CS 265

Stratos Idreos

BIG DATA SYSTEMS

NoSQL | Neural Networks | Image AI | LLMs | Data Science

Today:
Go quickly over logistics again

Intro to self-designing systems concept

Very high-level intro into NoSQL Big Data Systems (key-value stores)

algorithm/system design = not just computation

data movement is the key

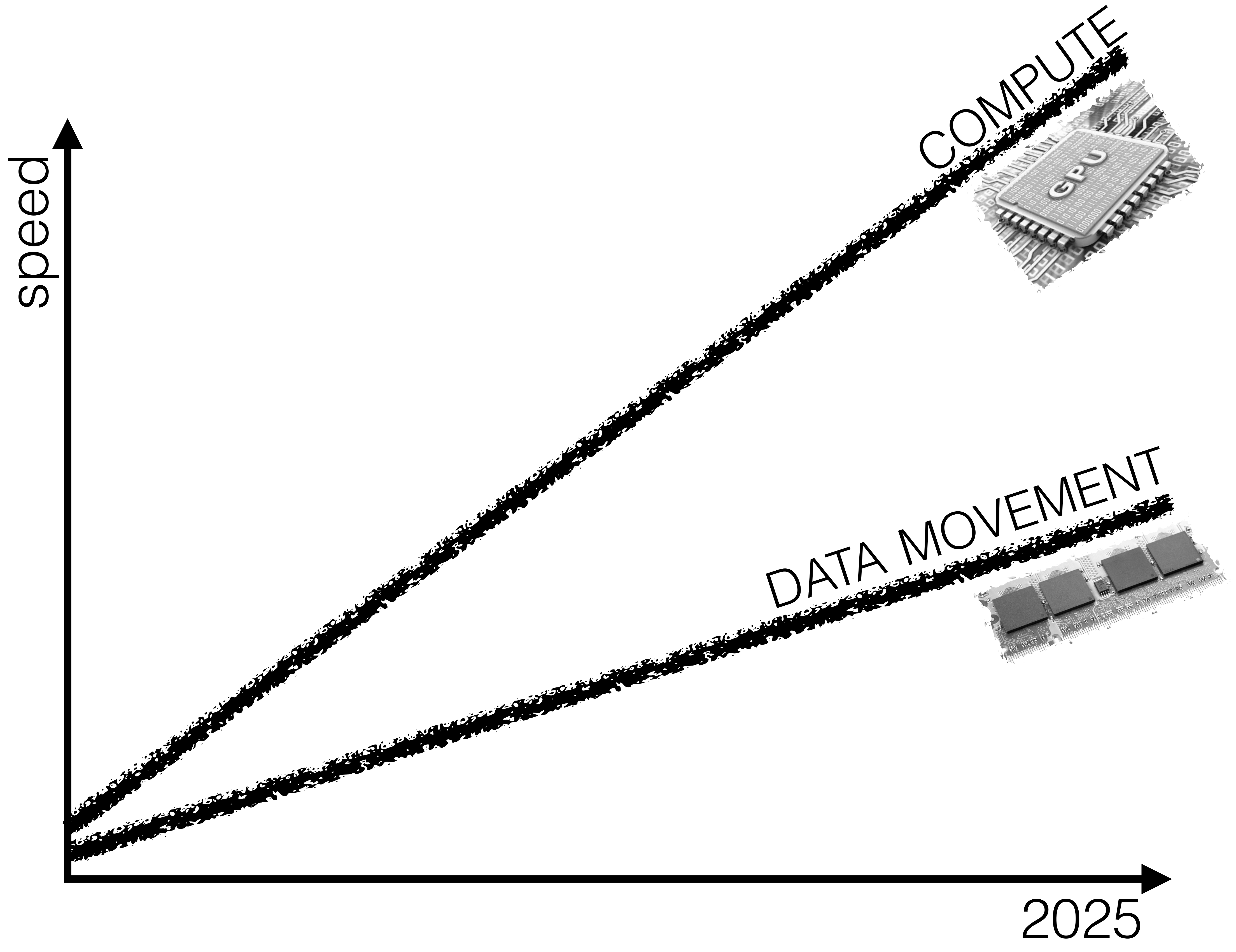
50-80% of end-to-end time is due to storage-related decisions

algorithm/system design = not just computation

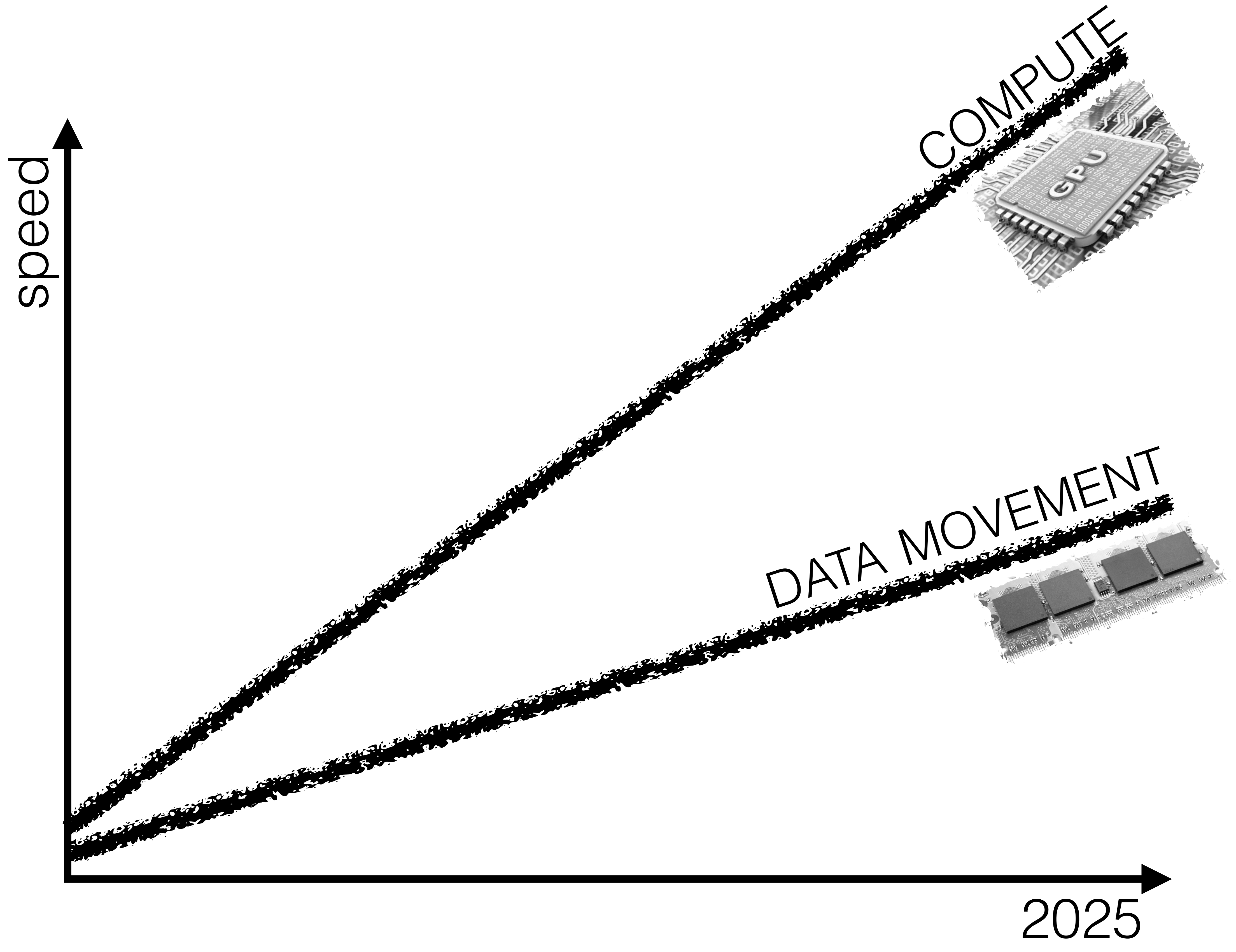
data movement is the key

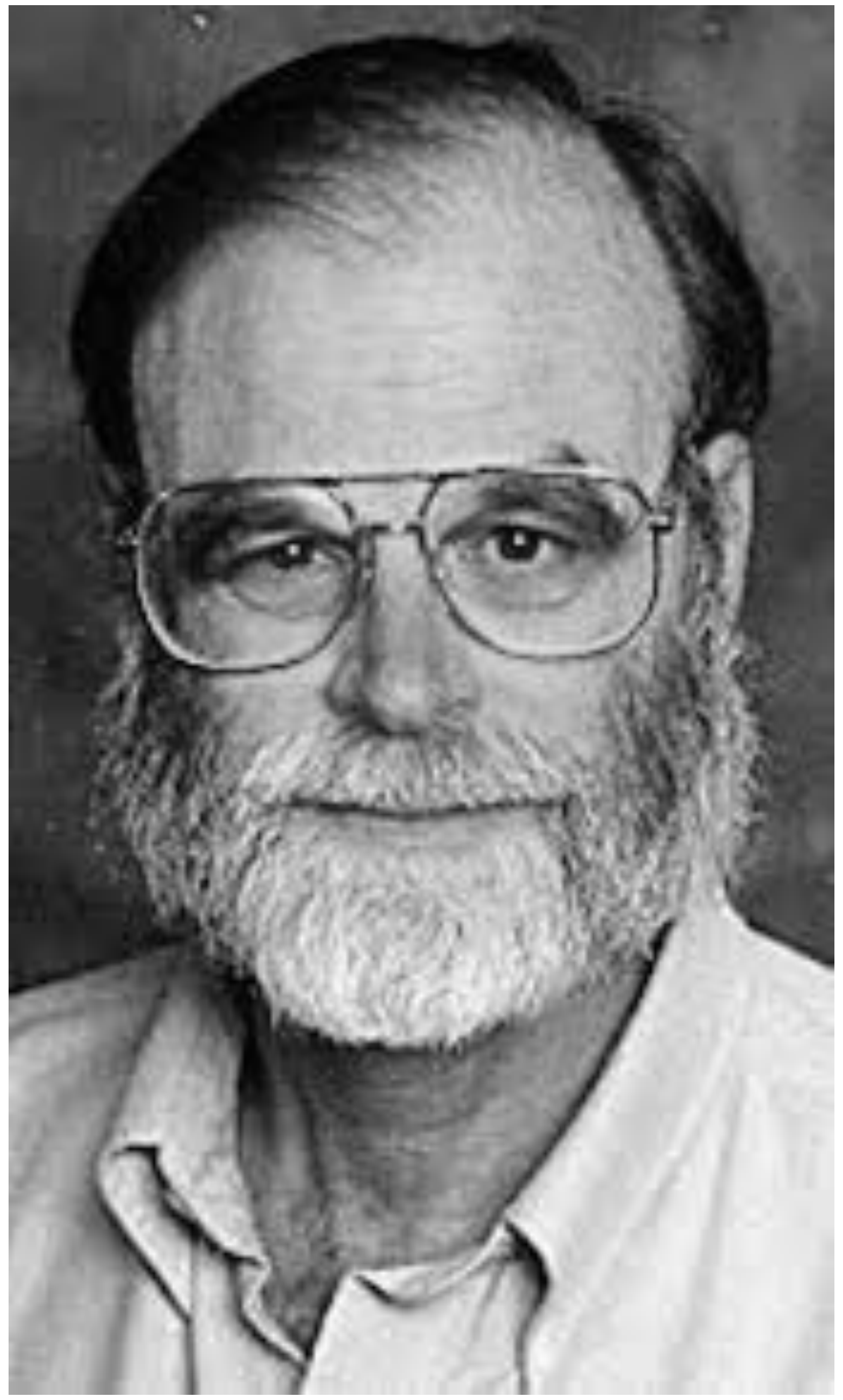
50-80% of end-to-end time is due to storage-related decisions

cloud cost



DATA STRUCTURES DEFINE PERFORMANCE





register = this room

caches = this city

memory = nearby city

disk = Pluto

Jim Gray, Turing Award 1998

What is a data system?

A data system is an end-to-end software system that:
manages storage, data movement, and provides access to data

What is a data system?

A data system is an end-to-end software system that:
manages storage, data movement, and provides access to data

A system is a complex set of components

interacting in harmony depending on the context

exposing as little as possible complexity to users

declarative interface

ask ''what'' you want

data* system

the system decides
"how" to best store
and access data

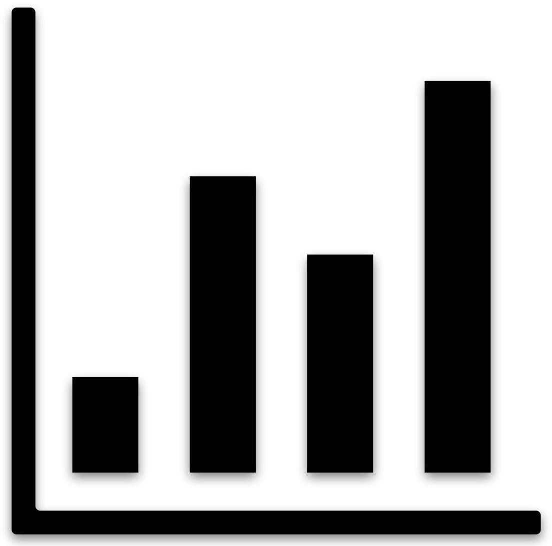
How do I make my **data system** run x times as fast?  (sql,nosql,bigdata, …)

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

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

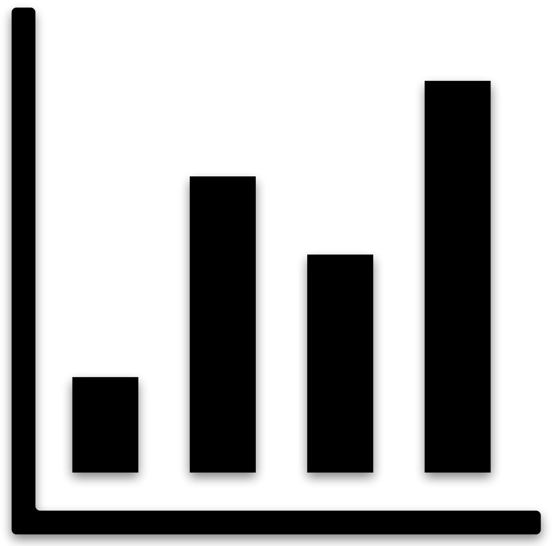
How do I make my **data system** run x times as fast?  (sql,nosql,bigdata, …)

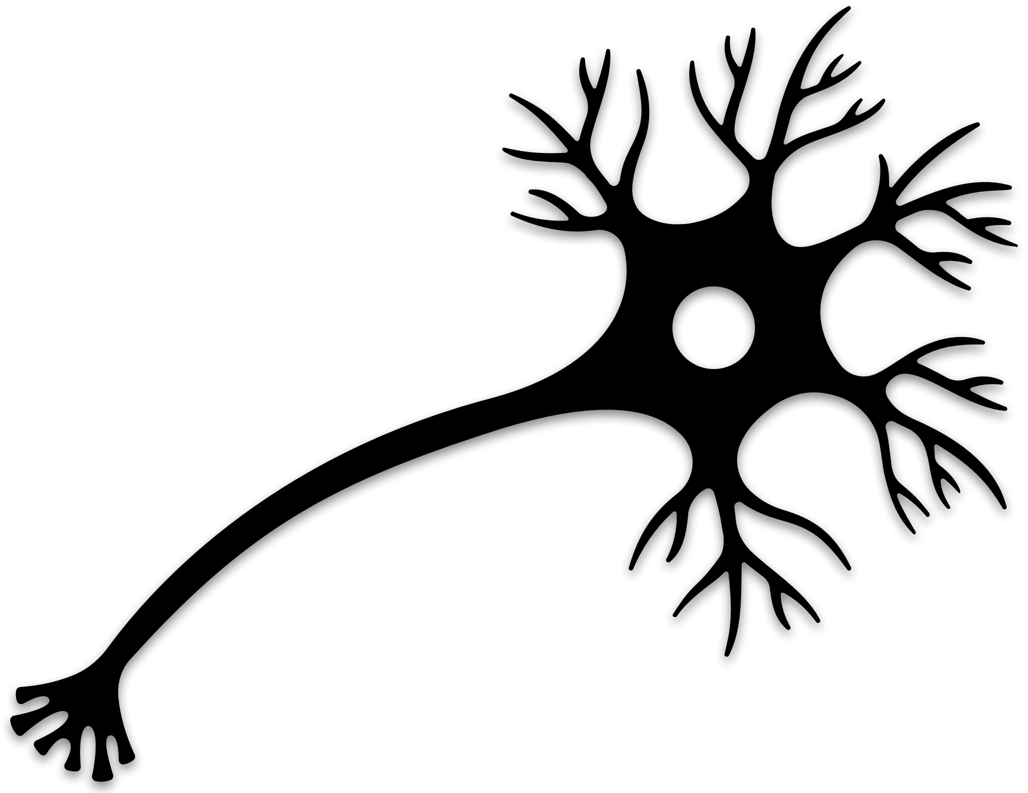
How do I minimize my **bill** in the **cloud**?

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

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

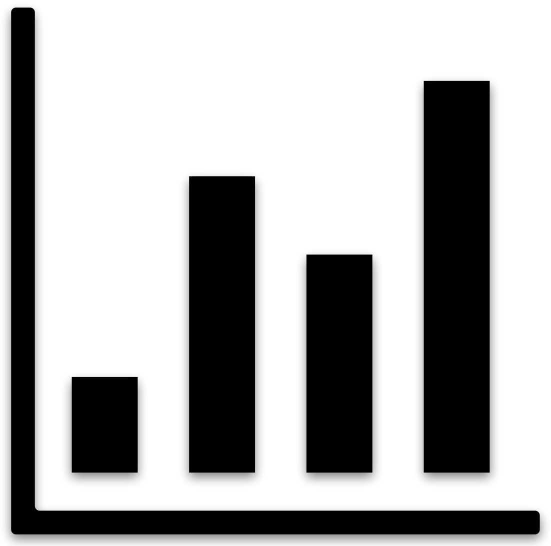


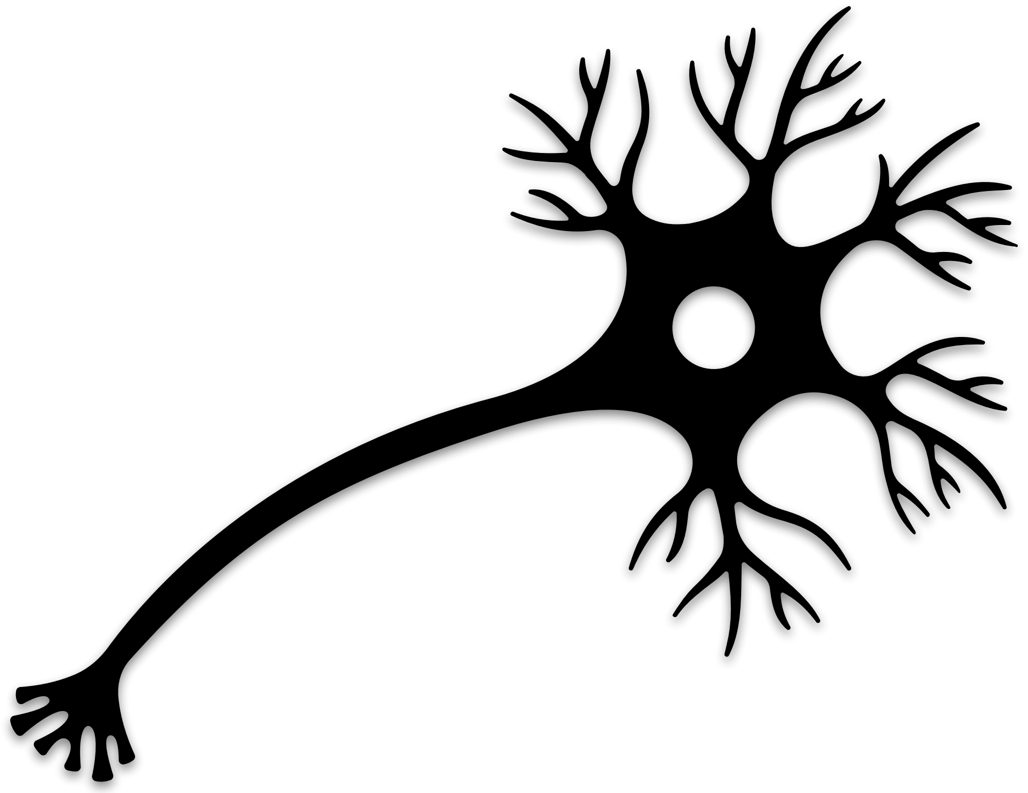
How do I train my **neural network/LLM** x times faster?

DASlab @ Harvard SEAS

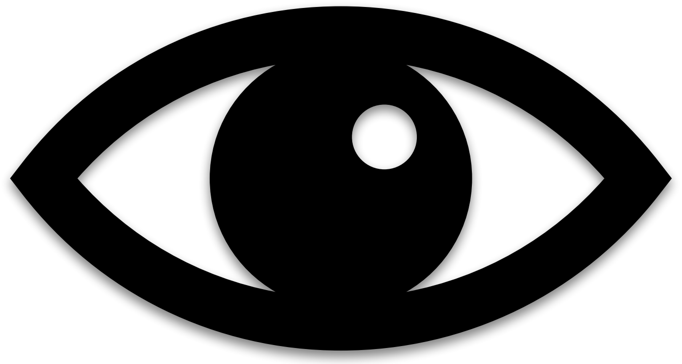
How do I make my **data system** run x times as fast?  (sql,nosql,bigdata, …)

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

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



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

How can I do 10x **Image AI inference**? 

