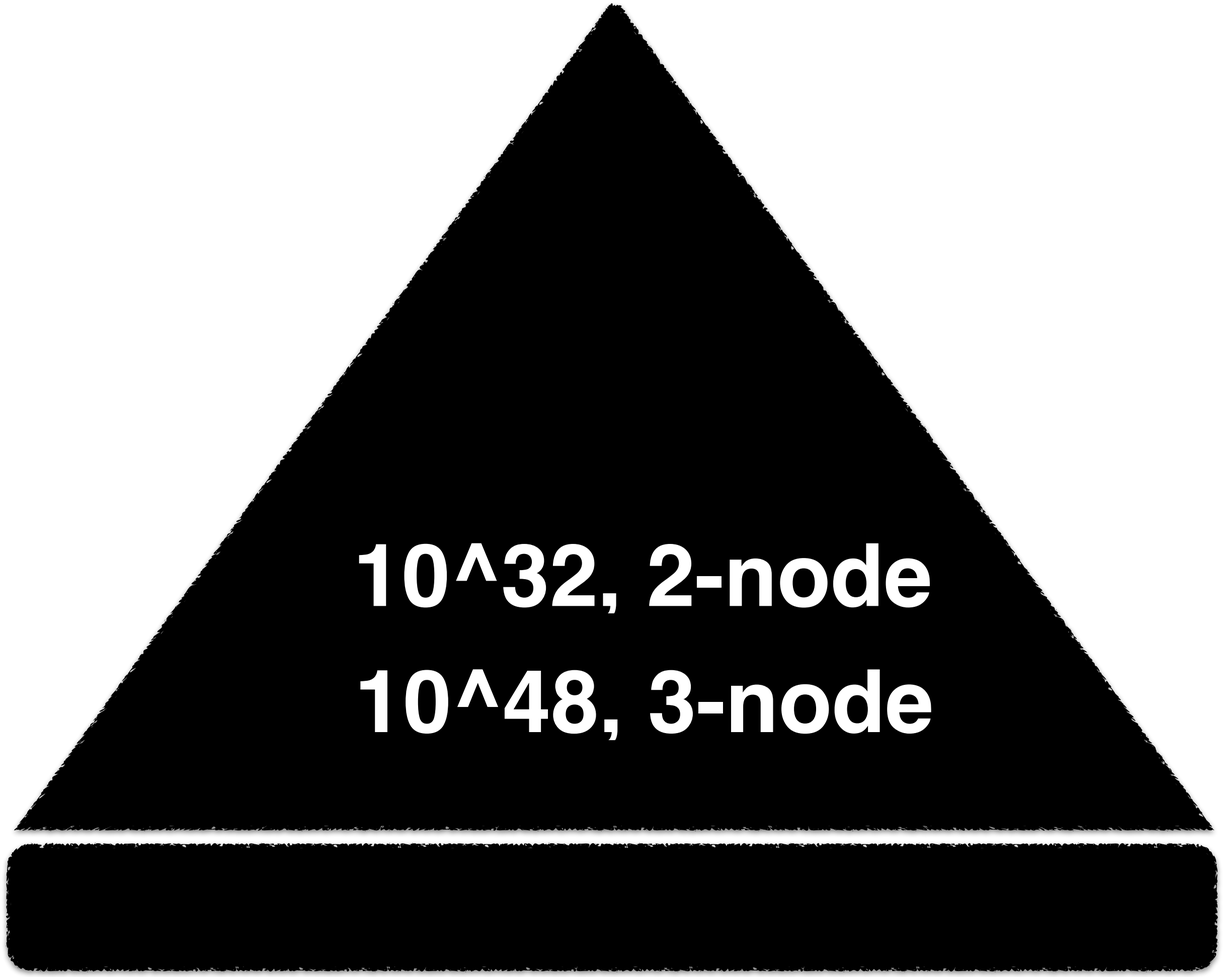
POSSIBLE NODE DESIGNS **POSSIBLE STRUCTURES**