DASlab @ Harvard SEAS

Is there maybe a perfect system? Nope…

learning outcome
Fundamentals of storage

data structures, SQL, NoSQL, Neural Networks, Data Science, Images, LLMs

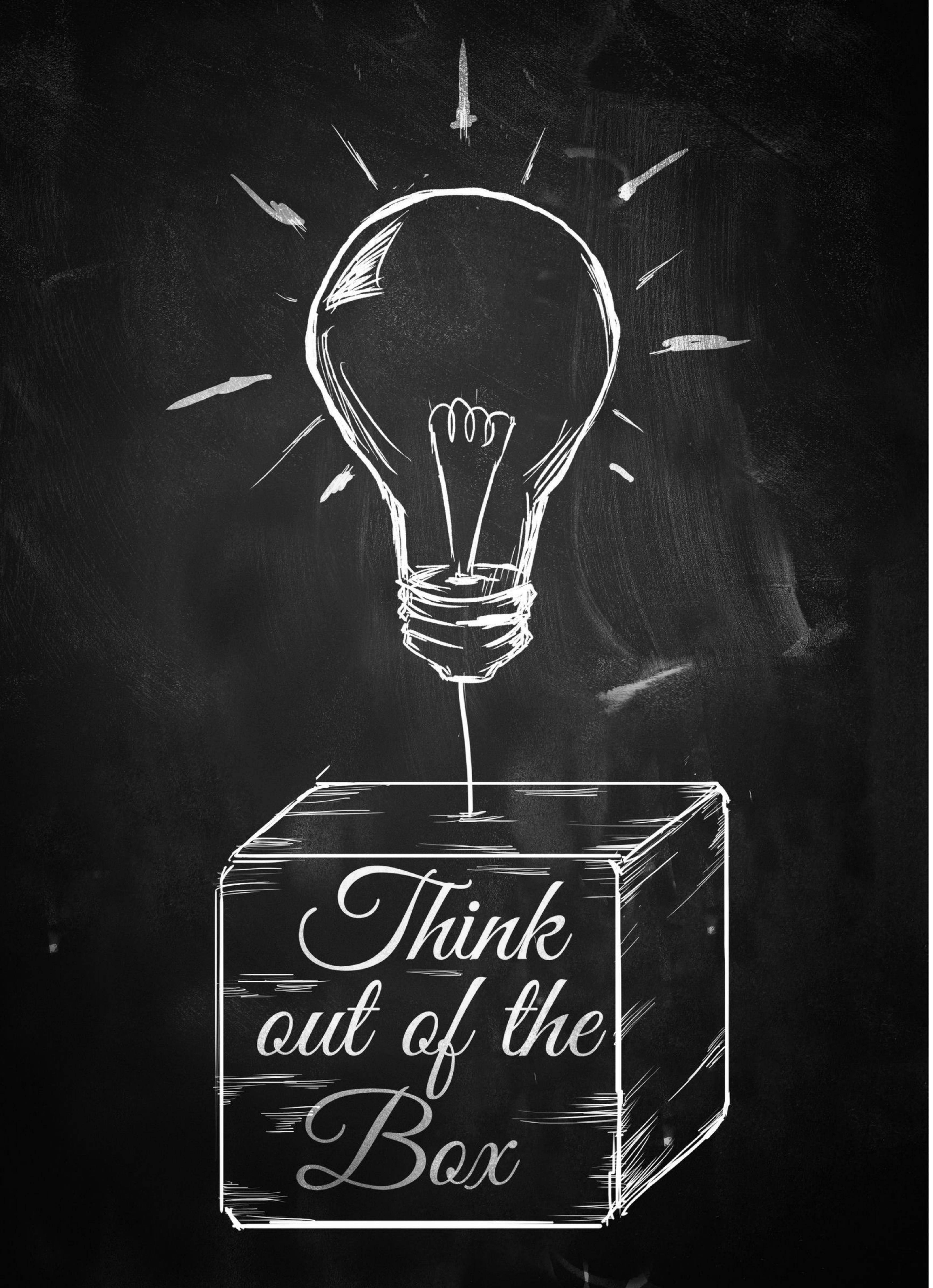
learning outcome

Fundamentals of storage

data structures, SQL, NoSQL, Neural Networks, Data Science, Images, LLMs

Self-designing systems

Automated system design: cloud cost, hardware, data & app requirements



first 4-5 weeks: Stratos/Sanket/Utku
Basic background
Self-designing systems
Neural network systems
Image AI systems
Research thinking

afterwards:
Students present research papers
One paper per class (ML systems)
In-class research/systems discussion
Research reviews
Research/systems projects

Recent Research Papers

Each student:

2 reviews per week/1 presentation

review and slides should focus on

what is the problem
why is it important
why is it hard
why existing solutions do not work
what is the core intuition for the solution
solution step by step
does the paper prove its claims
exact setup of analysis/experiments
are there any gaps in the logic/proof
possible next steps

* follow a few citations to gain more background

learn to judge constructively

learn to present

learn to prepare slides

Each student:

2 reviews per week/1 presentation

review and slides should focus on

what is the problem
why is it important
why is it hard
why existing solutions do not work
what is the core intuition for the solution
solution step by step
does the paper prove its claims
exact setup of analysis/experiments
are there any gaps in the logic/proof
possible next steps

* follow a few citations to gain more background

semester project: due in the end of semester + a midway check in (mid March, 10%)

systems project

research project

semester project: due in the end of semester + a midway check in (mid March, 10%)

systems project

individual project

NoSQL, in c/c++

MLsys, in pytorch

research project

semester project: due in the end of semester + a midway check in (mid March, 10%)

systems project

individual project

NoSQL, in c/c++

MLsys, in pytorch

research project

groups of max three

Adaptivity/Performance
Across all subject areas

semester project: due in the end of semester + a midway check in (mid March, 10%)

systems project

individual project

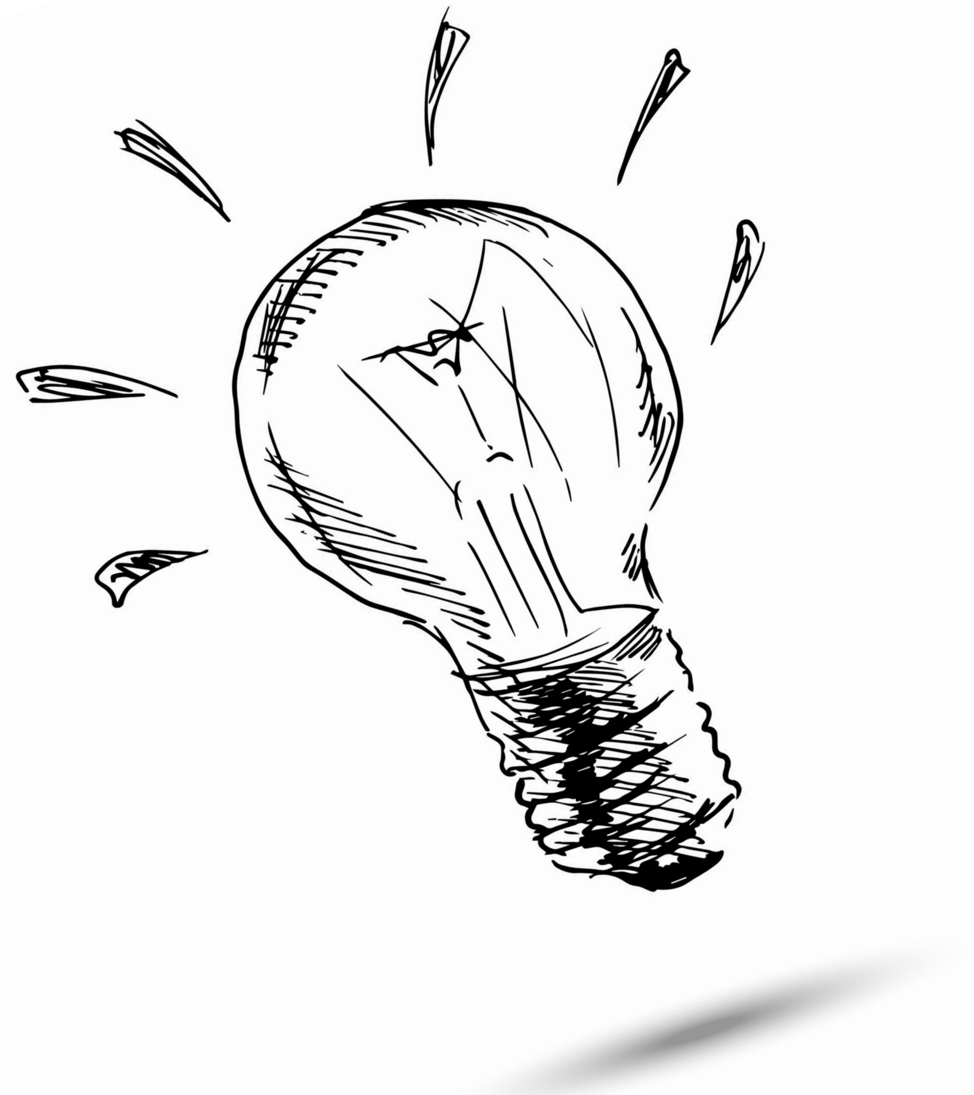
NoSQL, in c/c++

MLsys, in pytorch

research project

groups of max three

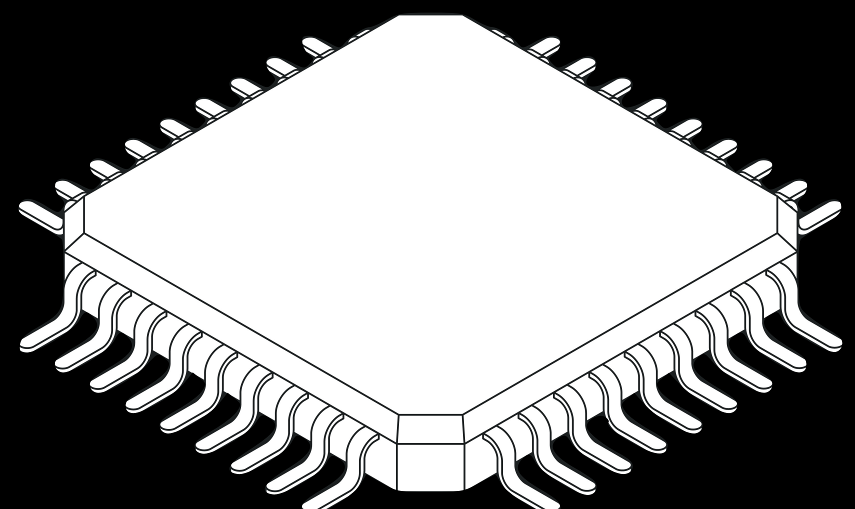
Adaptivity/Performance
Across all subject areas

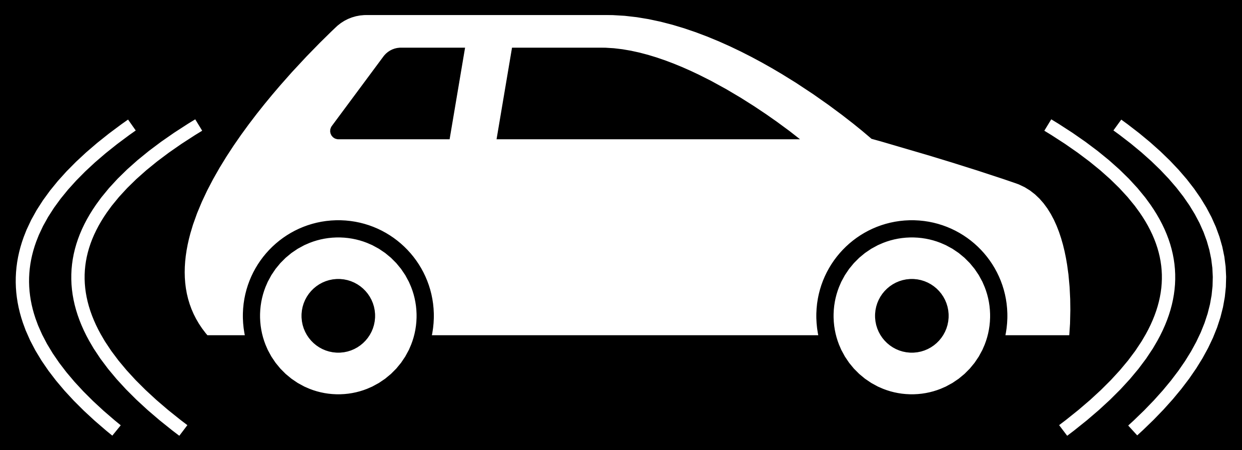


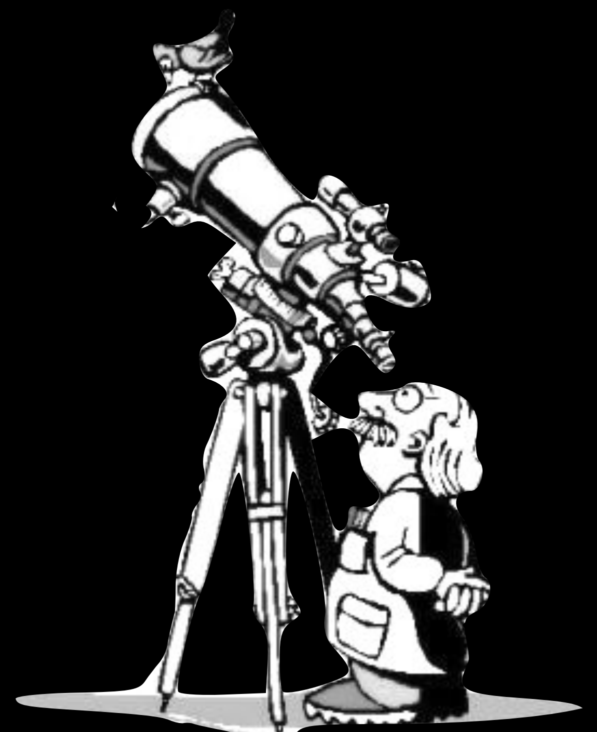
Questions on logistics?

Self-designing Systems

The problem: as the big data/AI world keeps changing…

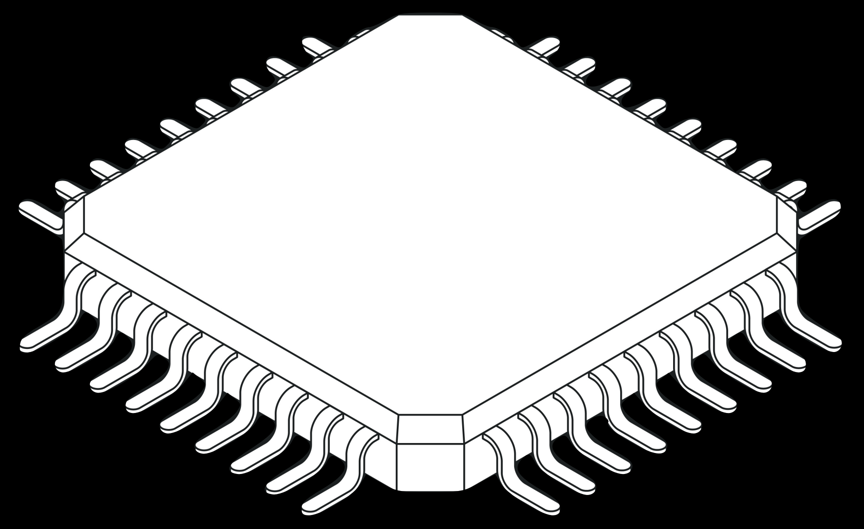


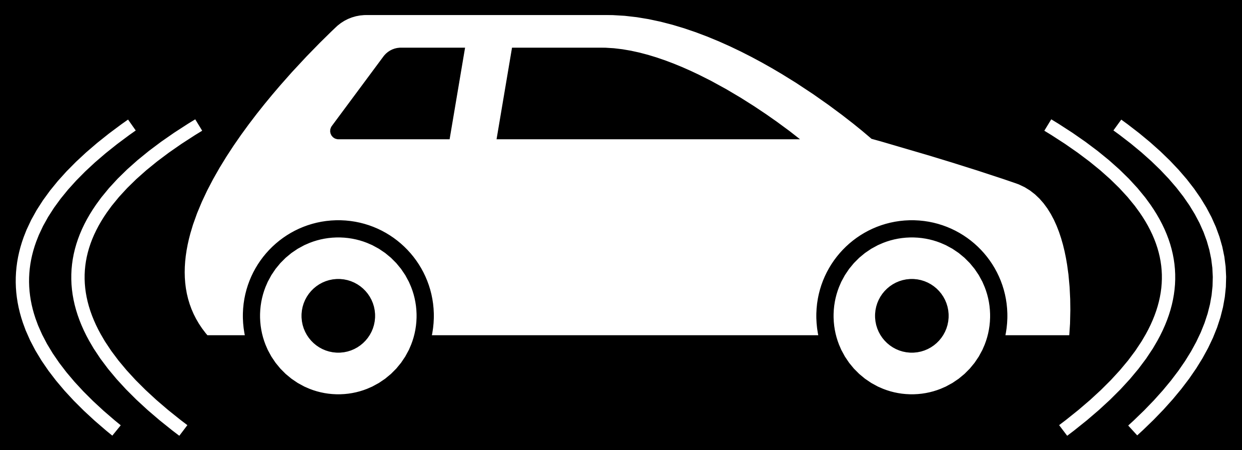


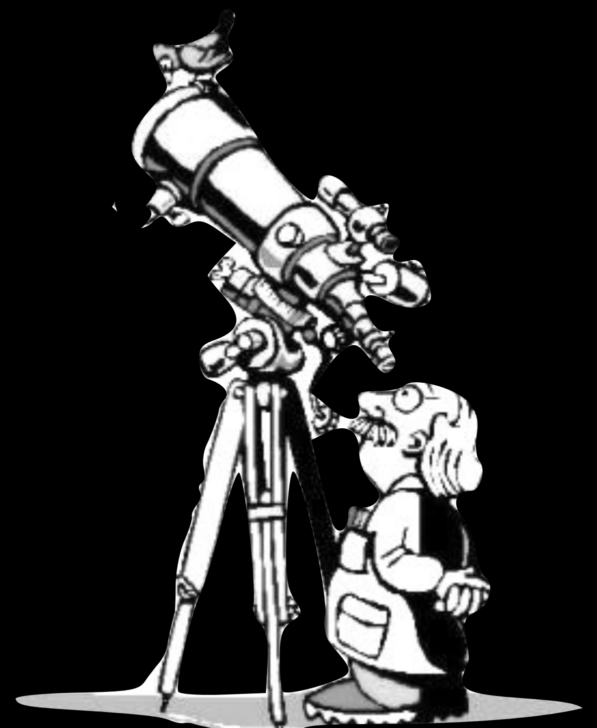


The problem: as the big data/AI world keeps changing…

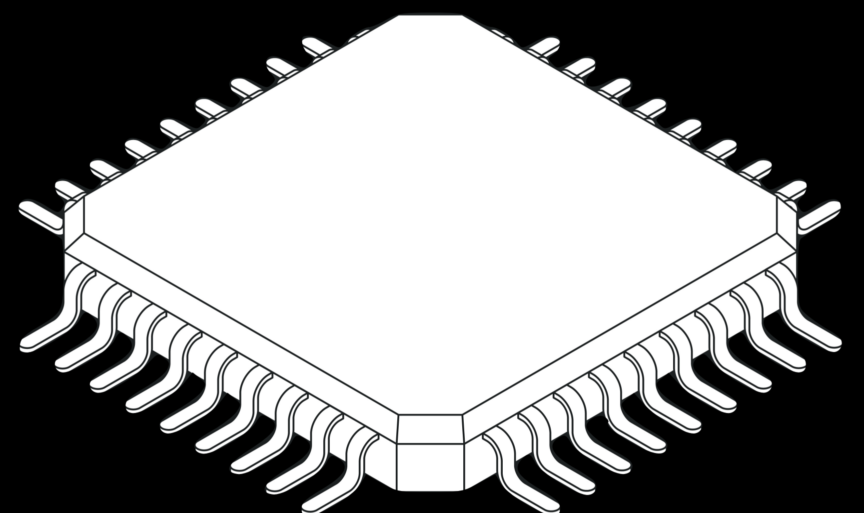
there is a continuous need for new data systems
but it is **extremely hard to design & build new systems**

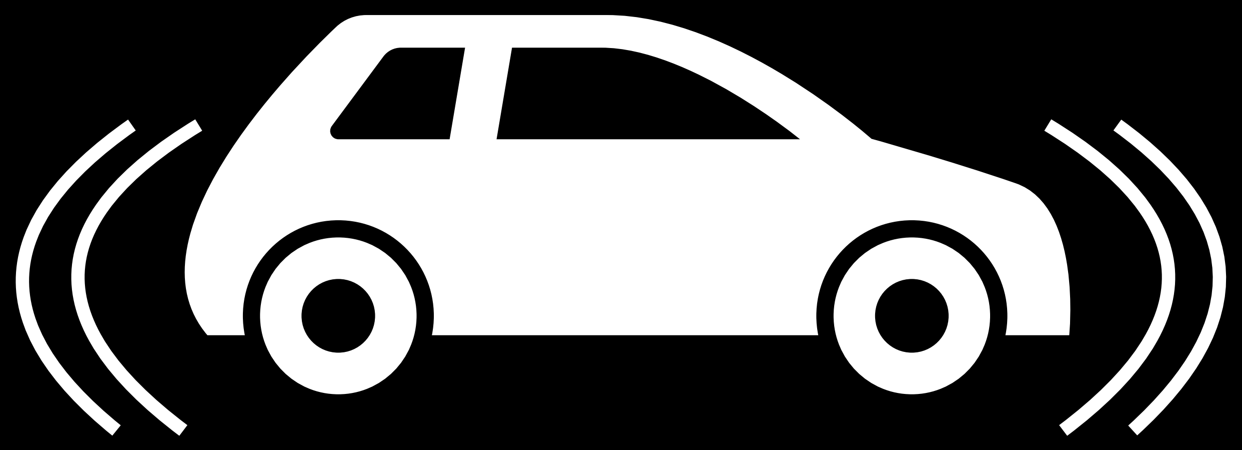


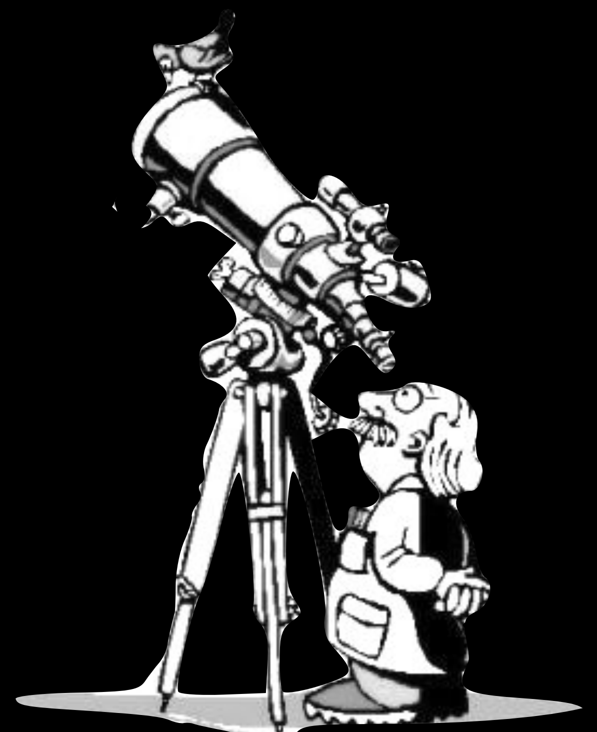




How do we design a data system that is **X times faster for a workload W**?

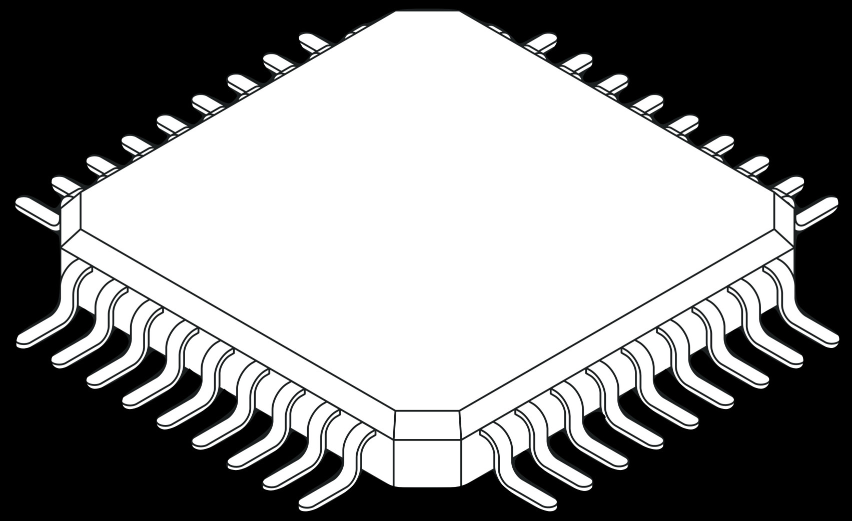


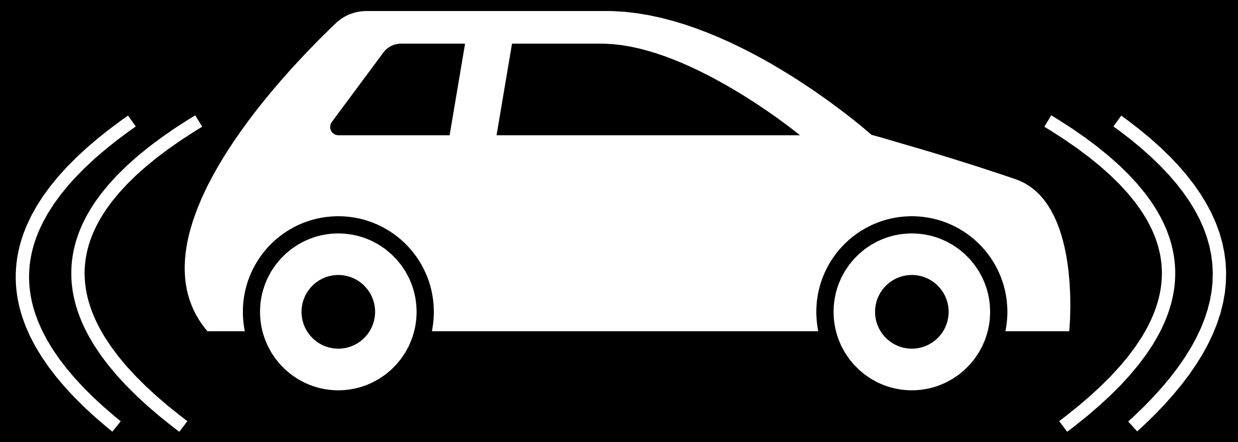


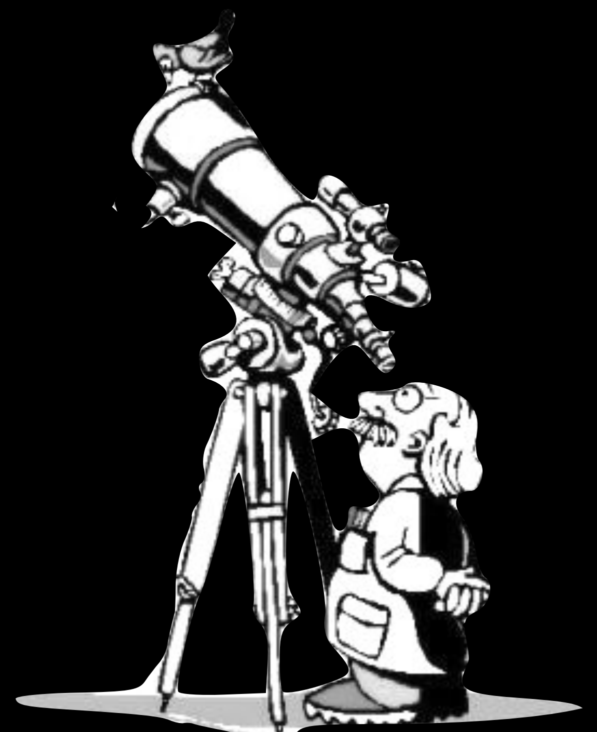


How do we design a data system that is **X times faster for a workload W**? 

 How do we design a data system that allows for control of **cloud cost**?



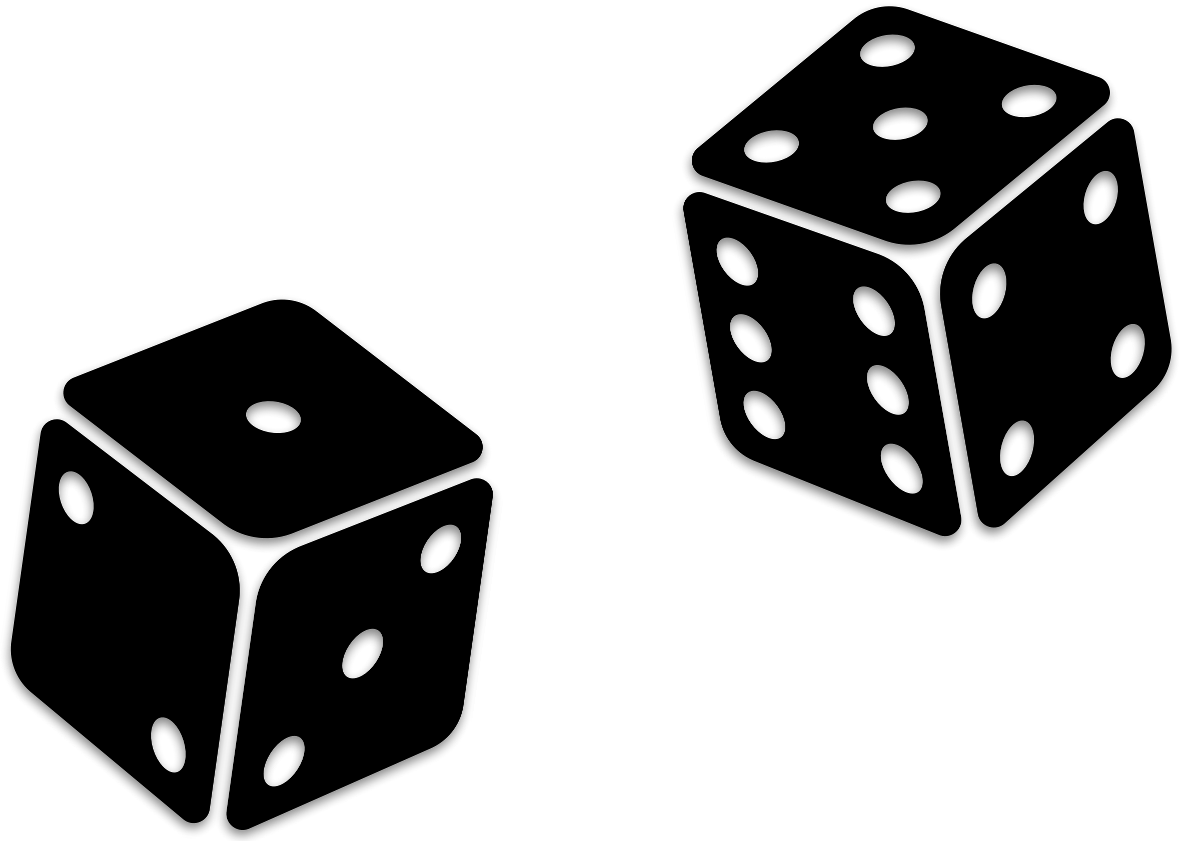




How do we design a data system that is **X times faster for a workload W**? 

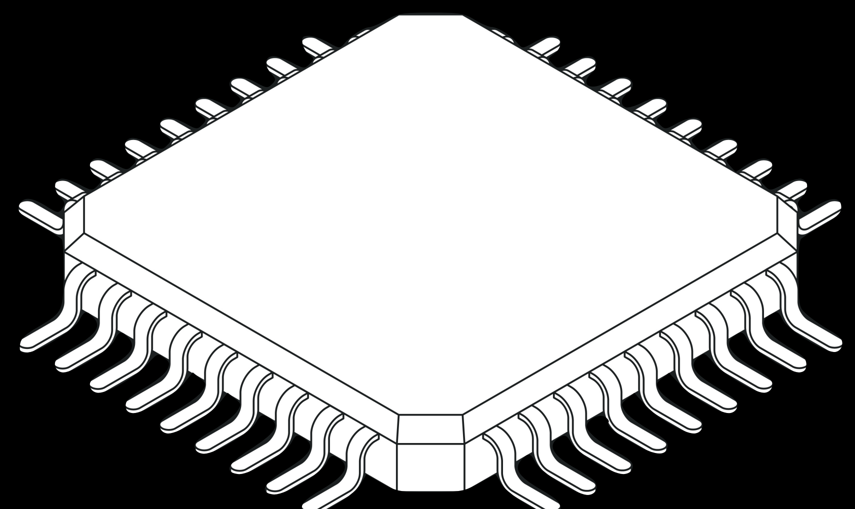
 How do we design a data system that allows for control of **cloud cost**?

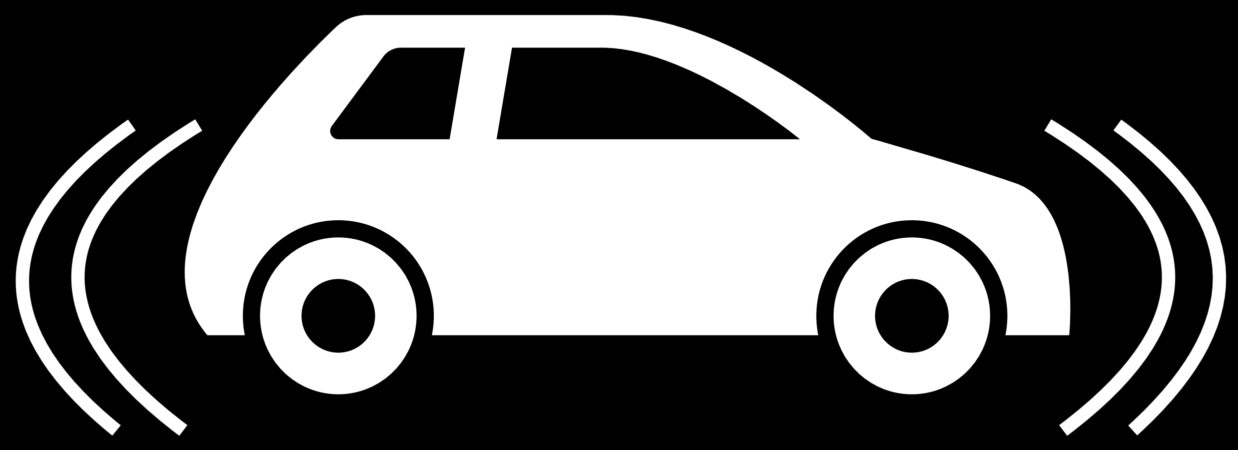
What happens if we introduce **new application feature** Y?

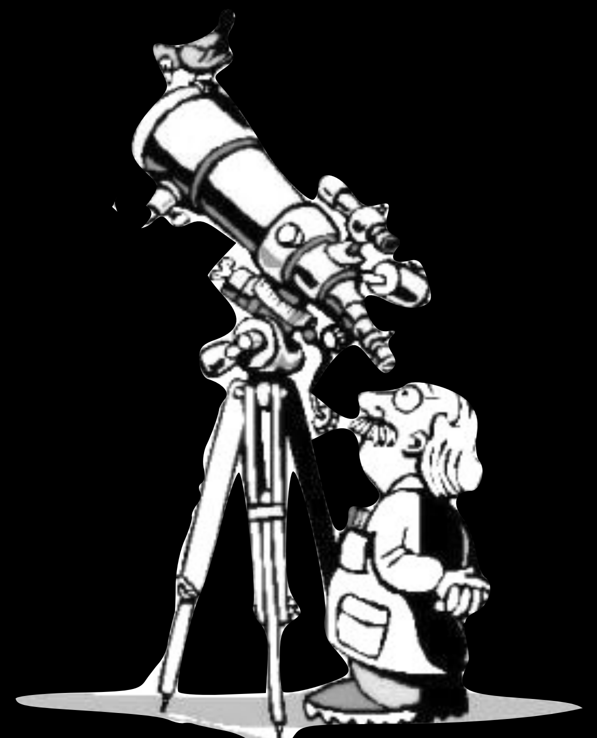


Should we **upgrade** to new version Z?

What will **break** our system?





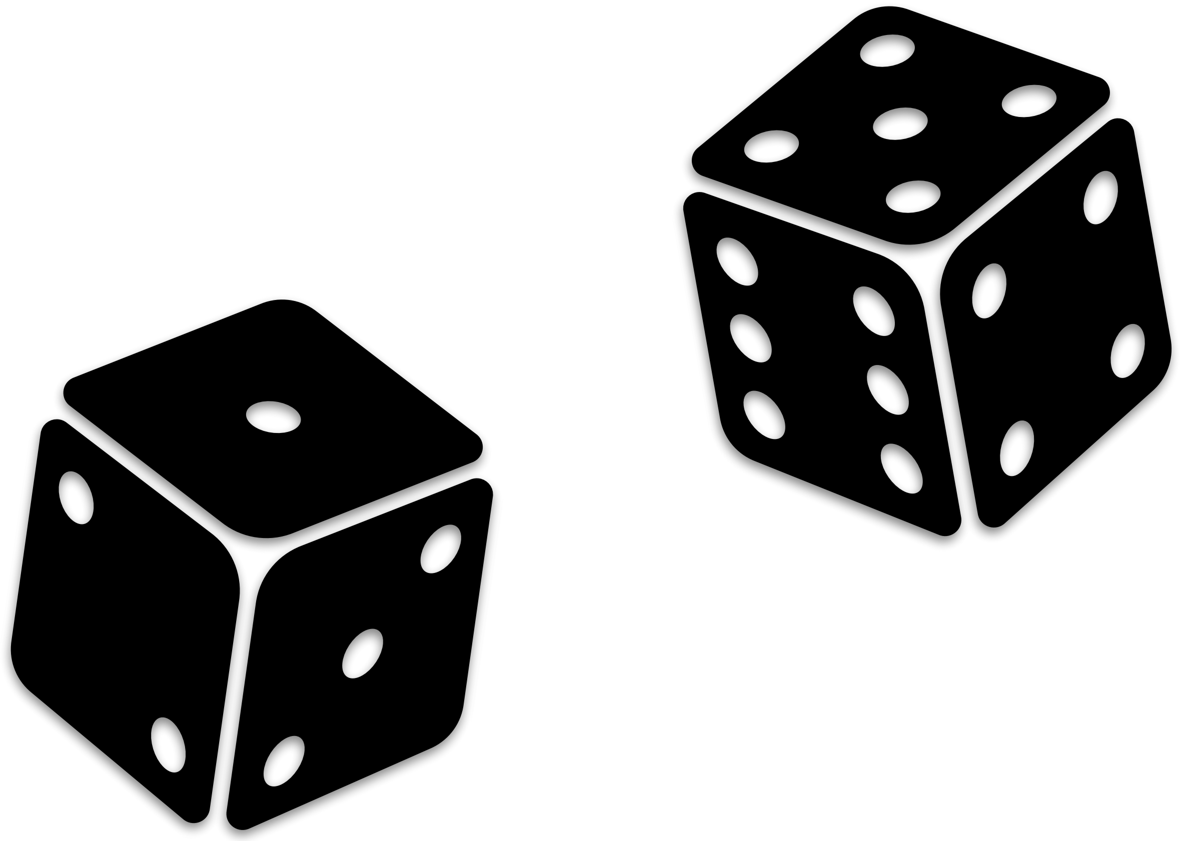


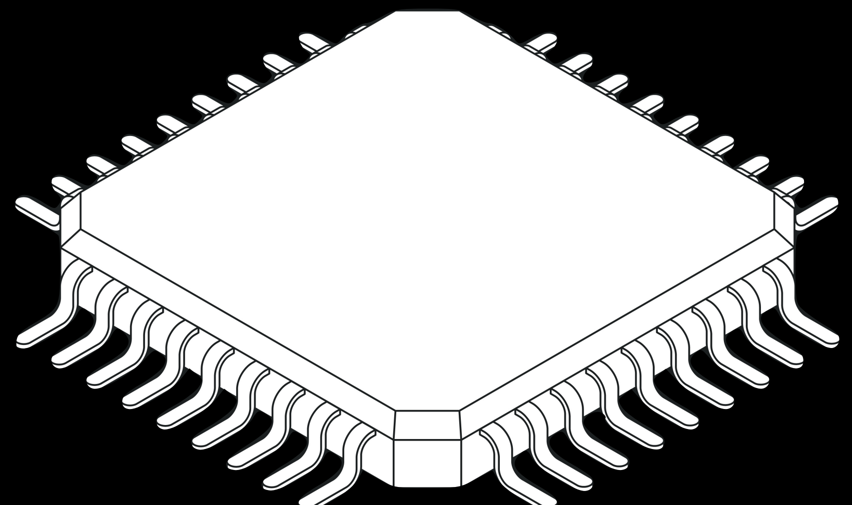
BOTTLENECK: SUB-OPTIMAL DATA SYSTEMS

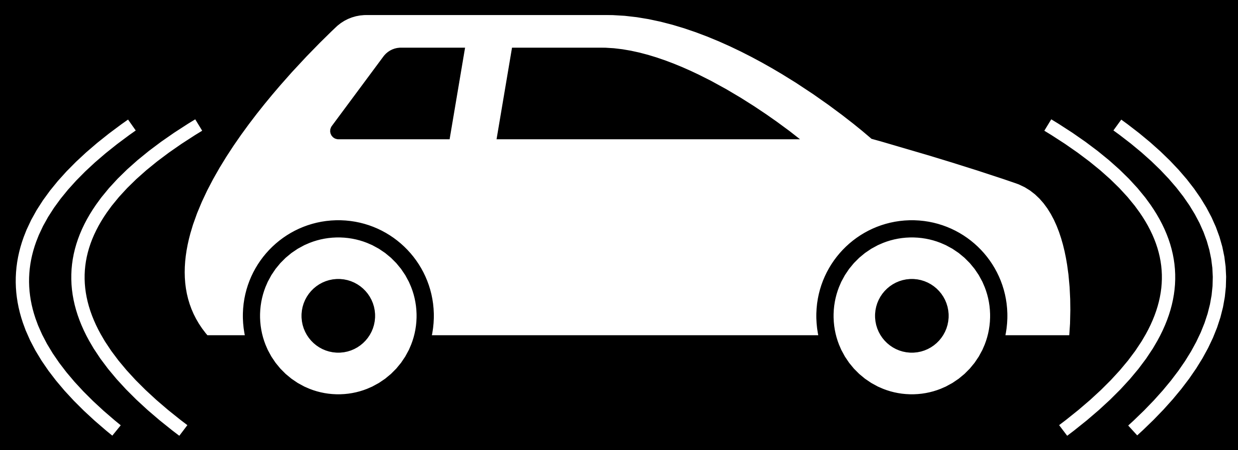
What happens if we introduce **new application feature** Y?

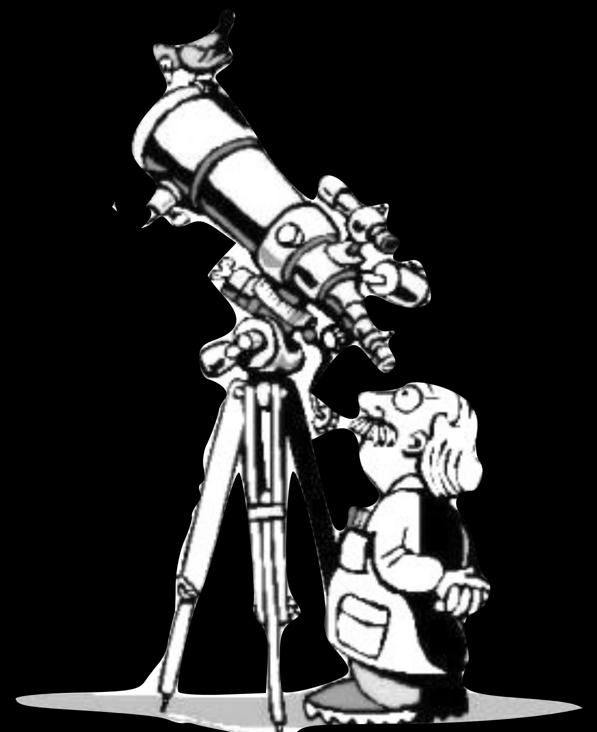
Should we **upgrade** to new version Z?

What will **break** our system?