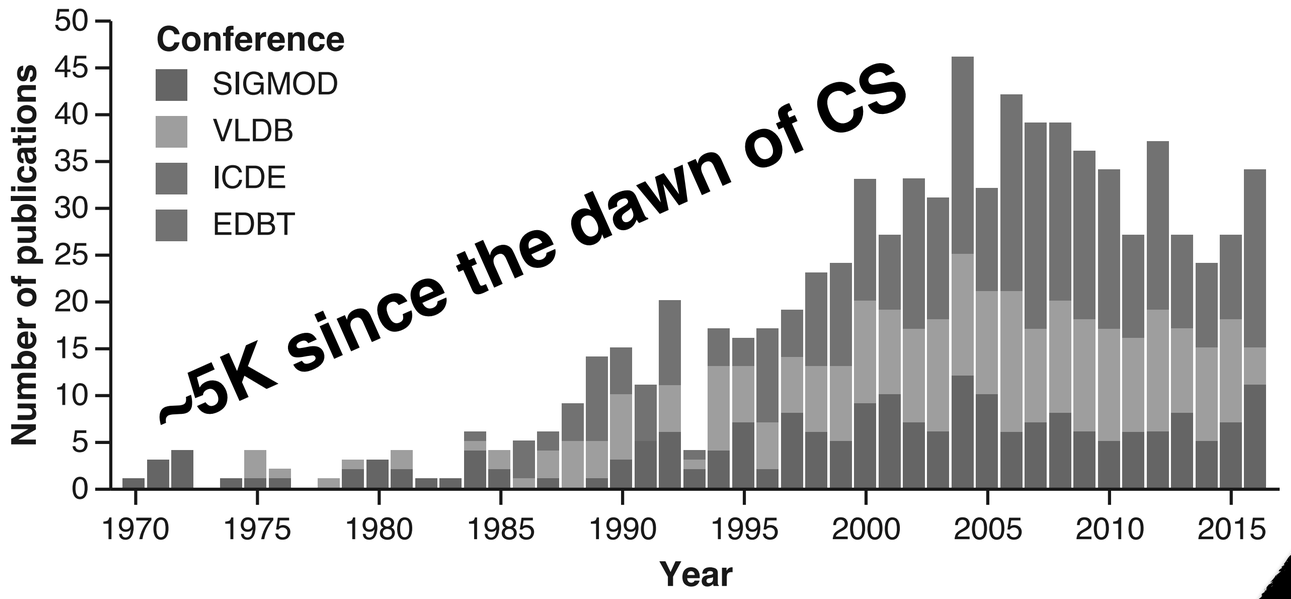
POSSIBLE NODE DESIGNS POSSIBLE STRUCTURES