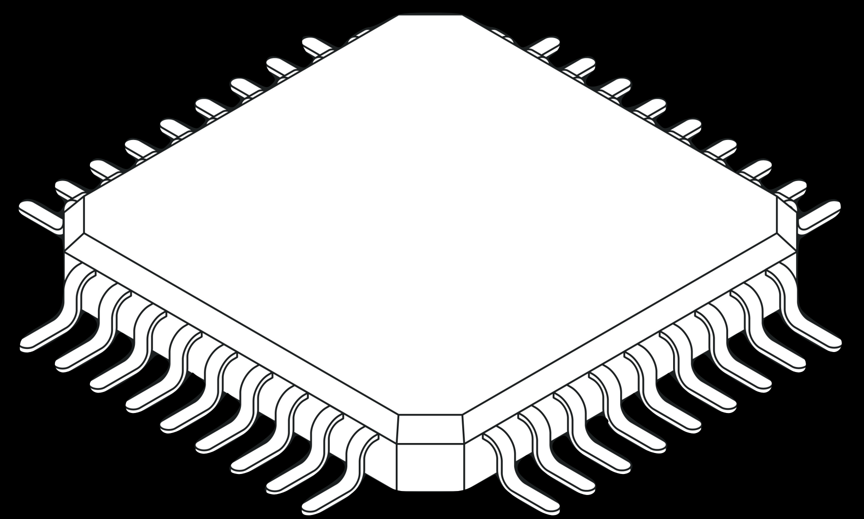


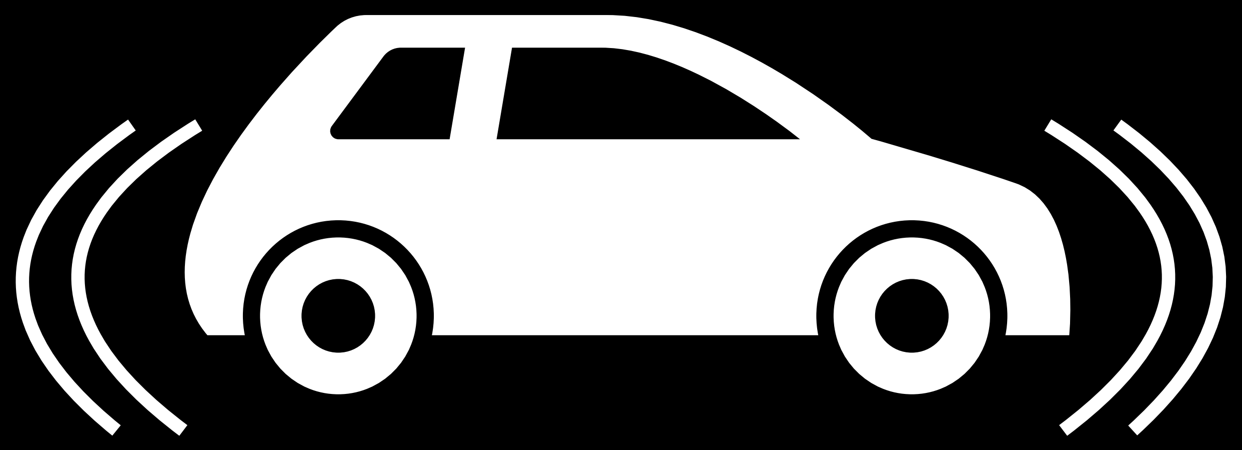


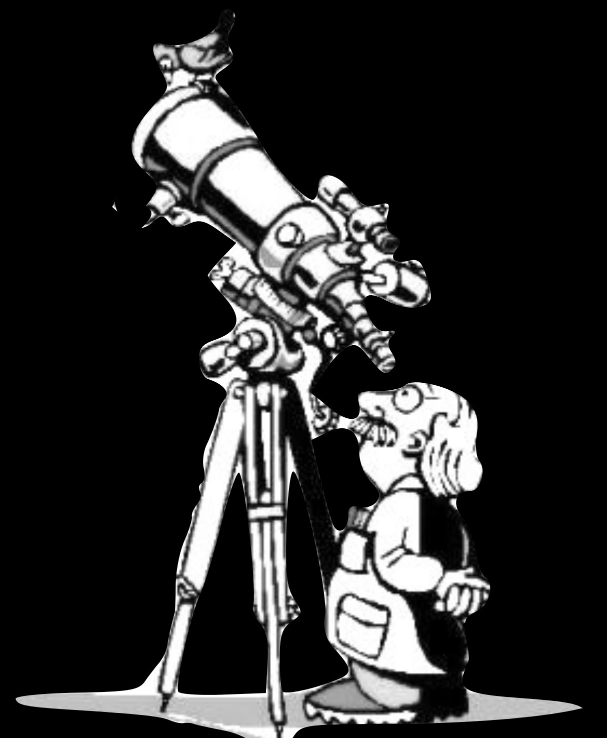
BOTTLENECK: SUB-OPTIMAL DATA SYSTEMS

huge cloud cost

environmental impact





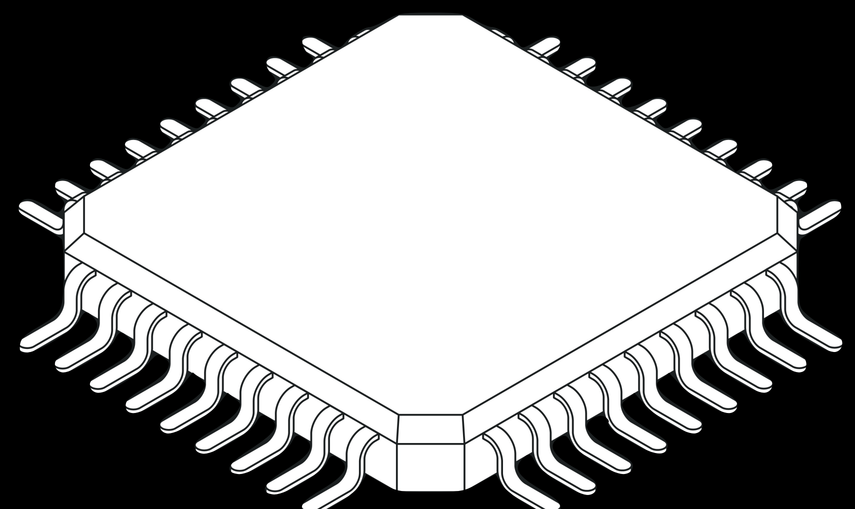


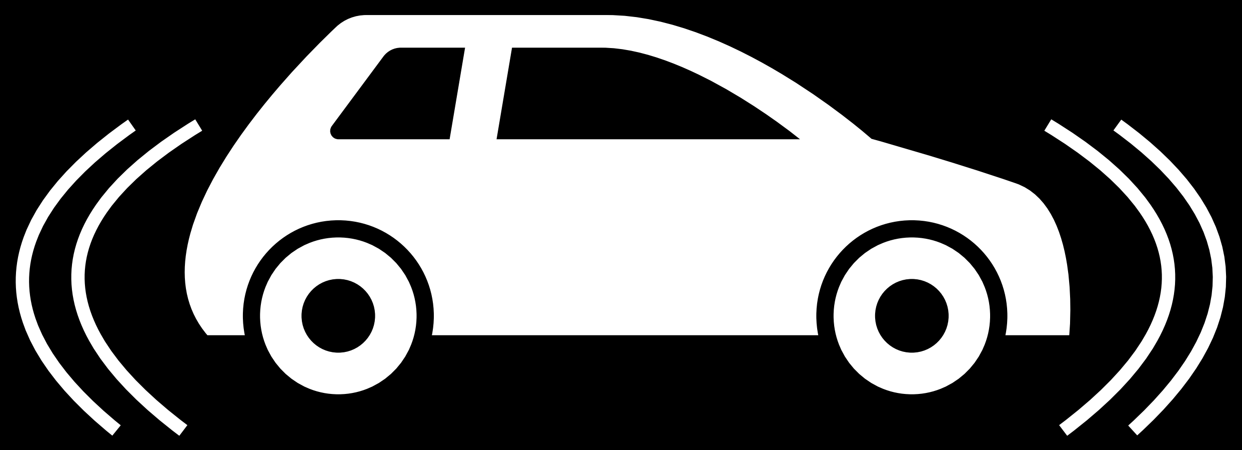
BOTTLENECK: SUB-OPTIMAL DATA SYSTEMS

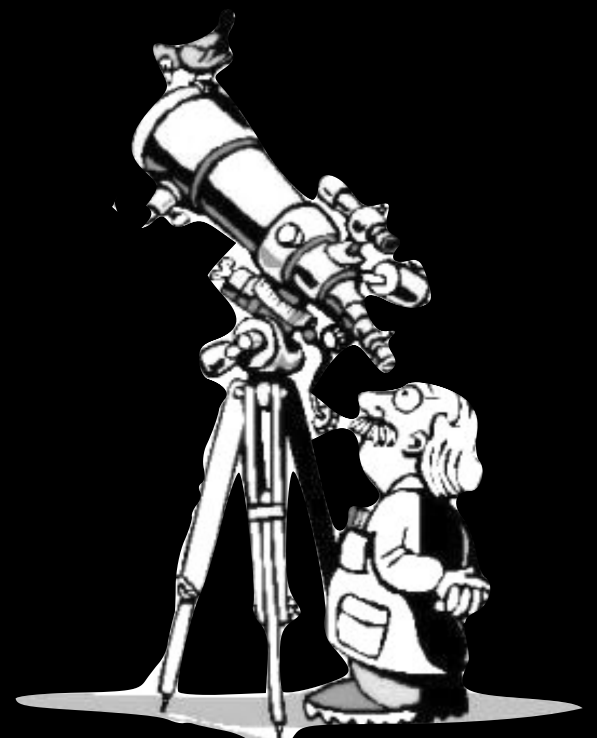
expensive transitions

huge cloud cost

environmental impact







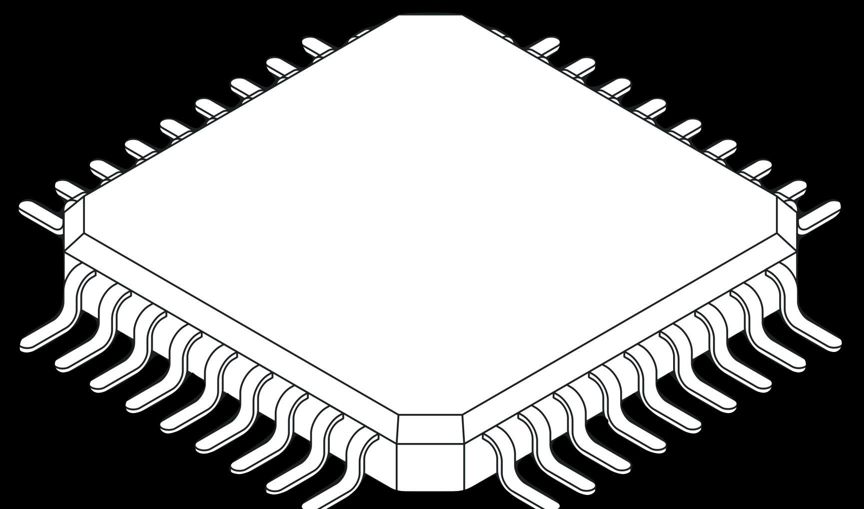
BOTTLENECK: SUB-OPTIMAL DATA SYSTEMS

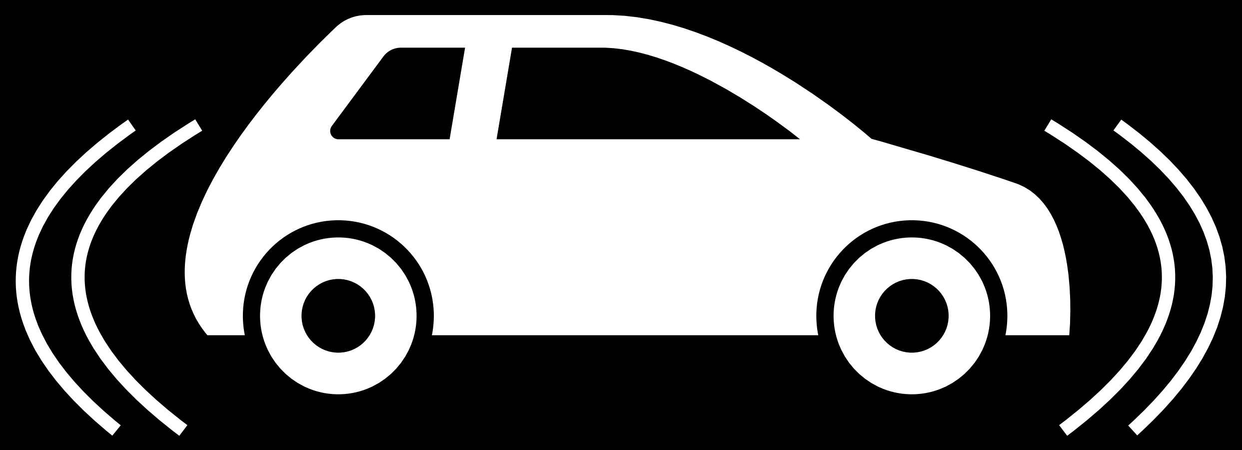
expensive transitions

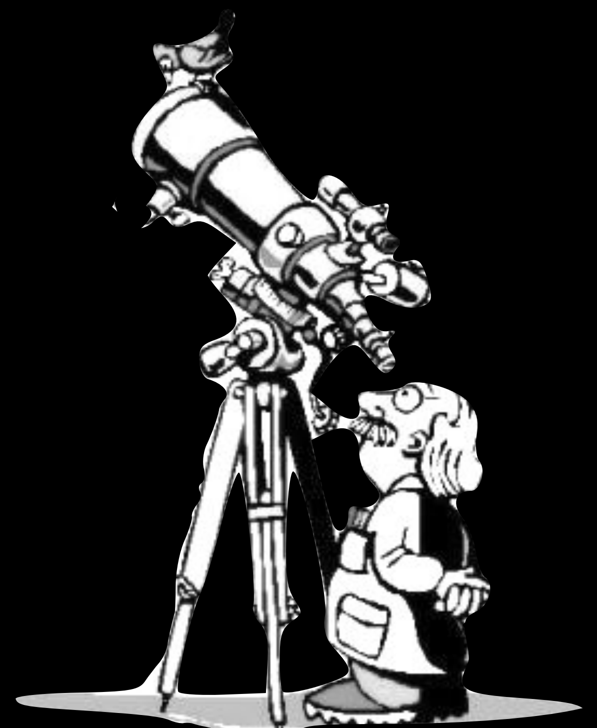
huge cloud cost

application feasibility

environmental impact





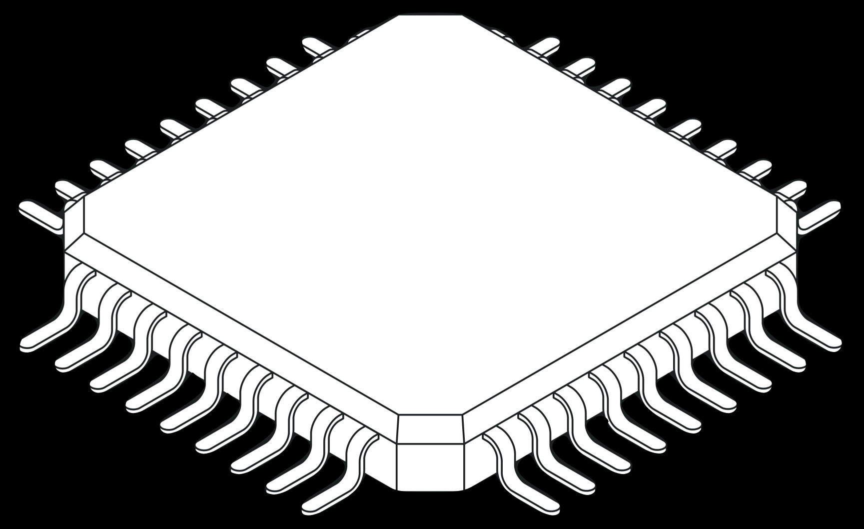


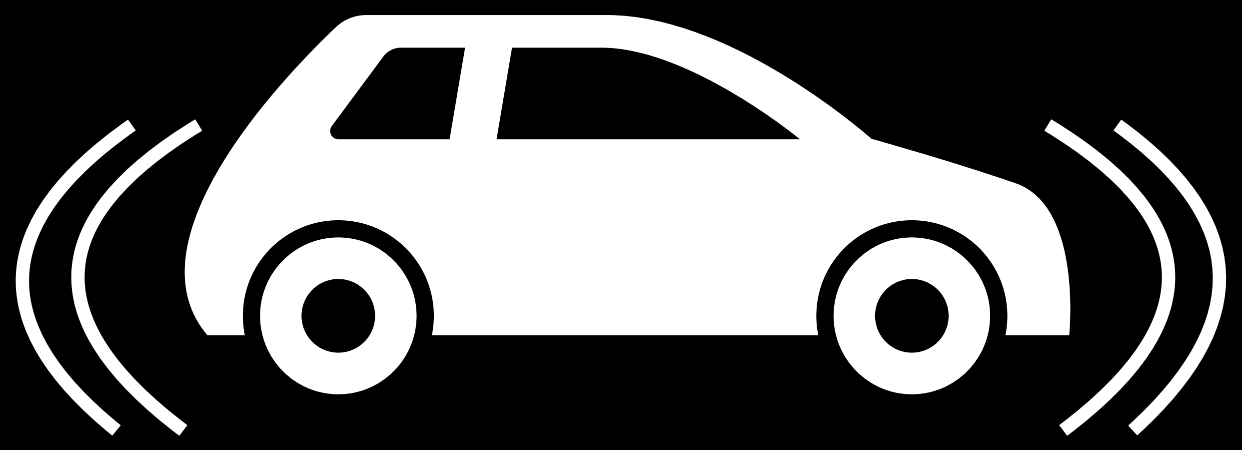
BOTTLENECK: SUB-OPTIMAL DATA SYSTEMS

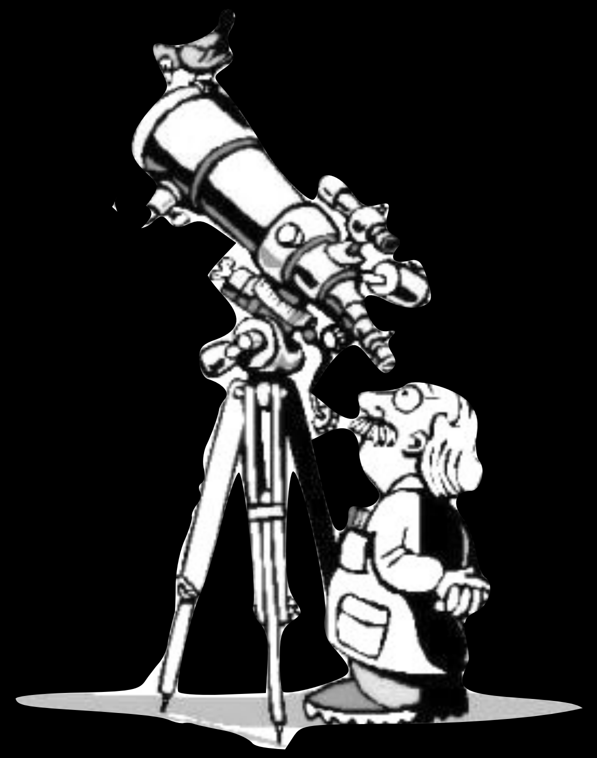
huge cloud cost expensive transitions application feasibility environmental impact

complexity

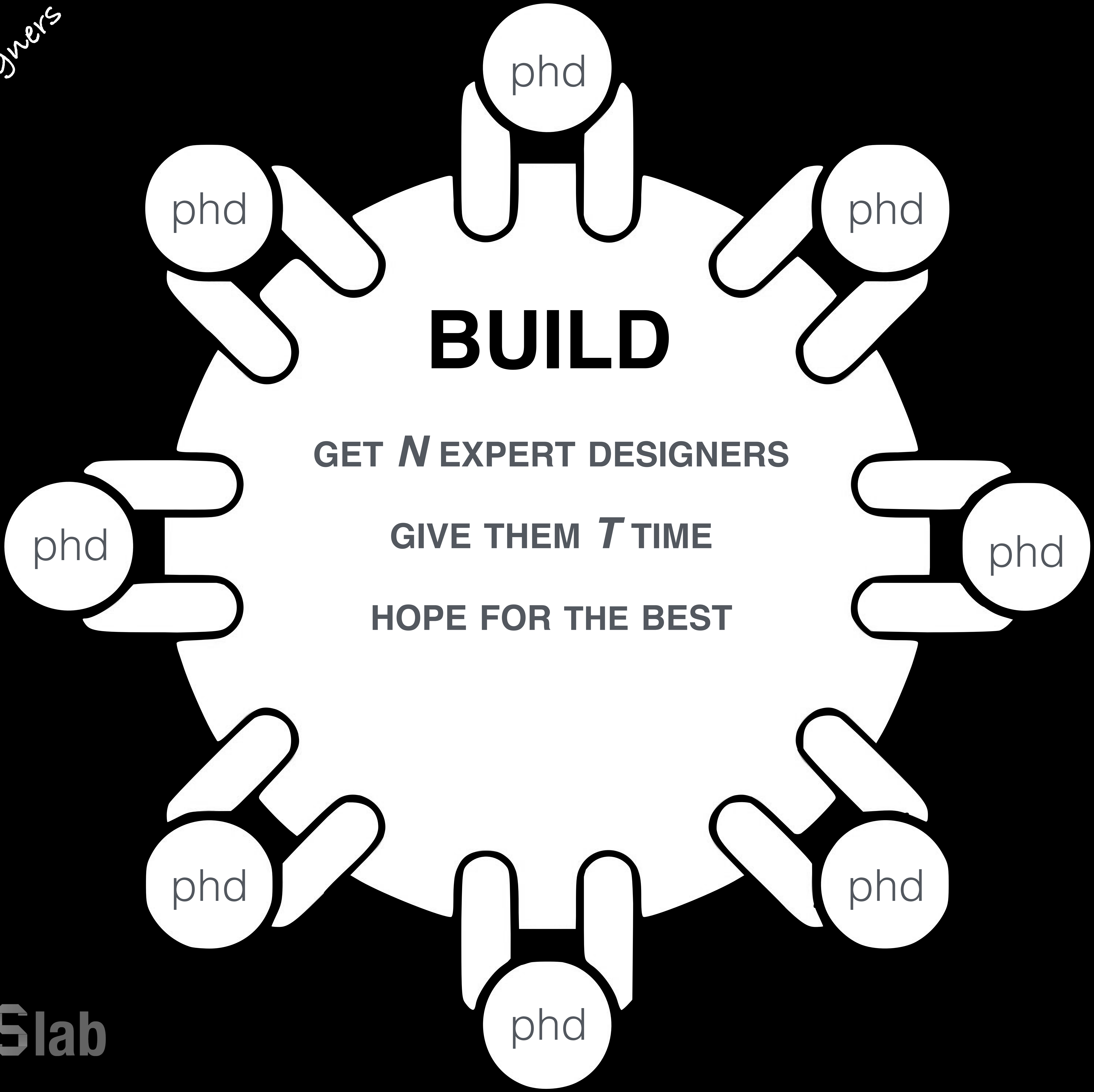
how we **BUILD** systems







the dinning
systems designers



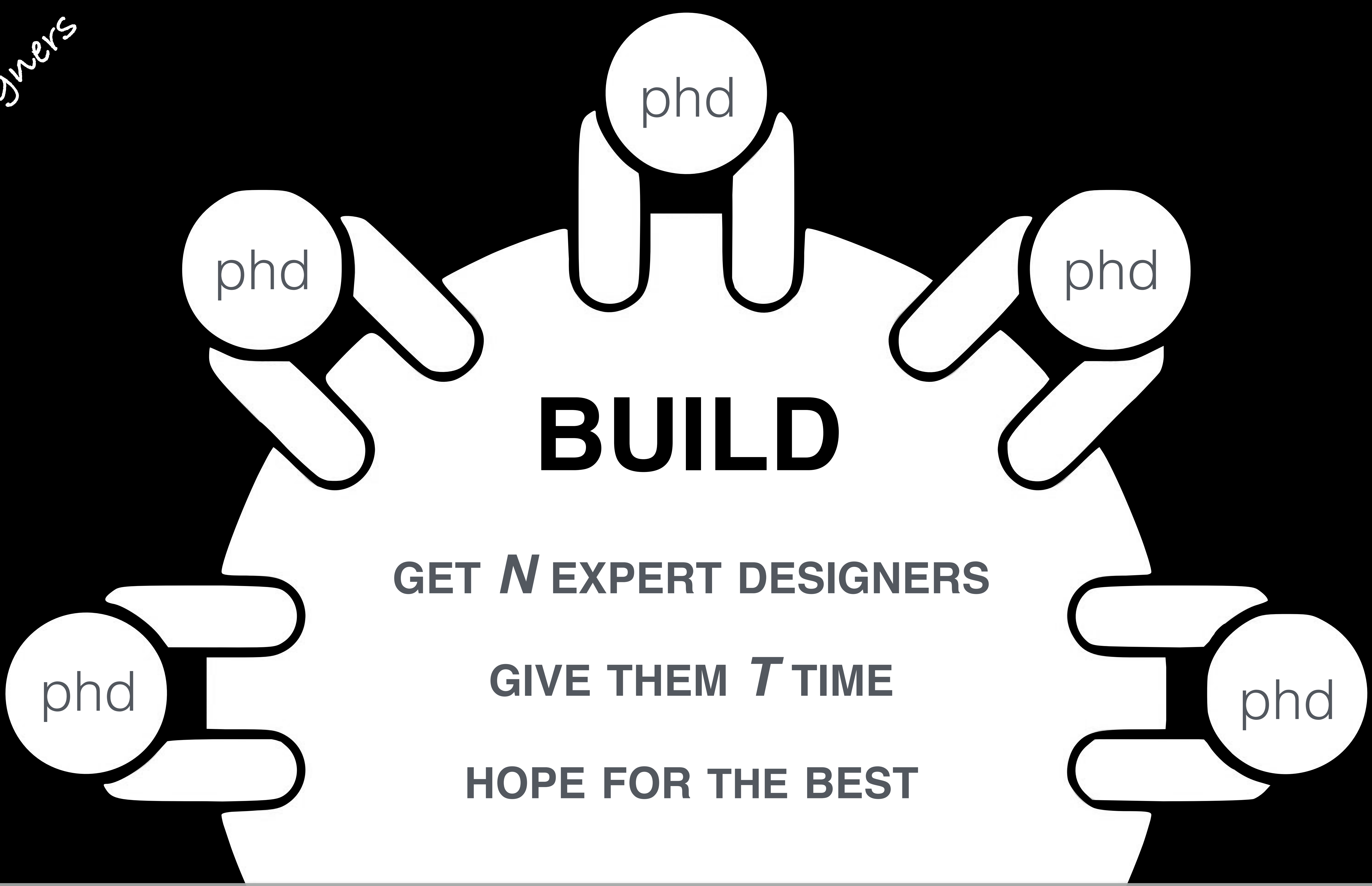
BUILD

GET *N* EXPERT DESIGNERS

GIVE THEM *T* TIME

HOPE FOR THE BEST

the dinning
systems designers

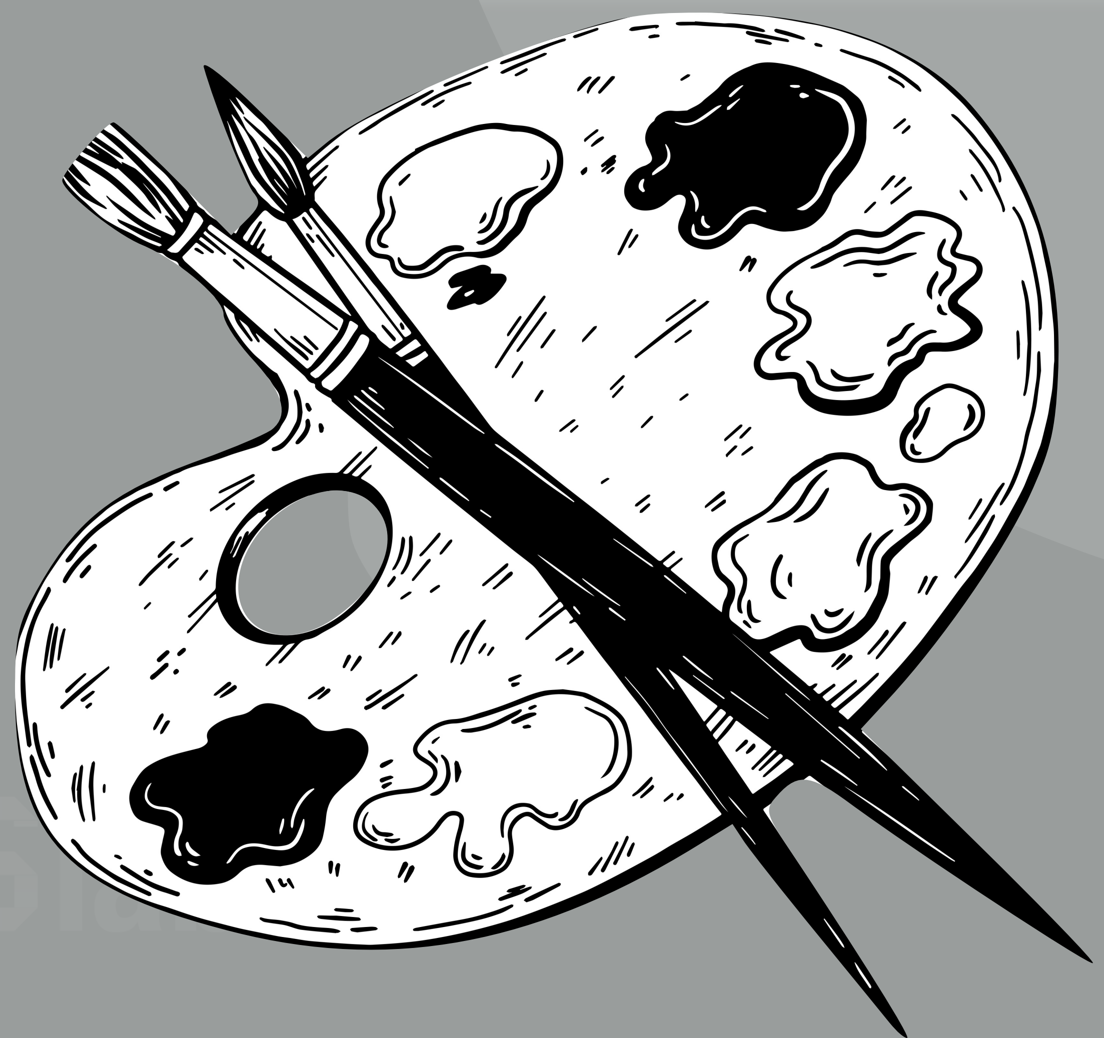


BUILD

GET *N* EXPERT DESIGNERS

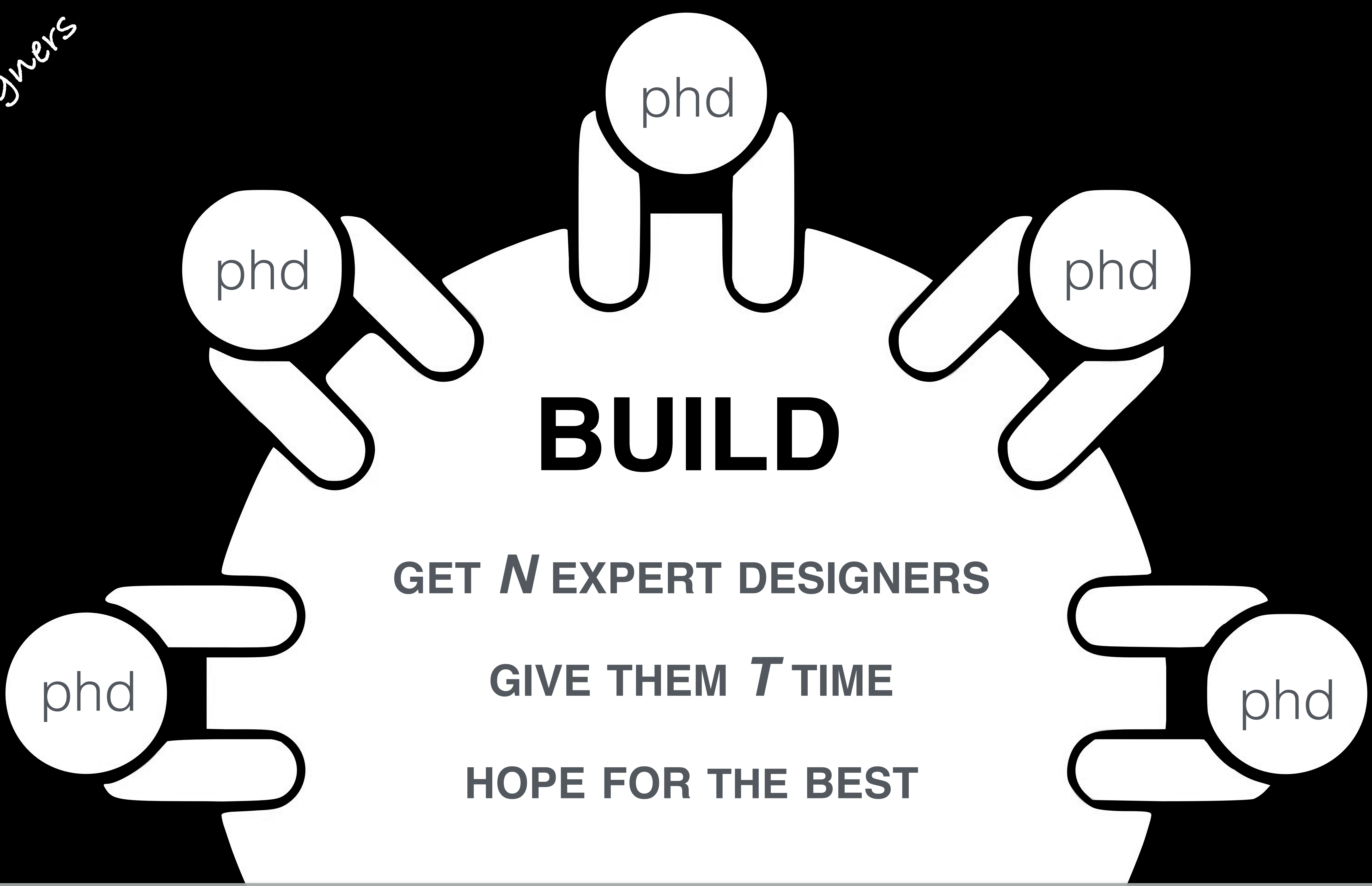
GIVE THEM *T* TIME

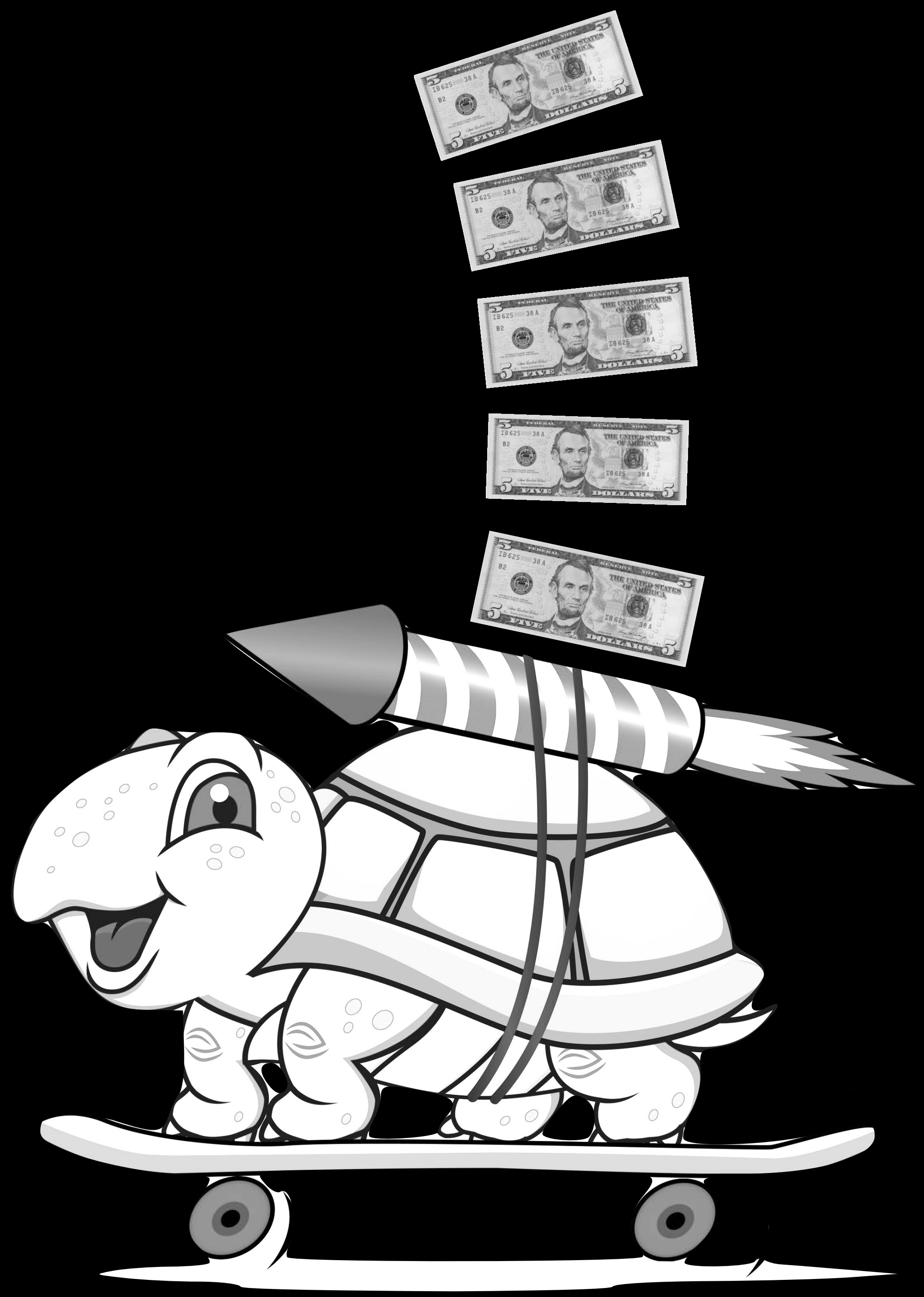
HOPE FOR THE BEST

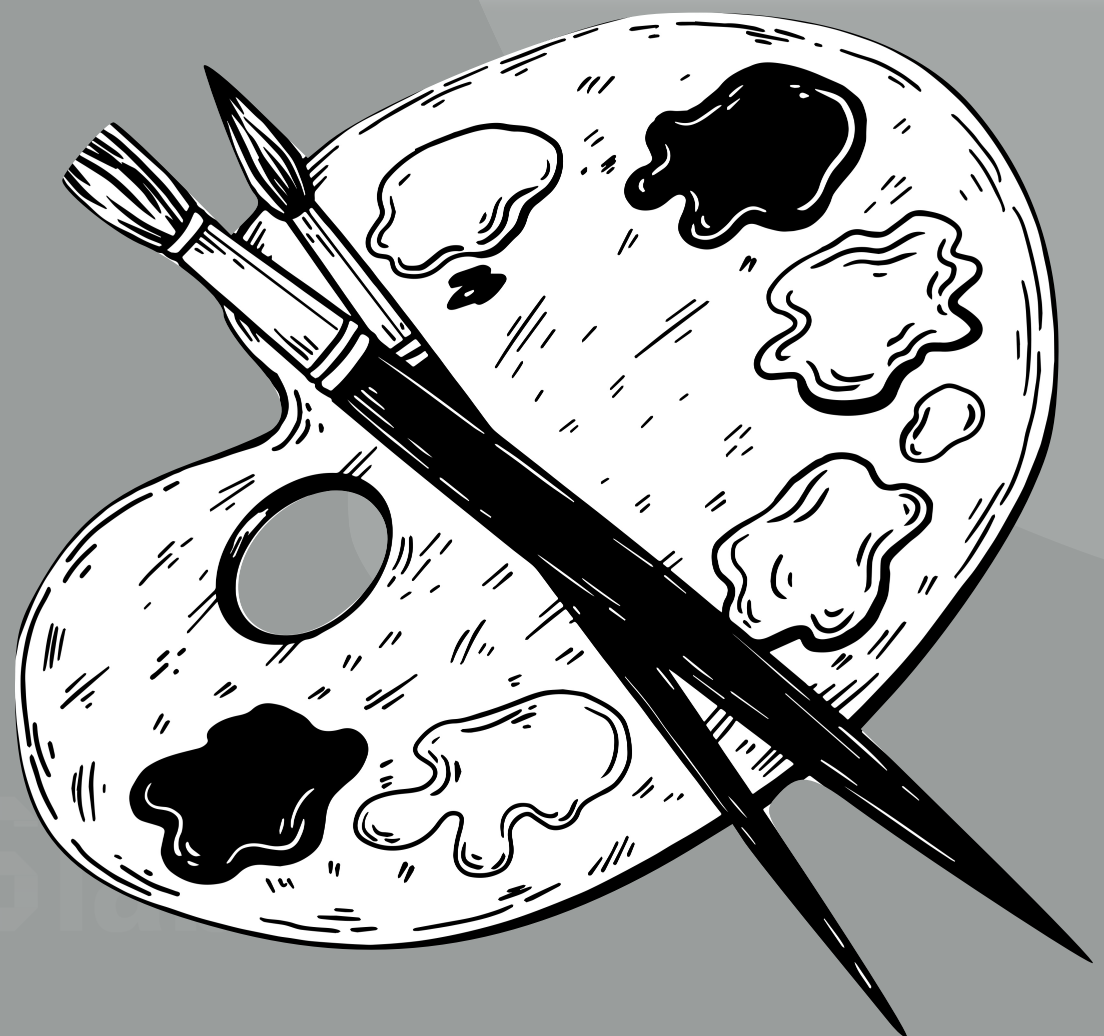


design is an art

the dinning
systems designers







design is an art

Design: 6-7 years
Reasoning: months/impossible

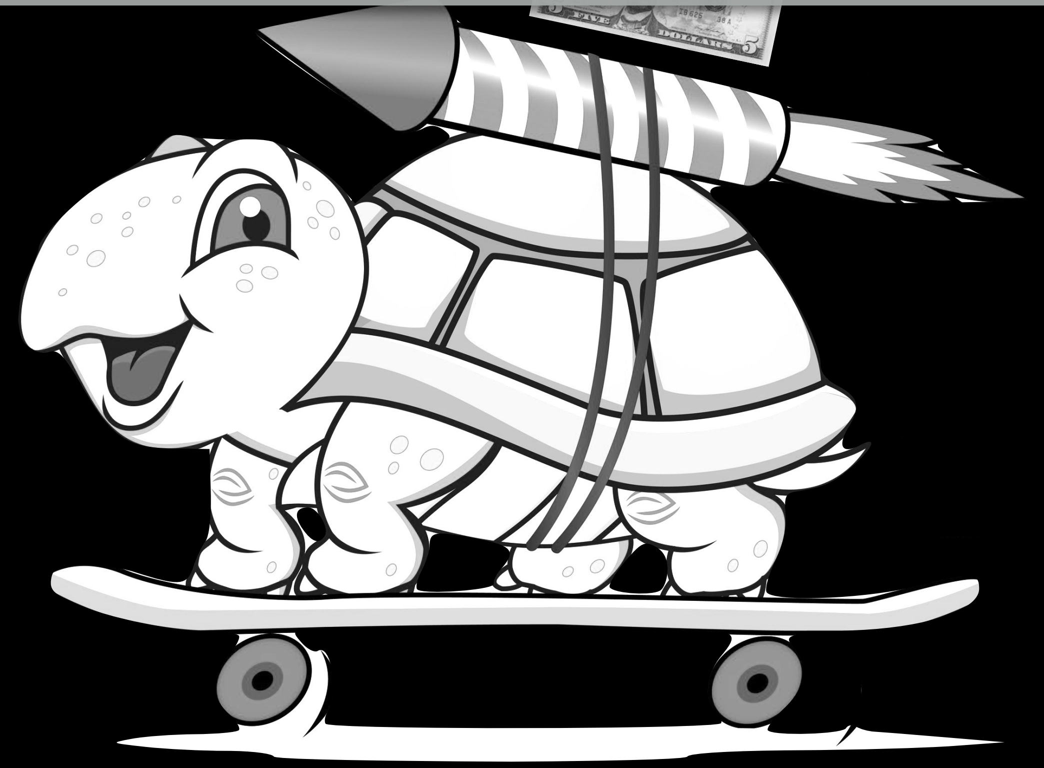
GET *N* EXPERT DESIGNERS

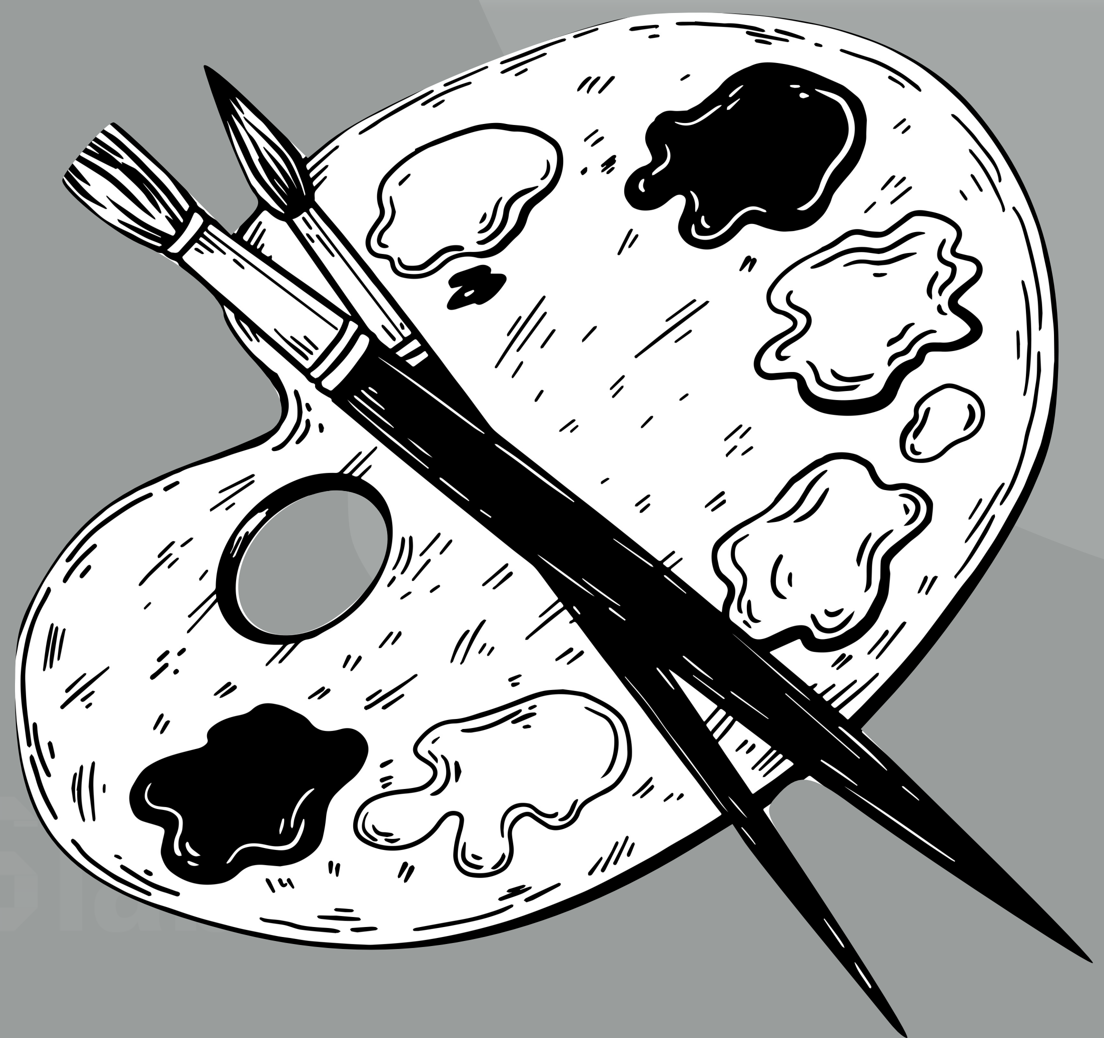
GIVE THEM *T* TIME

HOPE FOR THE BEST

phd

phd

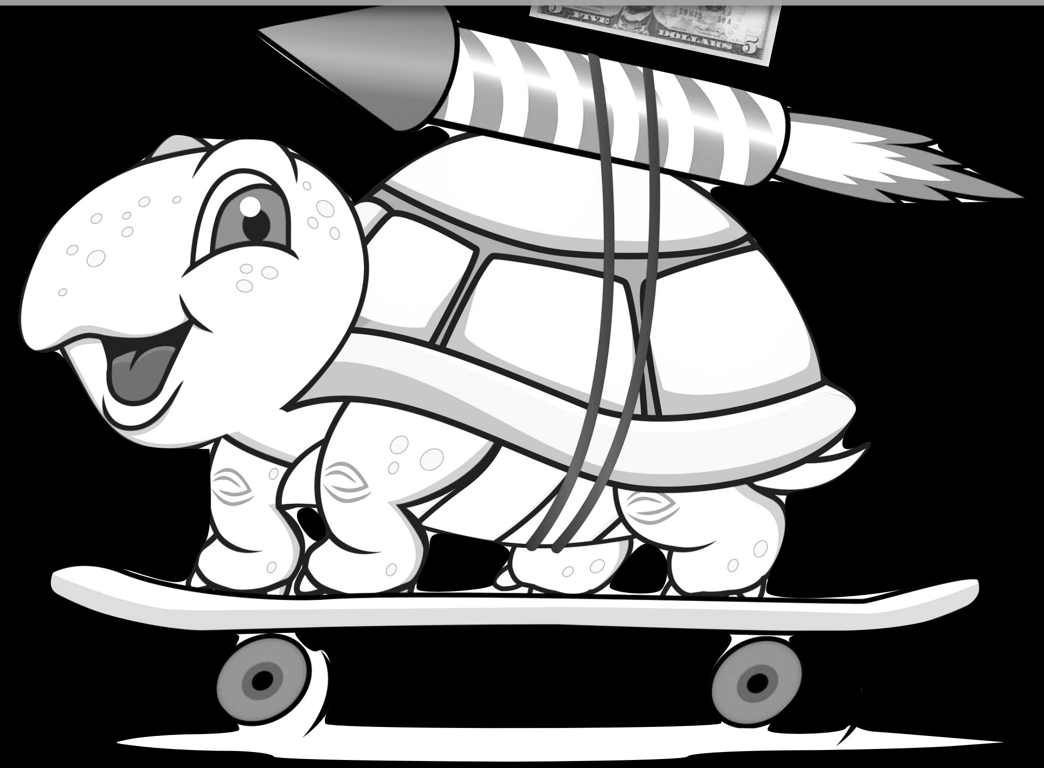




design is an art

sign: … s

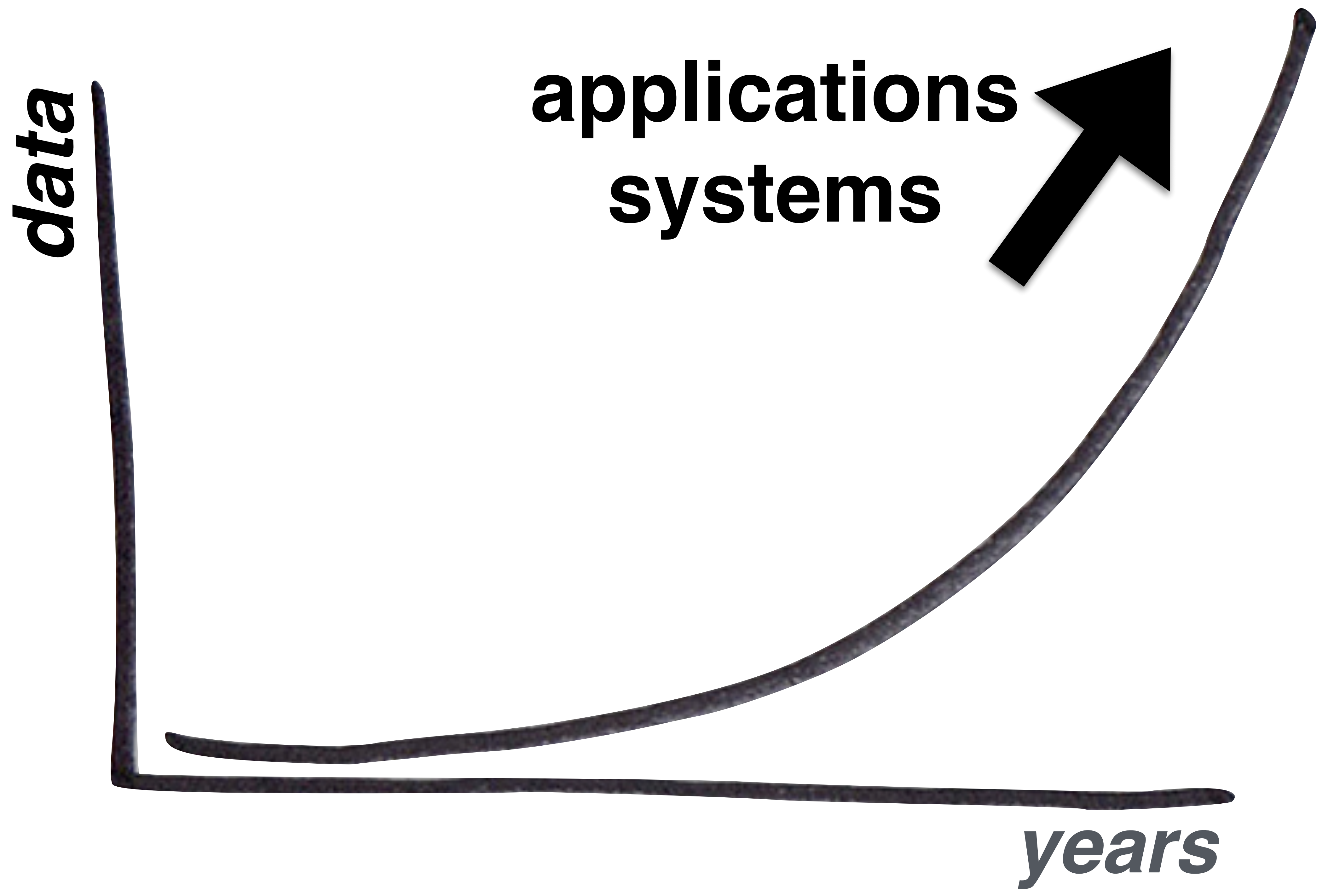
Re … : … mpossible

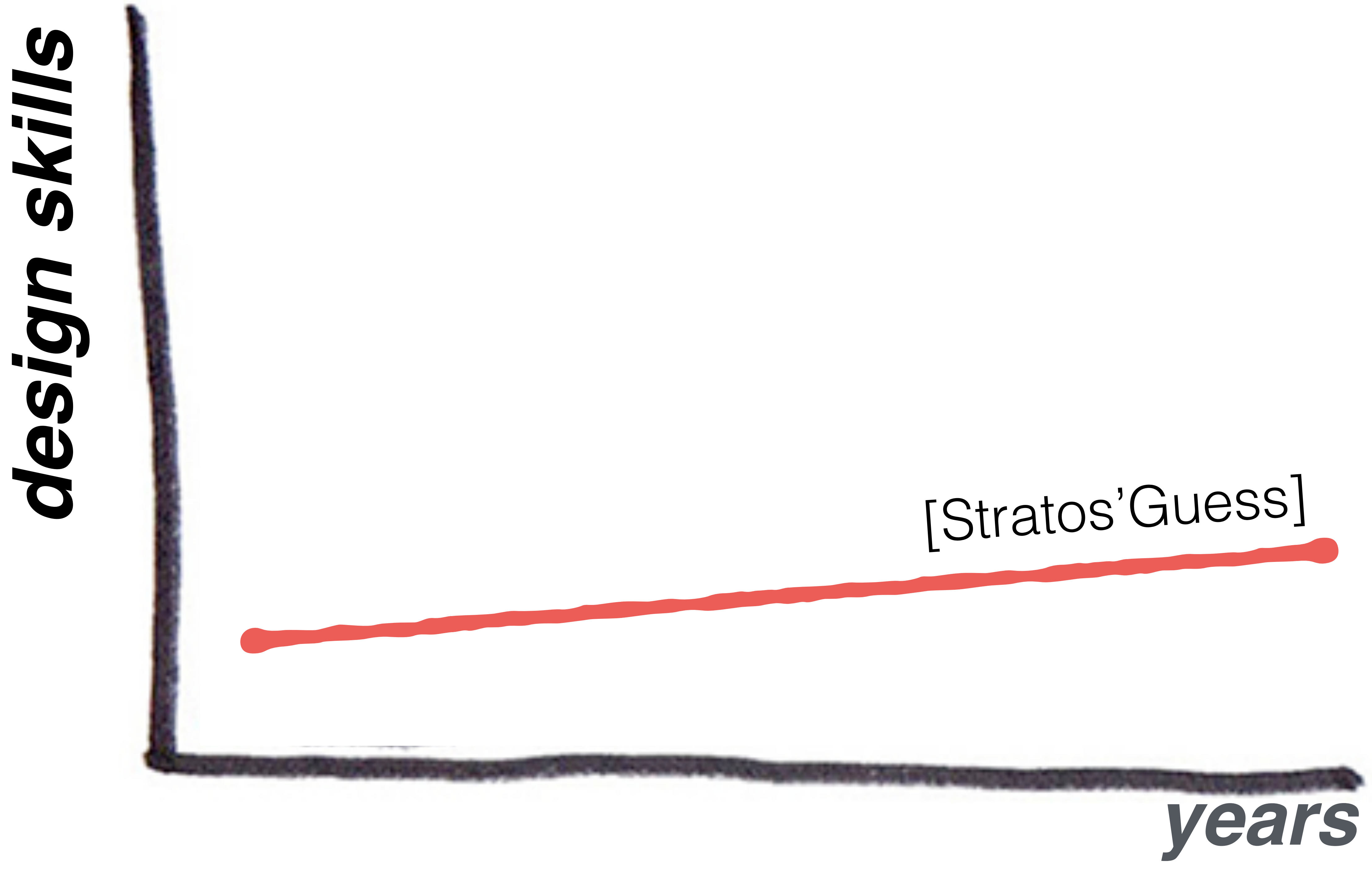


is an art

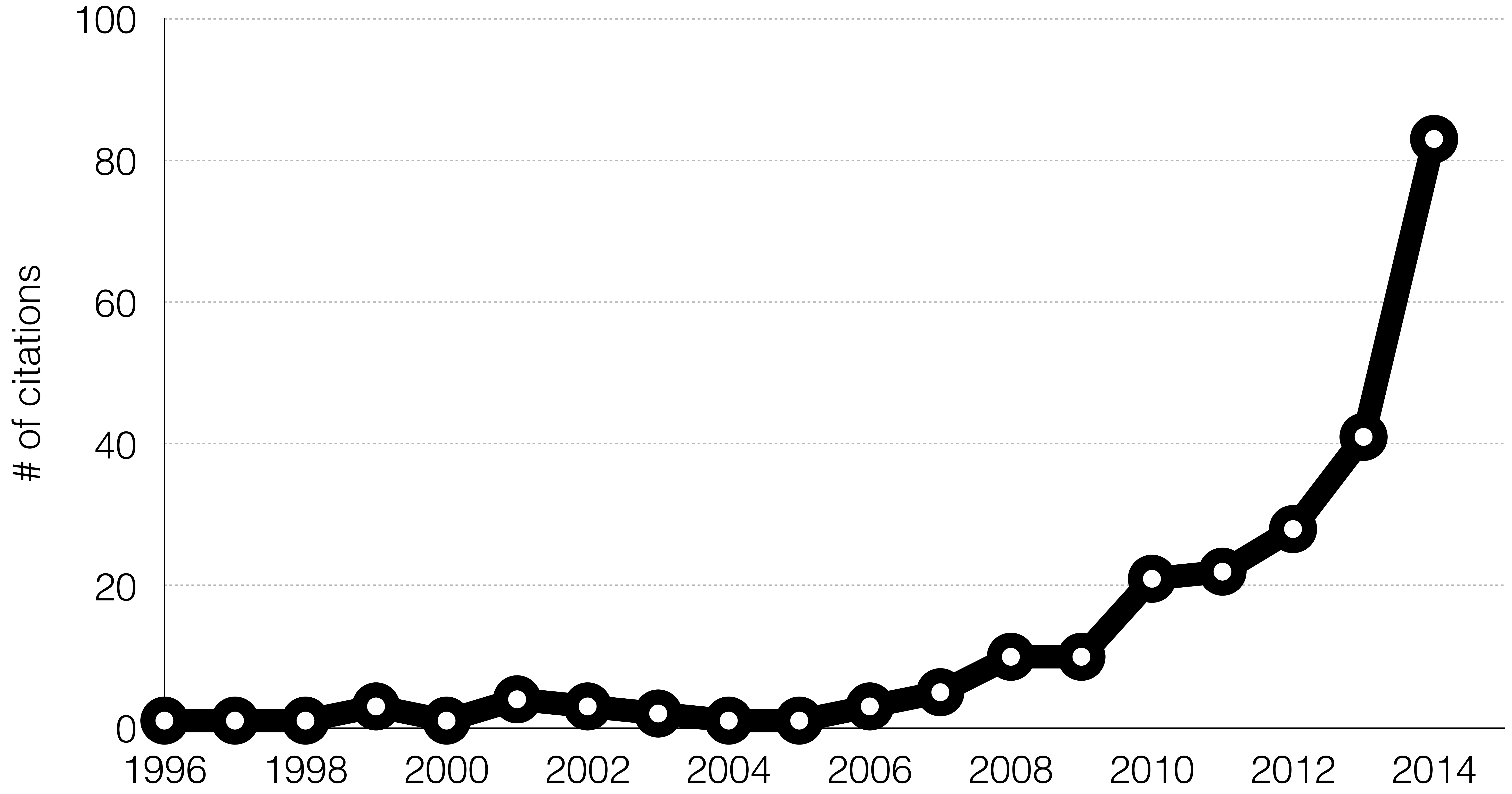

data
hardware
applications
years
phd
phd
is an art

1 design/research skills do not scale





2 no one knows everything out there

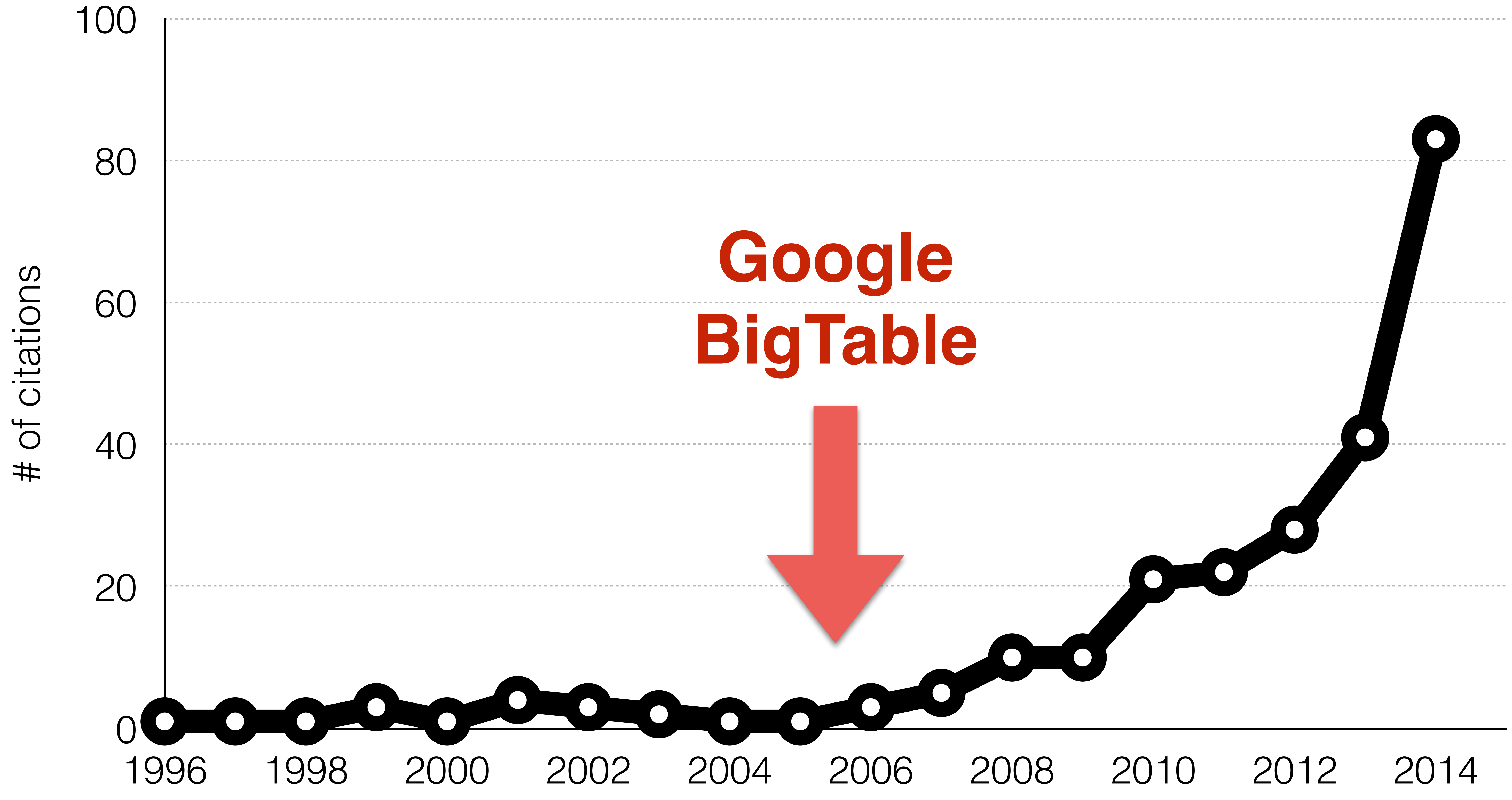


NoSQL storage

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

DASlab
@ Harvard SEAS

2 no one knows everything out there



NoSQL storage

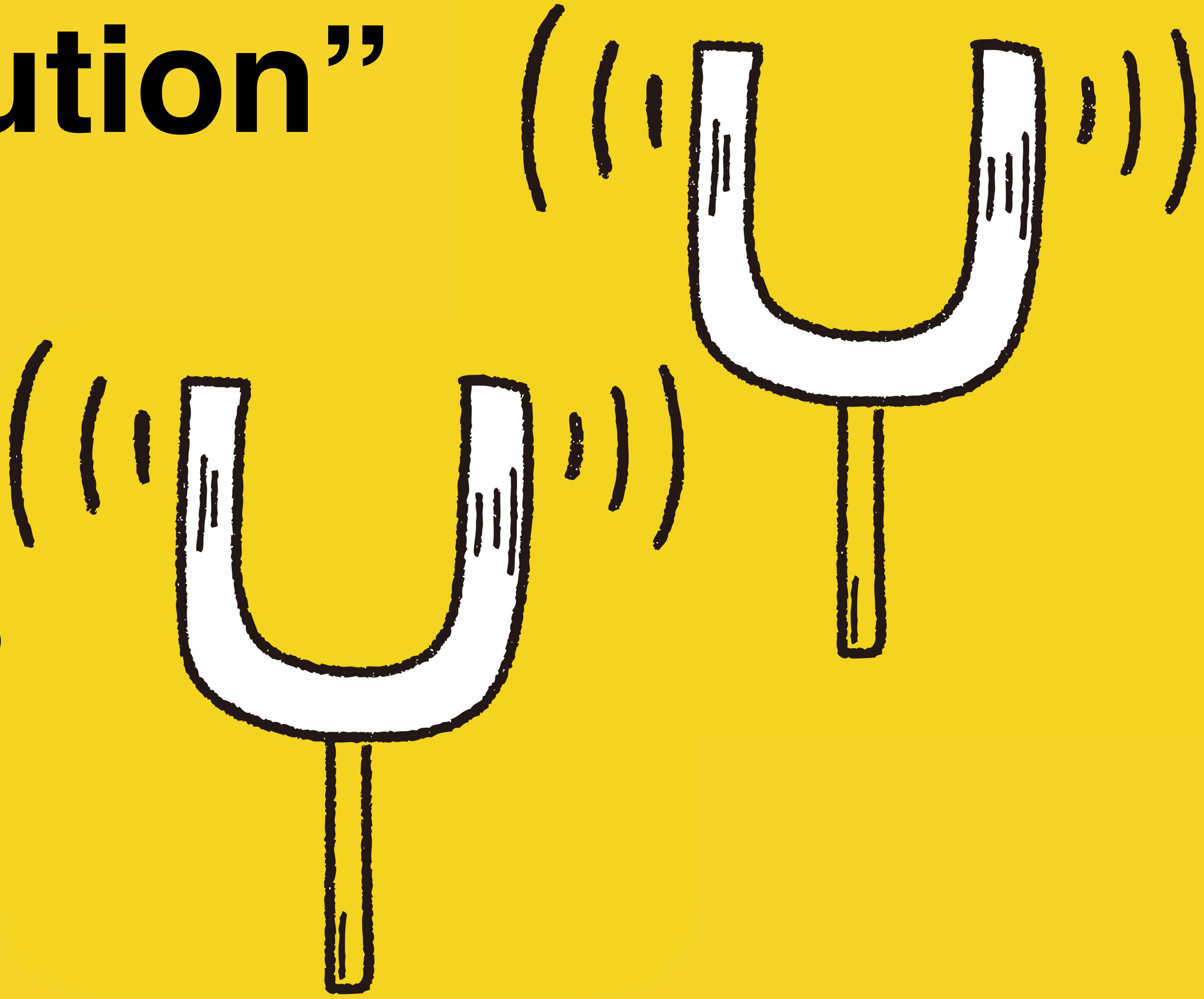
P. O'Neil, E. Cheng, D. Gawlick, E, O'Neil
The log-structured merge-tree (LSM-tree)
Acta Informatica 33 (4): 351–385, 1996

DASlab @ Harvard SEAS

The HIPPO method

"Highest Paid Person's Opinion"

standard “solution”



expose knobs

Some possible ideas

Some possible ideas

1. **Aren’t data systems already “adaptive”, e.g., optimizer makes the best online decision?**

Some possible ideas

1. **Aren't data systems already "adaptive", e.g., optimizer makes the best online decision?**
   Yes, but only around a narrow design space.

Some possible ideas

1. **Aren't data systems already "adaptive", e.g., optimizer makes the best online decision?**
   Yes, but only around a narrow design space.

2. **Aren't adaptive data systems architectures able to adapt to new applications?**

Some possible ideas

1. **Aren't data systems already "adaptive", e.g., optimizer makes the best online decision?**
   Yes, but only around a narrow design space.

2. **Aren't adaptive data systems architectures able to adapt to new applications?**
   Yes, better than #1 (e.g., query adaptivity), but still only around a narrow design space.

Some possible ideas

1. **Aren’t data systems already “adaptive”, e.g., optimizer makes the best online decision?**
   Yes, but only around a narrow design space.