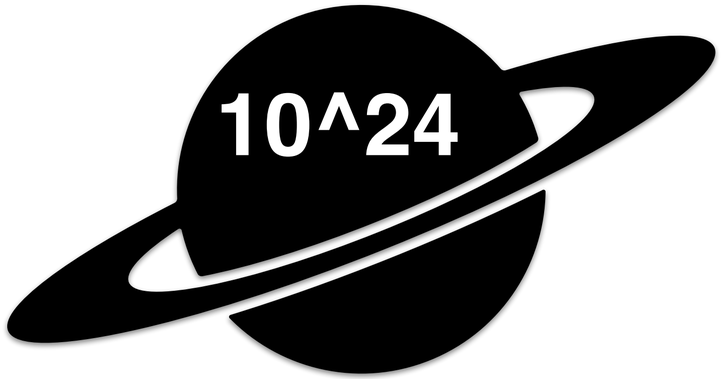


STARS IN THE SKY

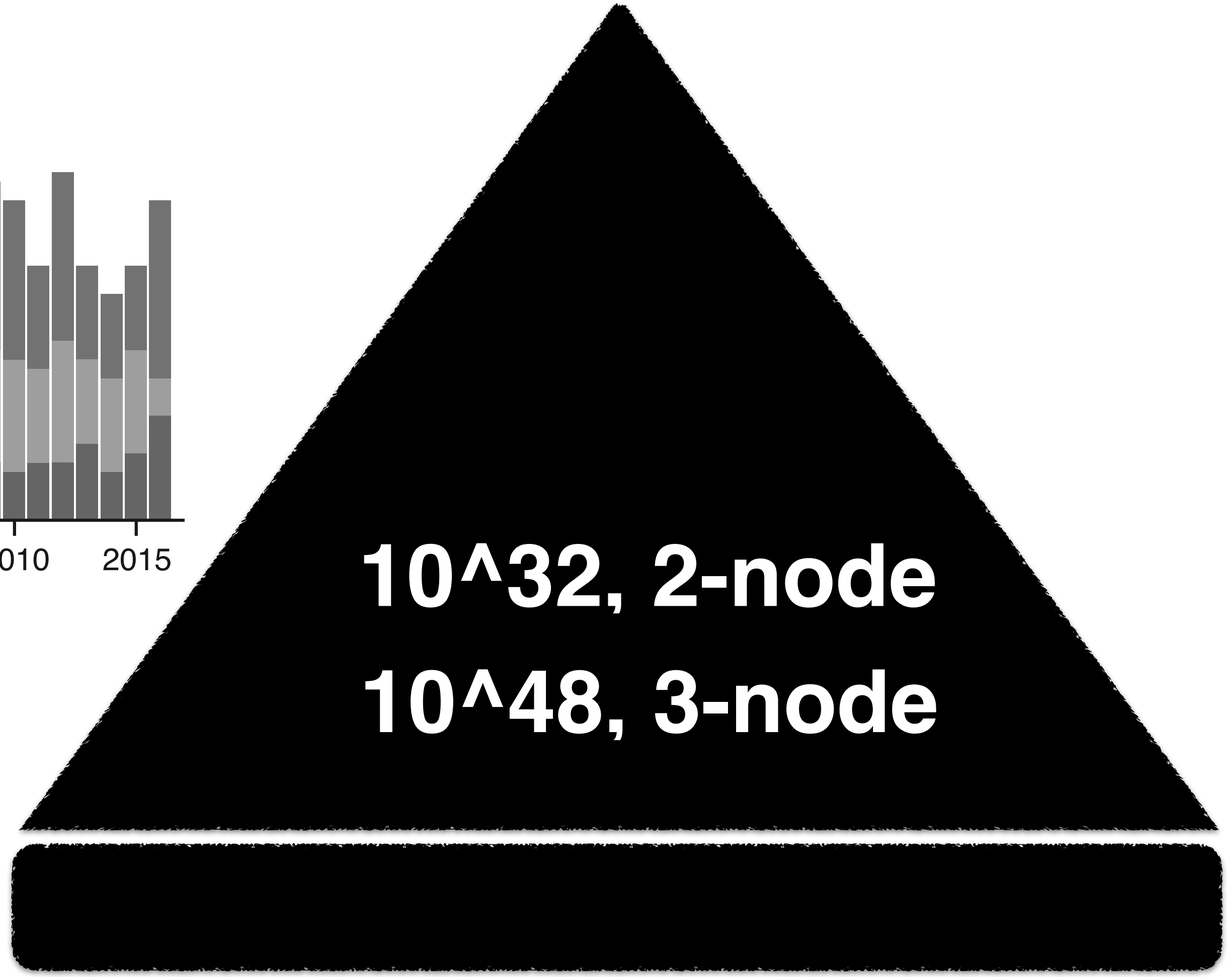


POSSIBLE DATA STRUCTURES

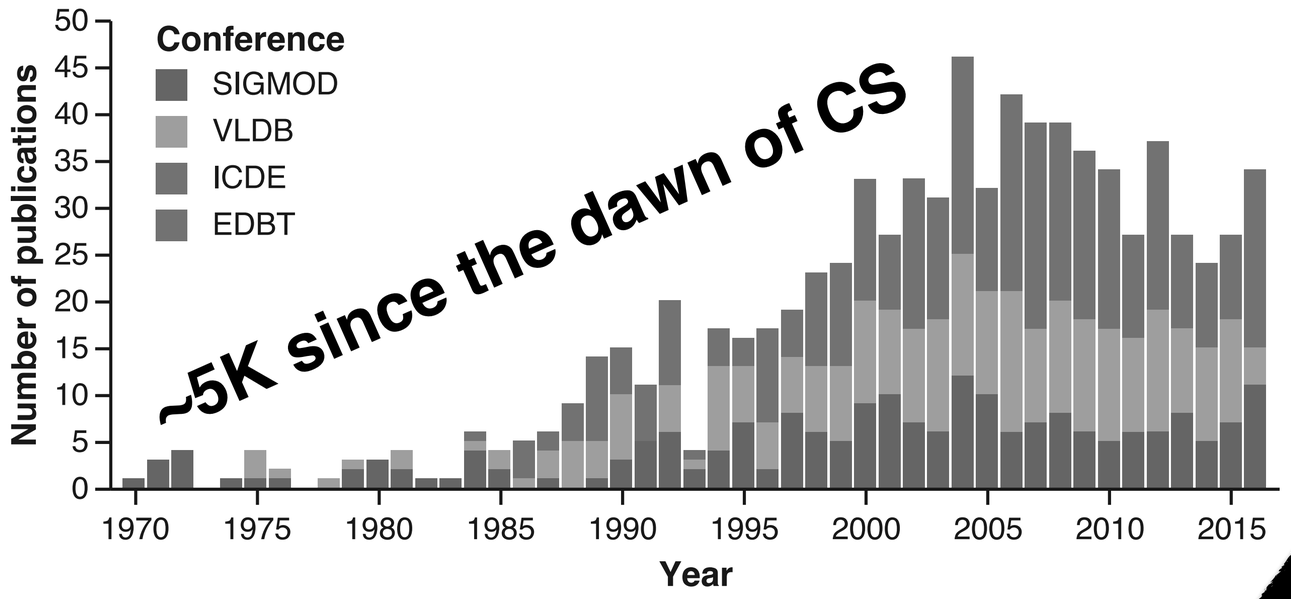


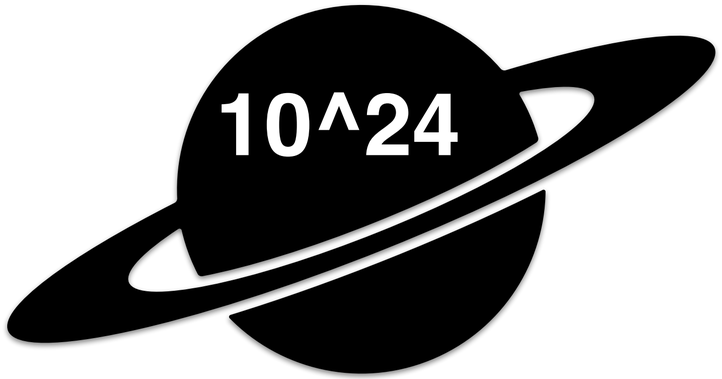