2. **Aren’t adaptive data systems architectures able to adapt to new applications?**
   Yes, better than #1 (e.g., query adaptivity), but still only around a narrow design space.

3. **Aren’t learned system components able to adapt even more?**

Some possible ideas

1. **Aren't data systems already "adaptive", e.g., optimizer makes the best online decision?**
   Yes, but only around a narrow design space.

2. **Aren't adaptive data systems architectures able to adapt to new applications?**
   Yes, better than #1 (e.g., query adaptivity), but still only around a narrow design space.

3. **Aren't learned system components able to adapt even more?**
   Yes, better than #2 (e.g., data adaptivity), but still only around a narrow design space.

Some possible ideas

1. **Aren't data systems already "adaptive", e.g., optimizer makes the best online decision?**
   Yes, but only around a narrow design space.

2. **Aren't adaptive data systems architectures able to adapt to new applications?**
   Yes, better than #1 (e.g., query adaptivity), but still only around a narrow design space.

3. **Aren't learned system components able to adapt even more?**
   Yes, better than #2 (e.g., data adaptivity), but still only around a narrow design space.

4. **Can't we just throw ML into the problem? ChatGPT?**

Some possible ideas

1. **Aren't data systems already "adaptive", e.g., optimizer makes the best online decision?**
   Yes, but only around a narrow design space.

2. **Aren't adaptive data systems architectures able to adapt to new applications?**
   Yes, better than #1 (e.g., query adaptivity), but still only around a narrow design space.

3. **Aren't learned system components able to adapt even more?**
   Yes, better than #2 (e.g., data adaptivity), but still only around a narrow design space.

4. **Can't we just throw ML into the problem? ChatGPT?**
   Yes, but the programming design space is massive. A correct design is not a desired one.

Some possible ideas

1. **Aren't data systems already "adaptive", e.g., optimizer makes the best online decision?**
   Yes, but only around a narrow design space.

2. **Aren't adaptive data systems architectures able to adapt to new applications?**
   Yes, better than #1 (e.g., query adaptivity), but still only around a narrow design space.

3. **Aren't learned system components able to adapt even more?**
   Yes, better than #2 (e.g., data adaptivity), but still only around a narrow design space.

4. **Can't we just throw ML into the problem? ChatGPT?**
   Yes, but the programming design space is massive. A correct design is not a desired one.

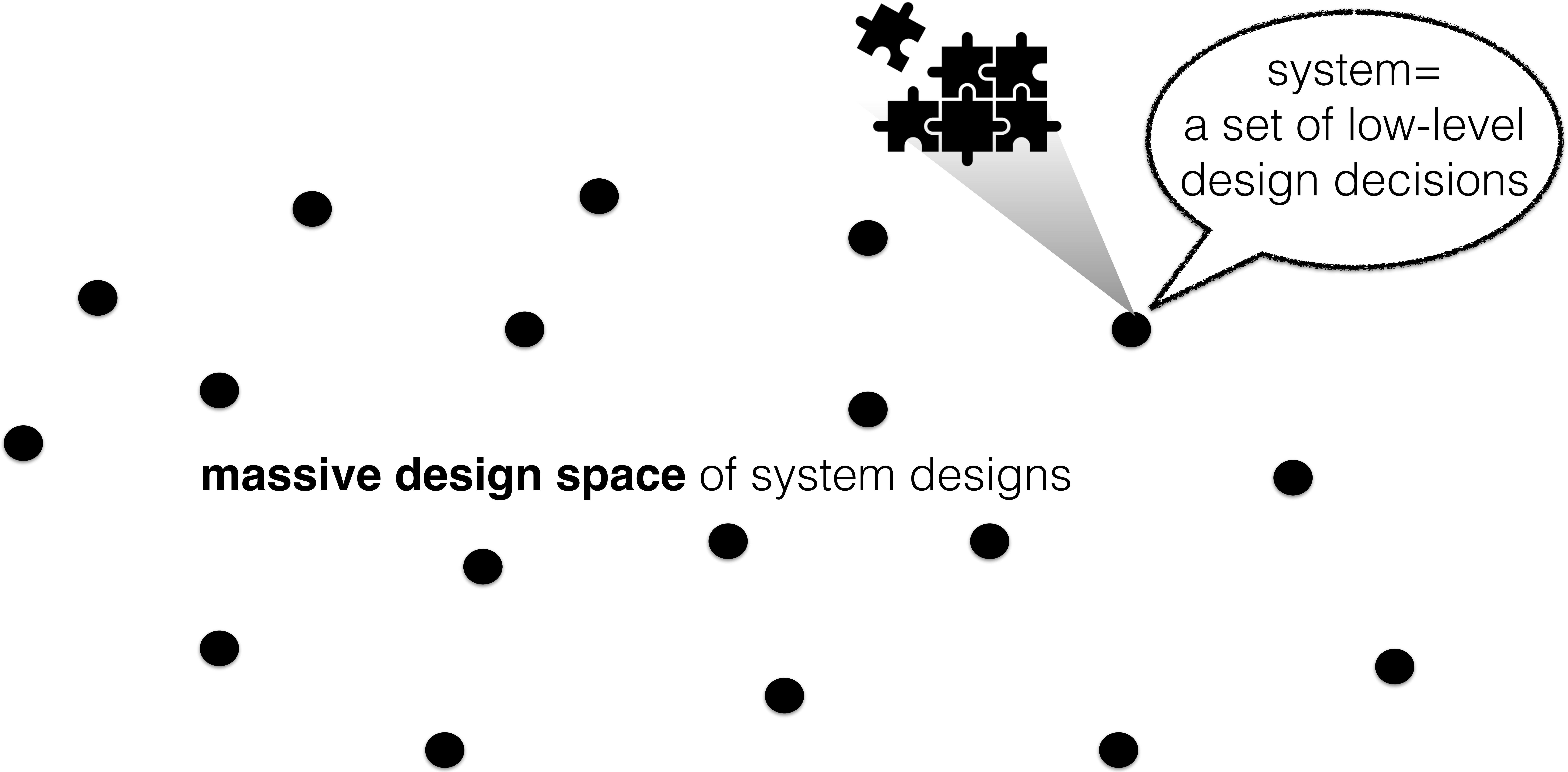
These ideas can lead to better systems but we need something more to

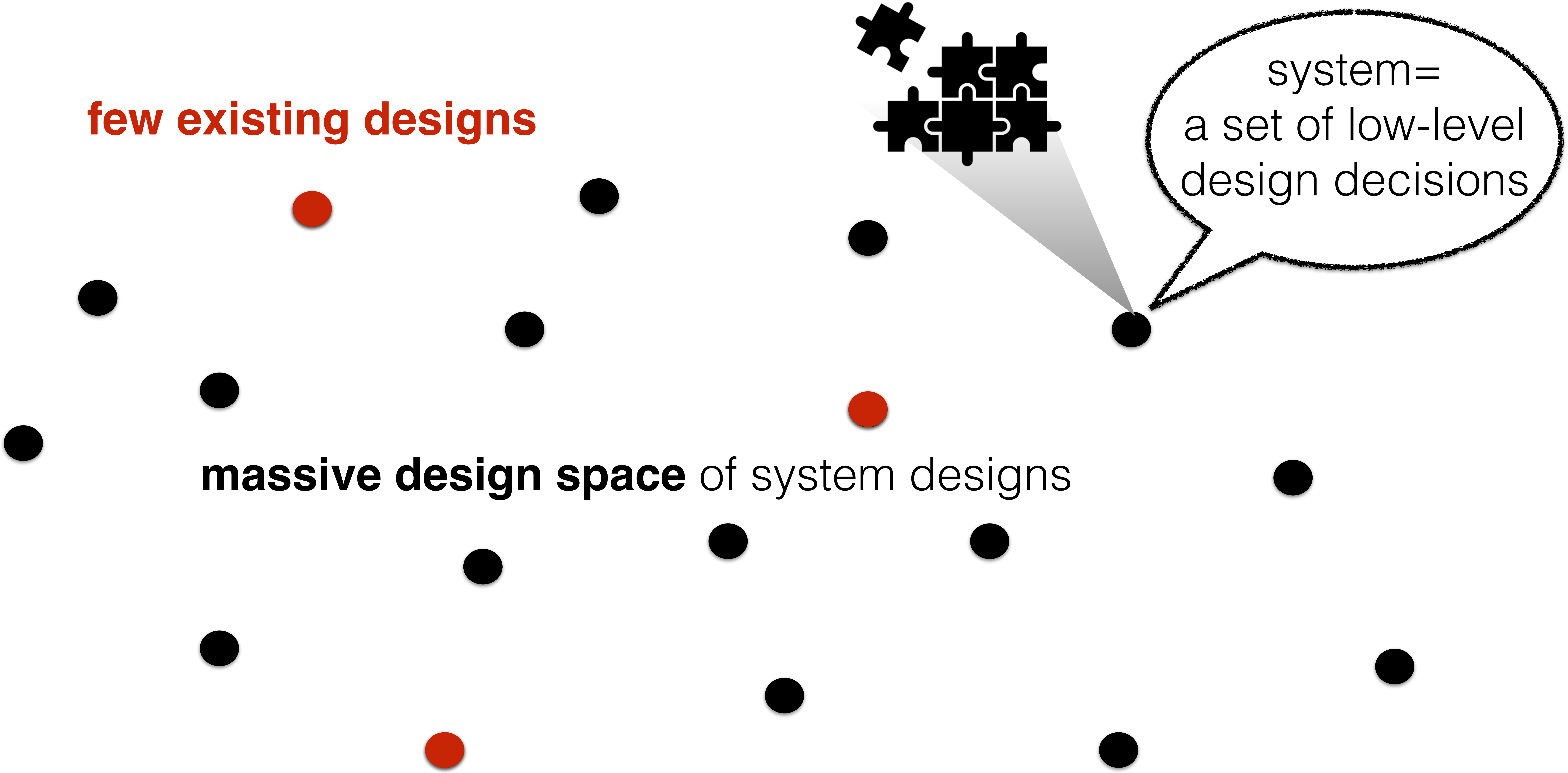
FIND FAST THE BEST POSSIBLE DESIGN

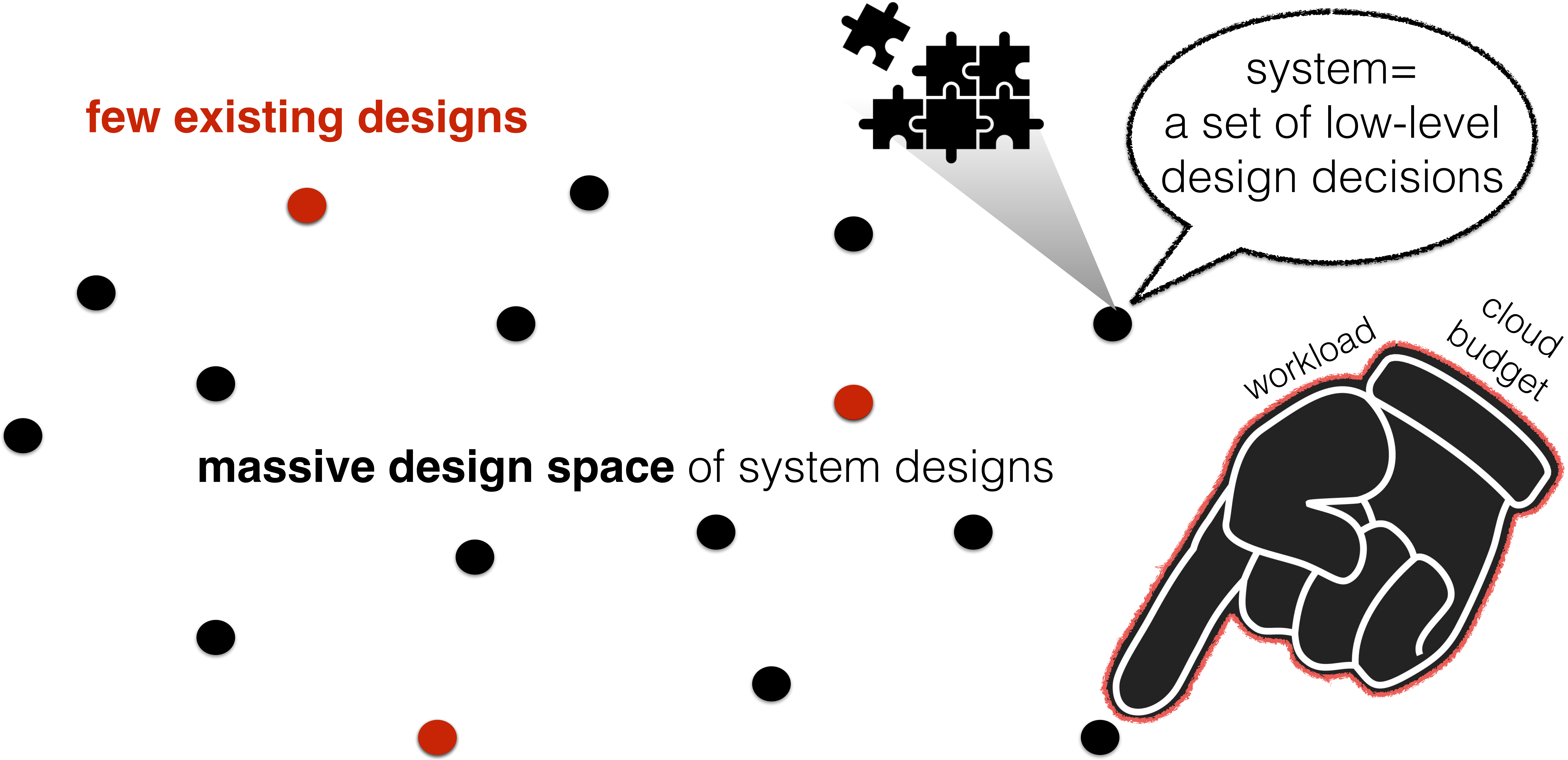
SELF-DESIGNING SYSTEMS

Automatically invent & build the perfect system for any new application

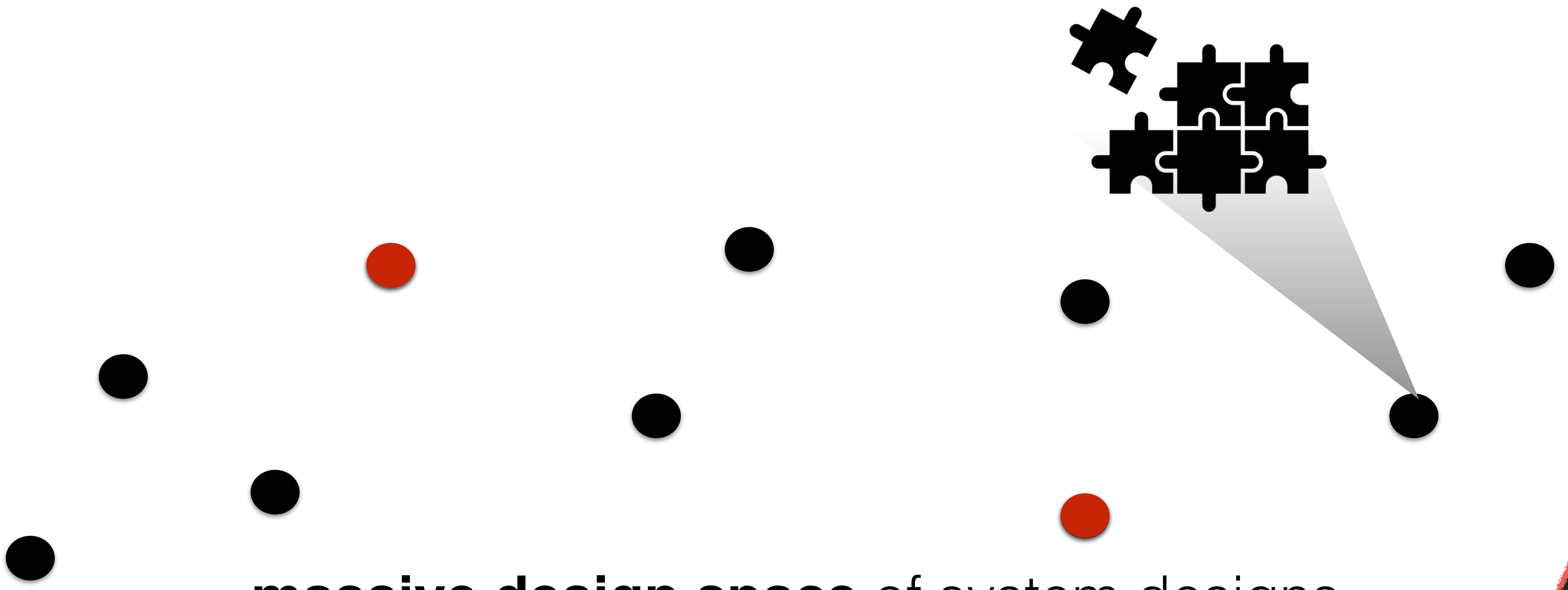
massive design space of system designs


system=
a set of low-level
design decisions
massive design space of system designs


few existing designs
system=
a set of low-level
design decisions
massive design space of system designs