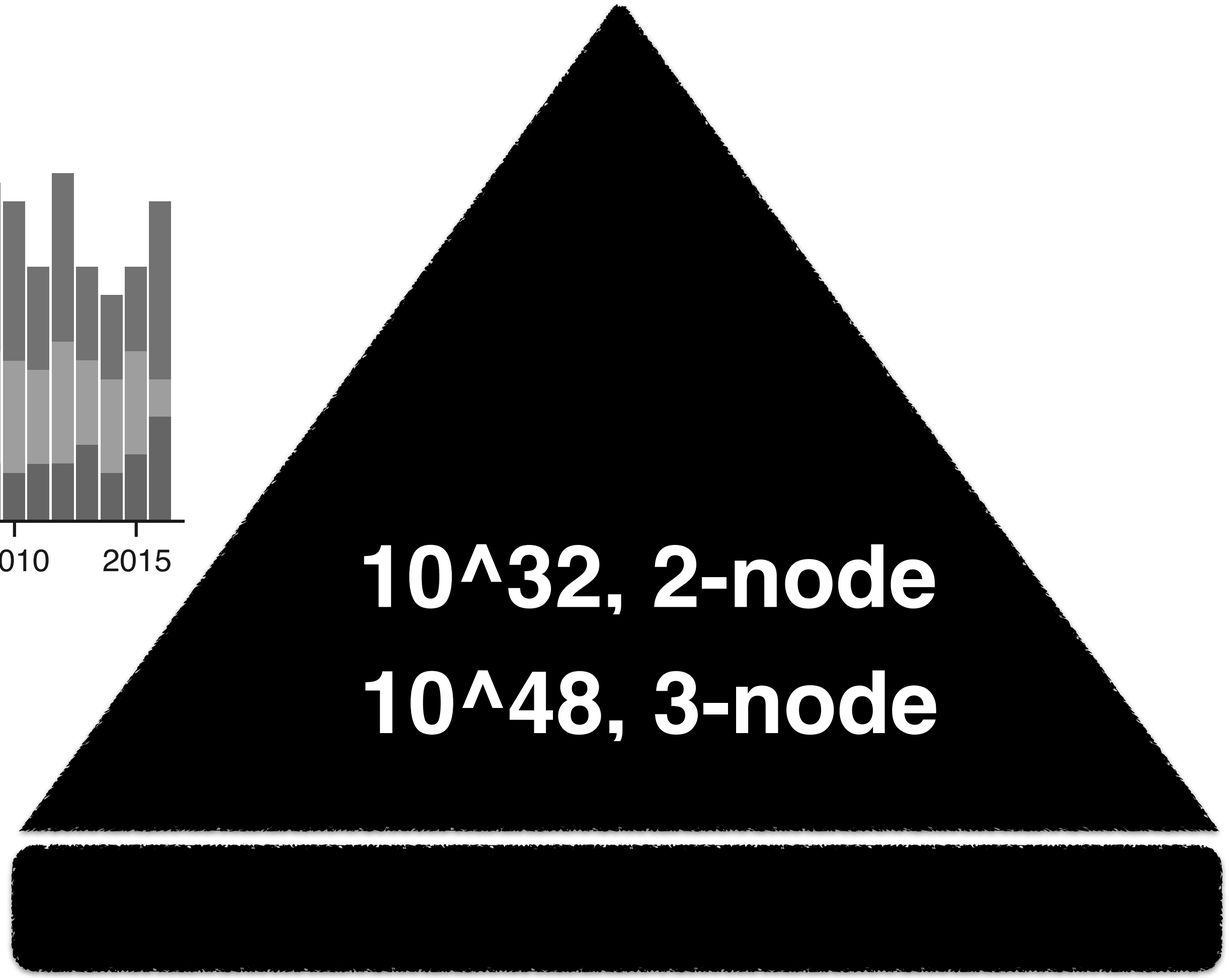
STARS IN THE SKY



POSSIBLE DATA STRUCTURES







10^{48}-5x10^{3} = 10^{48} **zero progress?**

CS 265

Stratos Idreos

BIG DATA SYSTEMS

NoSQL | Neural Networks | SQL | Graph | Data Science