DASlab @ Harvard SEAS



massive design space of system designs

reasoning: understand all the design decisions & their impact

HOW
DO WE
START

SLAs

robustness

concurrency

multi-tenancy

data types

HOW DO WE START

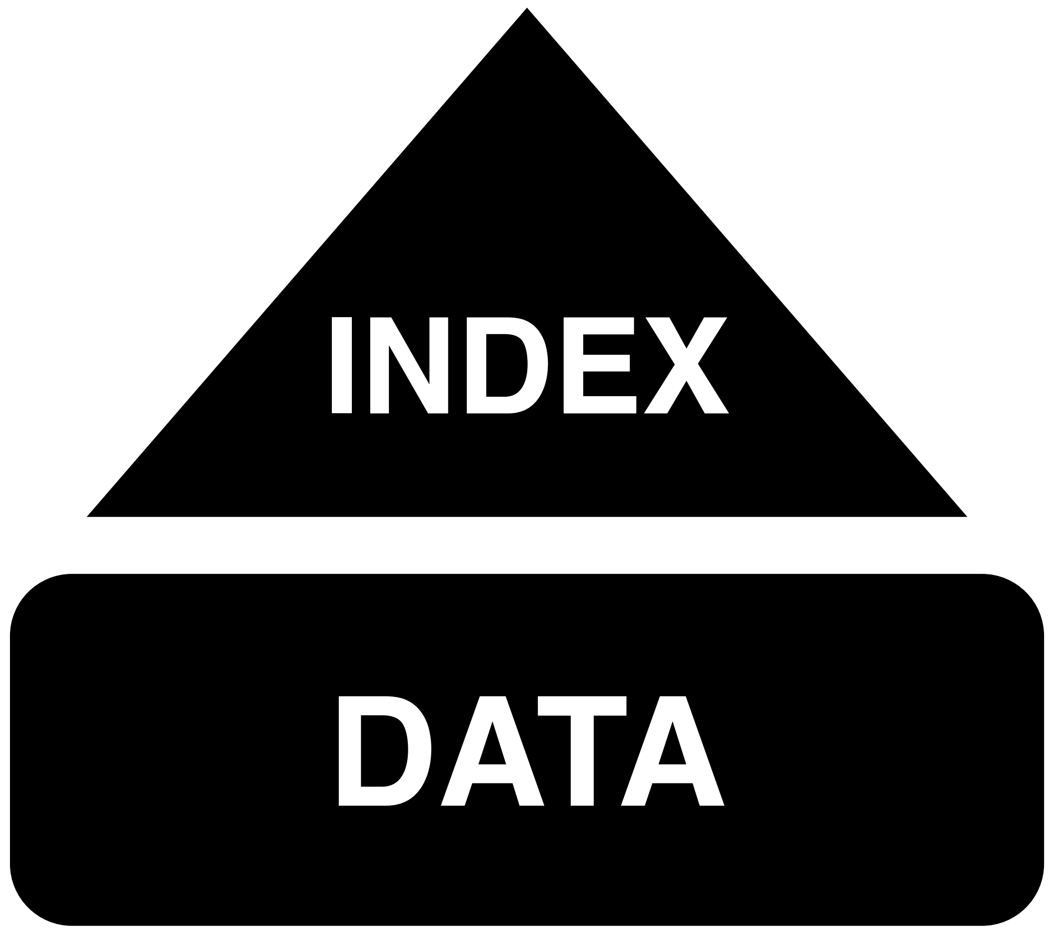
indexing

complex operations

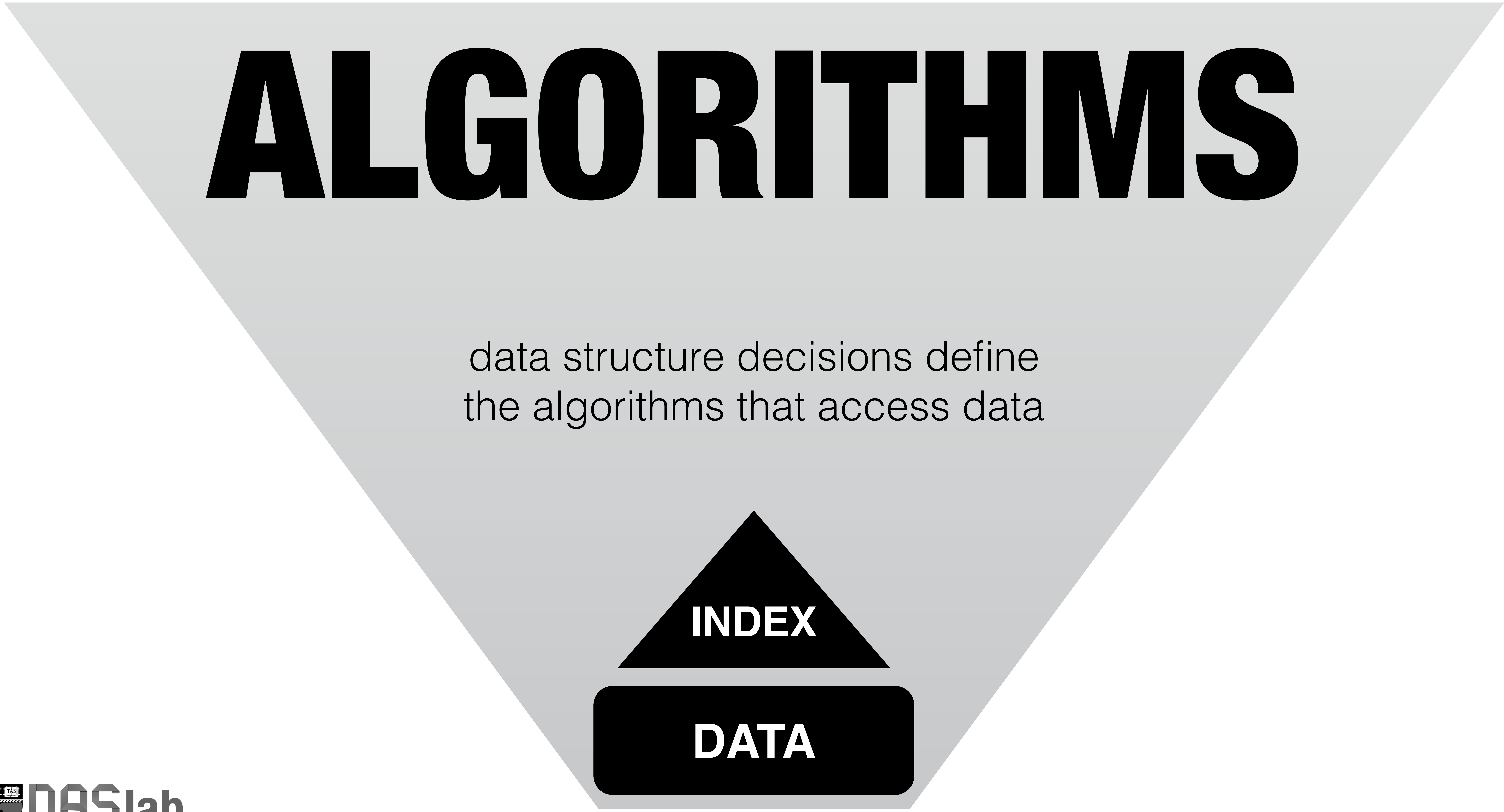
optimizer

cloud

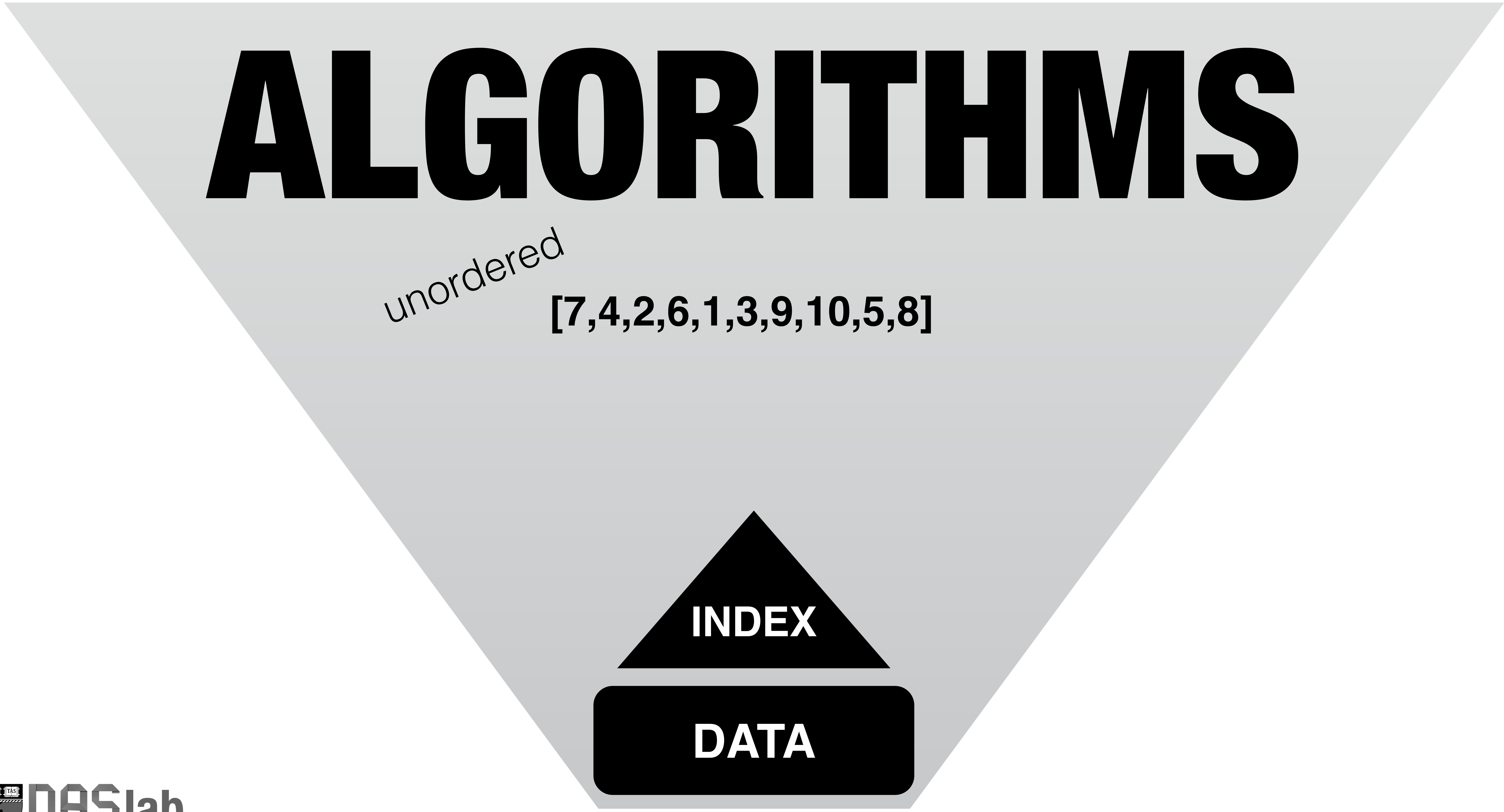
hardware

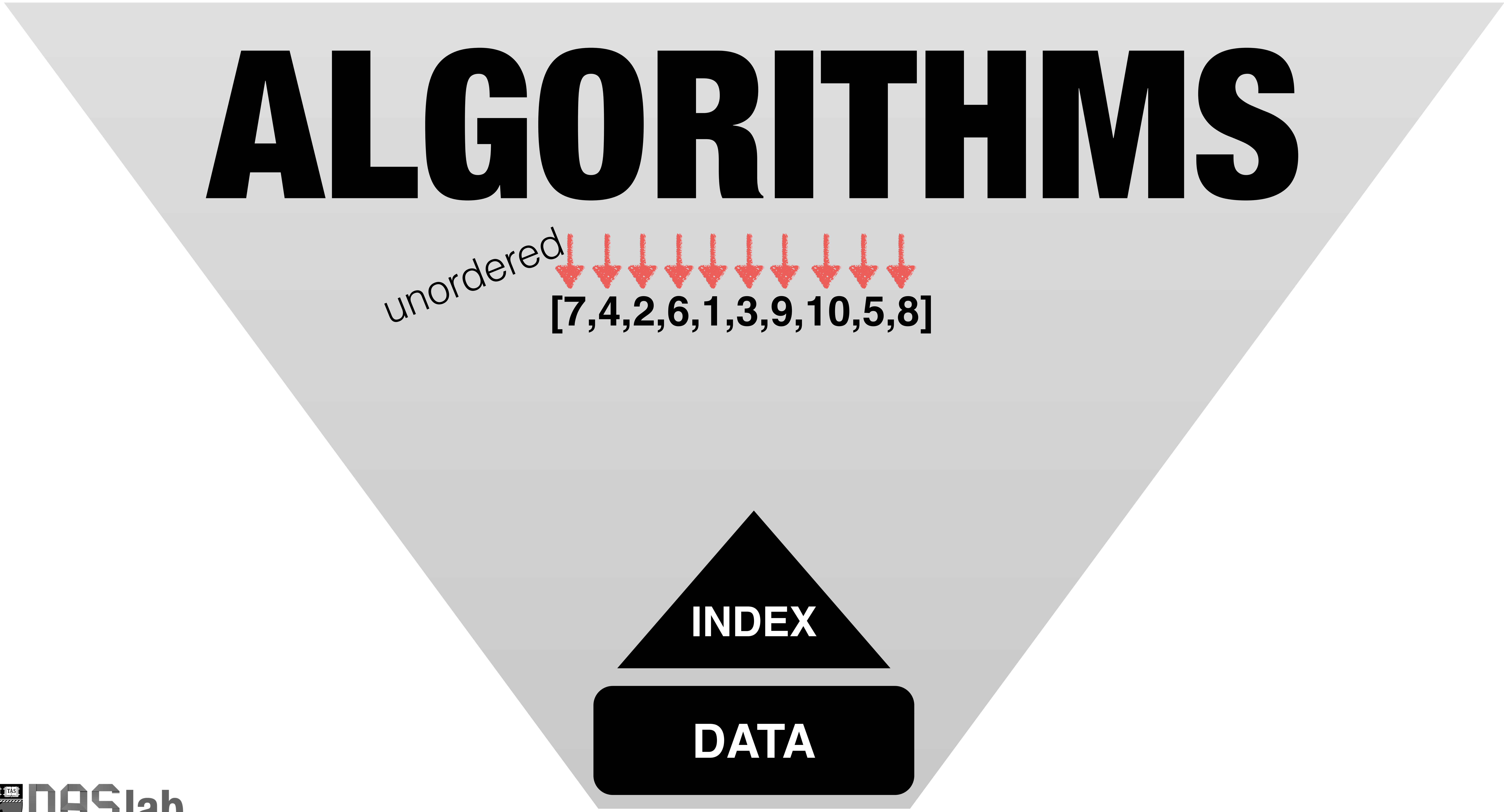


HOW DO WE START

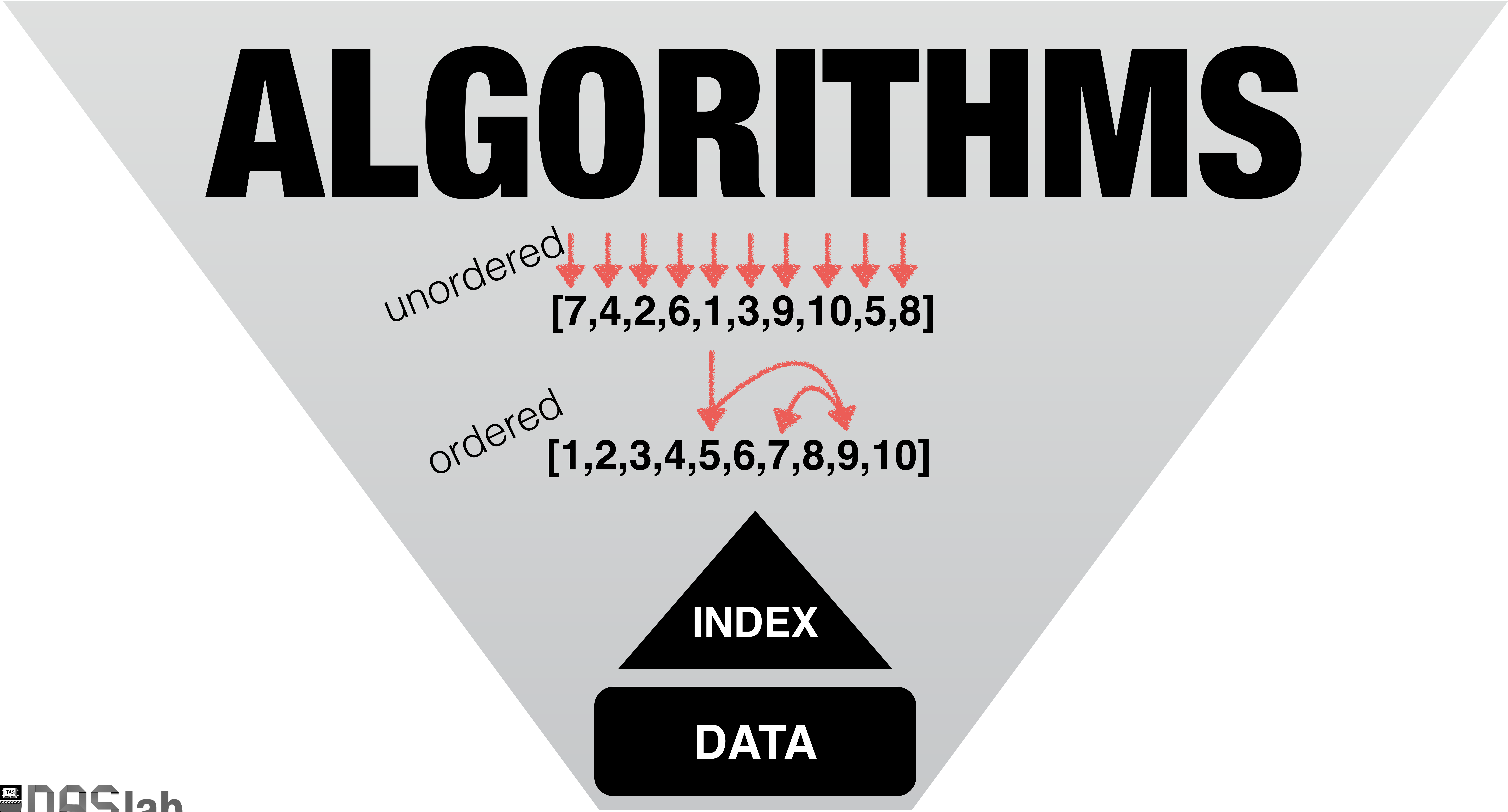


DASlab
@ Harvard SEAS


ALGORITHMS
unordered
[7,4,2,6,1,3,9,10,5,8]
INDEX
DATA



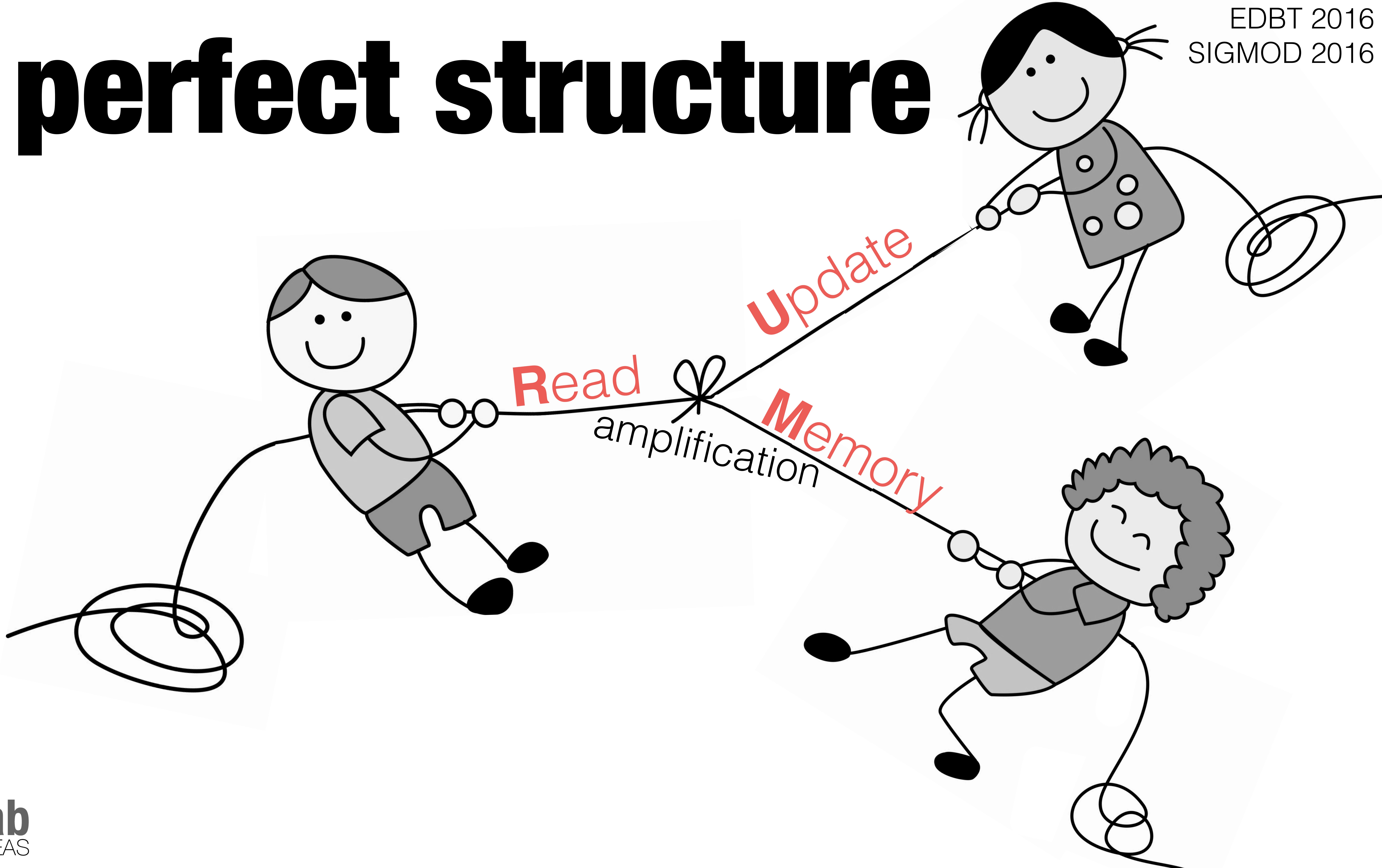
DASlab
@ Harvard SEAS



DASlab @ Harvard SEAS

no perfect structure

EDBT 2016
SIGMOD 2016



no perfect structure

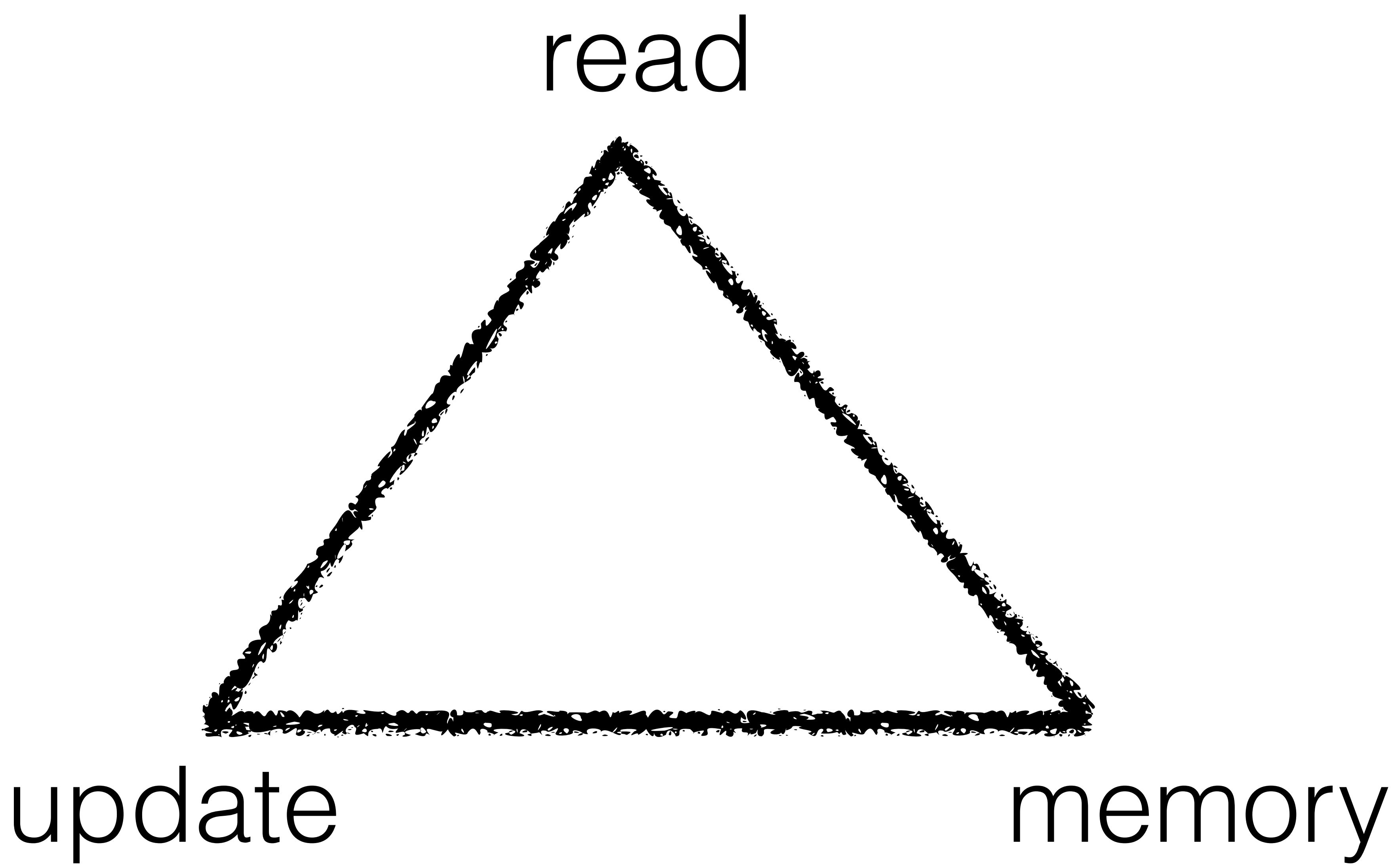
EDBT 2016
SIGMOD 2016

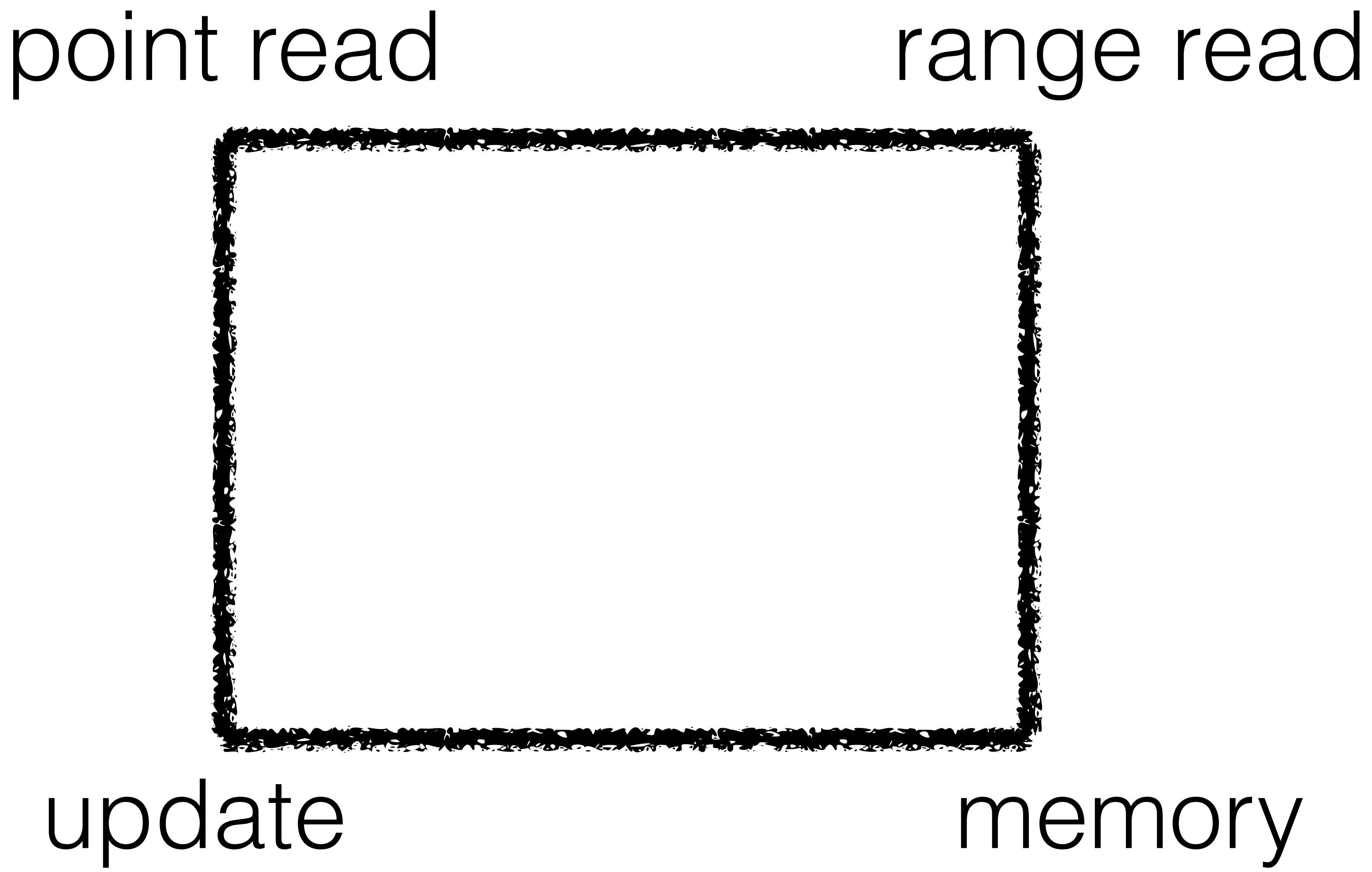
DASlab
@ Harvard SEAS

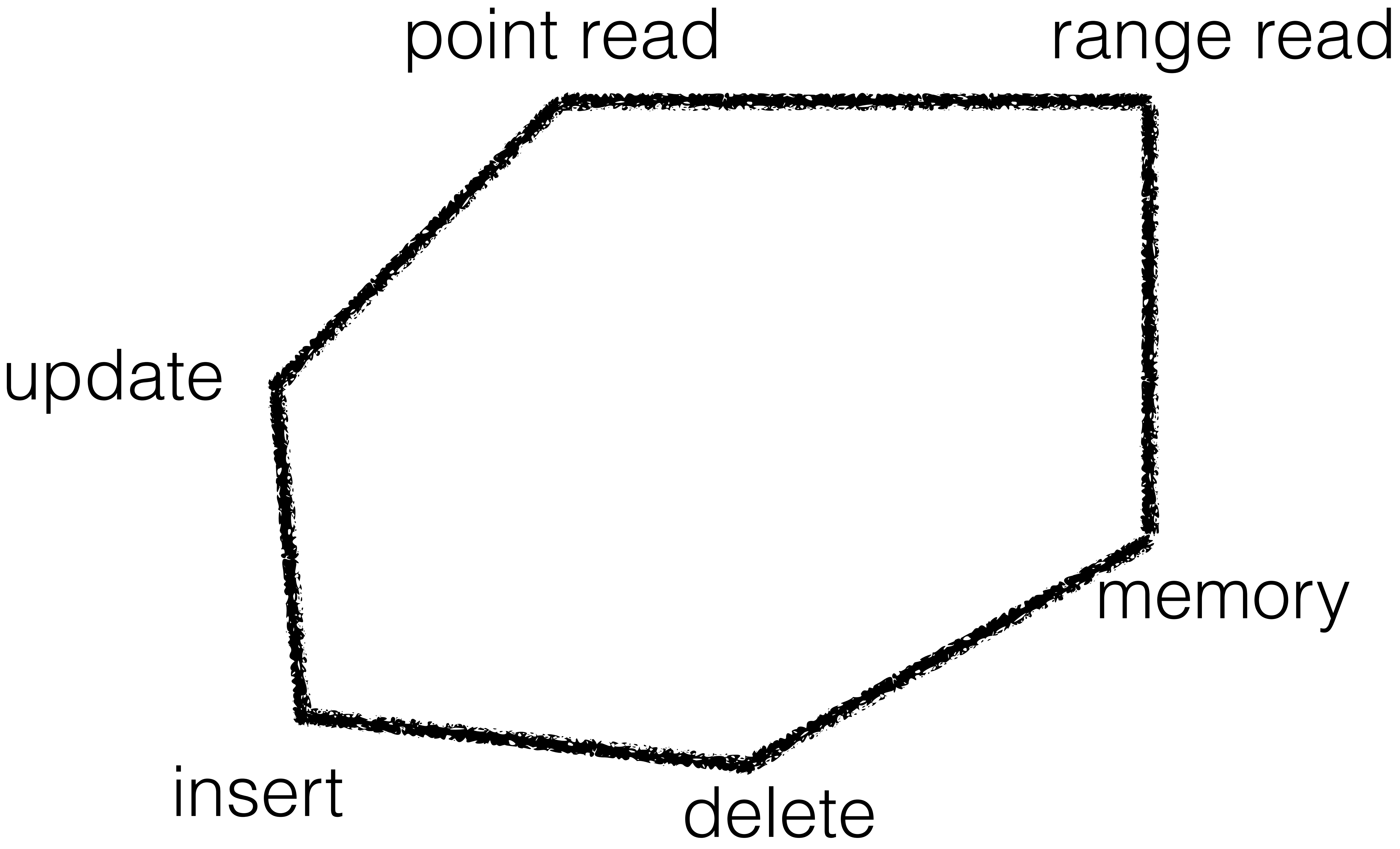
no perfect structure

EDBT 2016
SIGMOD 2016

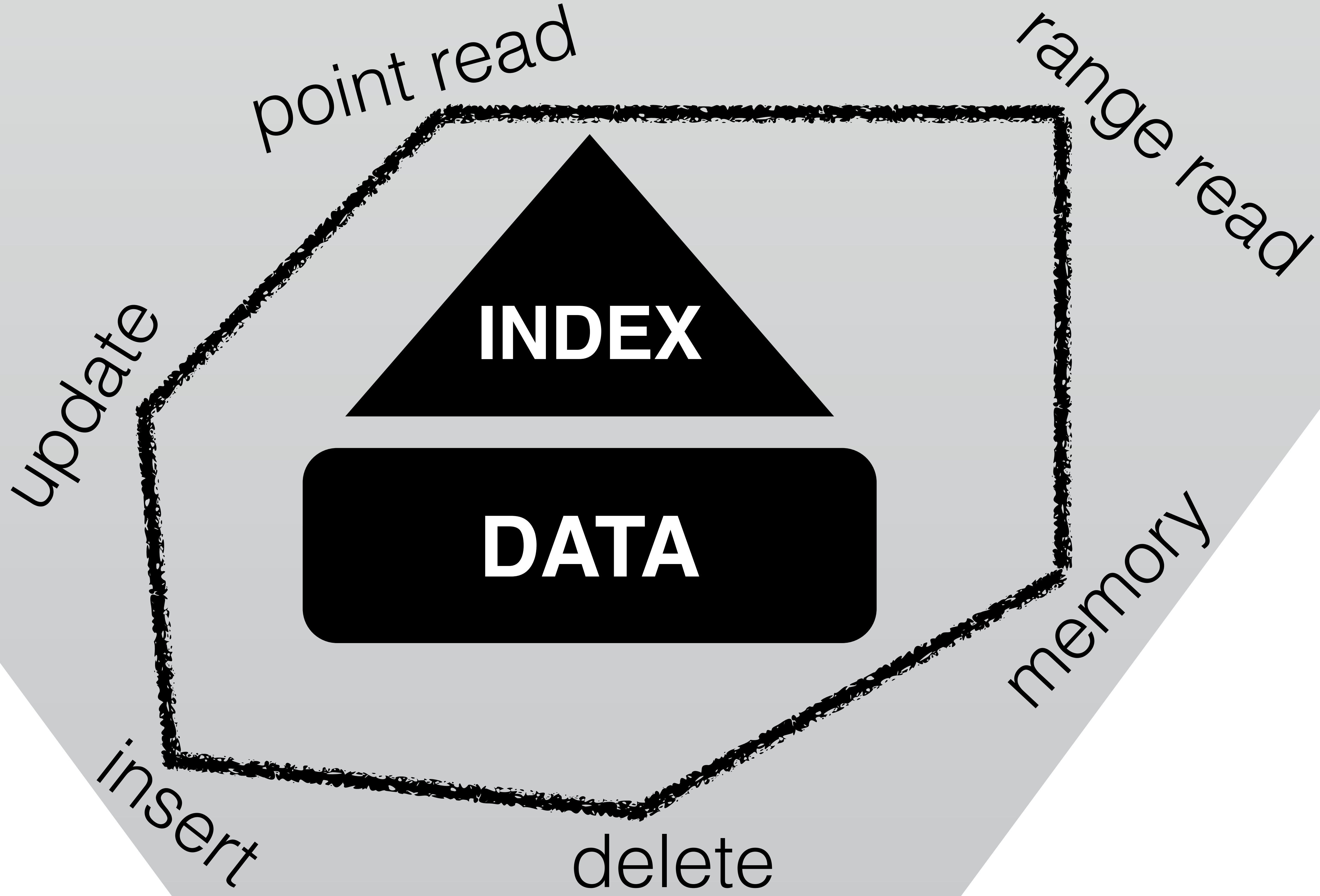
DASlab
@ Harvard SEAS


read
update
memory

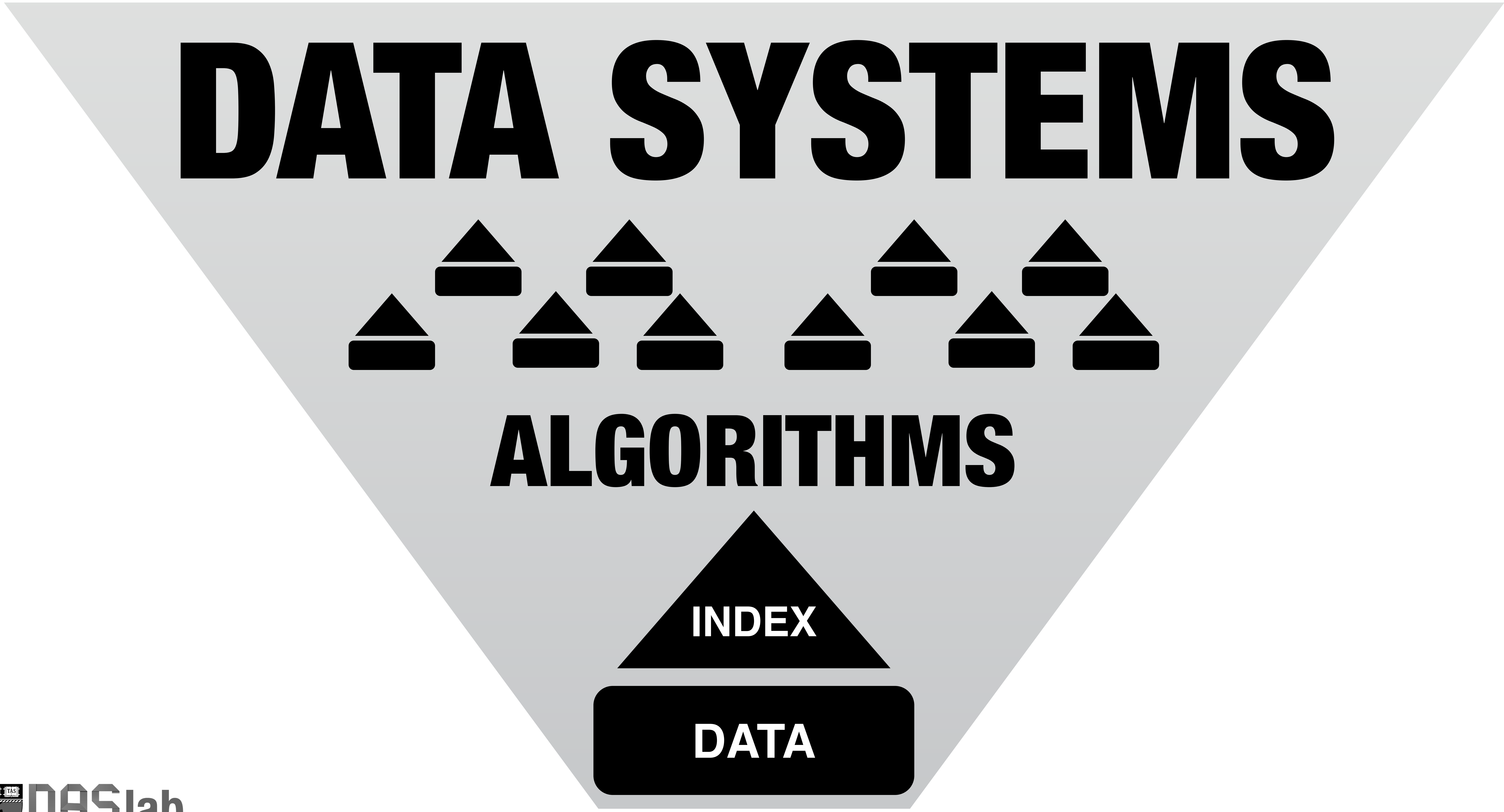

point read
range read
update
memory


point read
range read
update
memory
insert
delete

ALGORITHMS



DASlab
@ Harvard SEAS

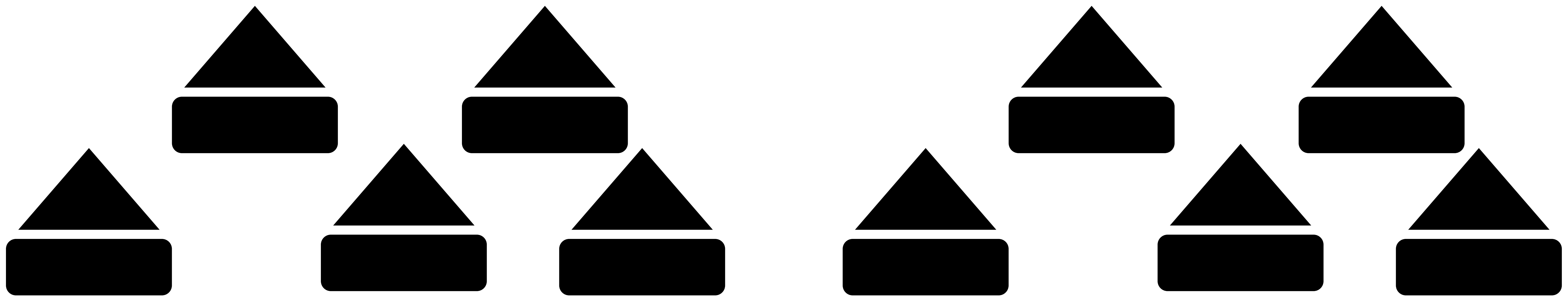

DATA SYSTEMS
ALGORITHMS
INDEX
DATA

NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN

MACHINE LEARNING, SQL, CRYPTO, SCIENCE

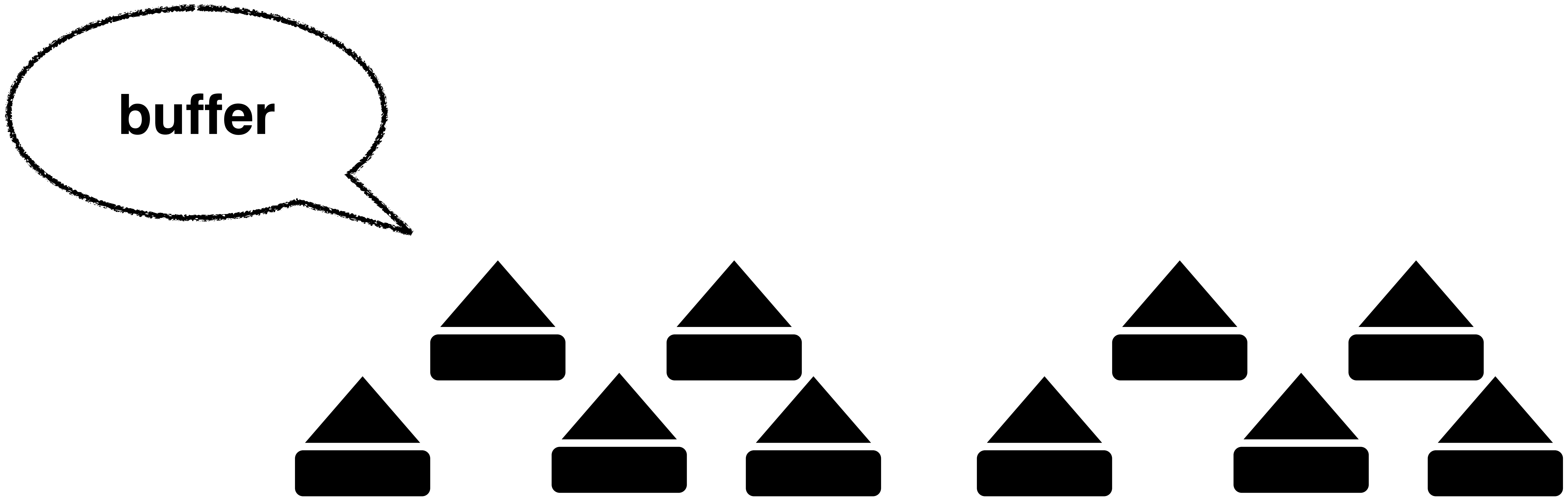


NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN

MACHINE LEARNING, SQL, CRYPTO, SCIENCE

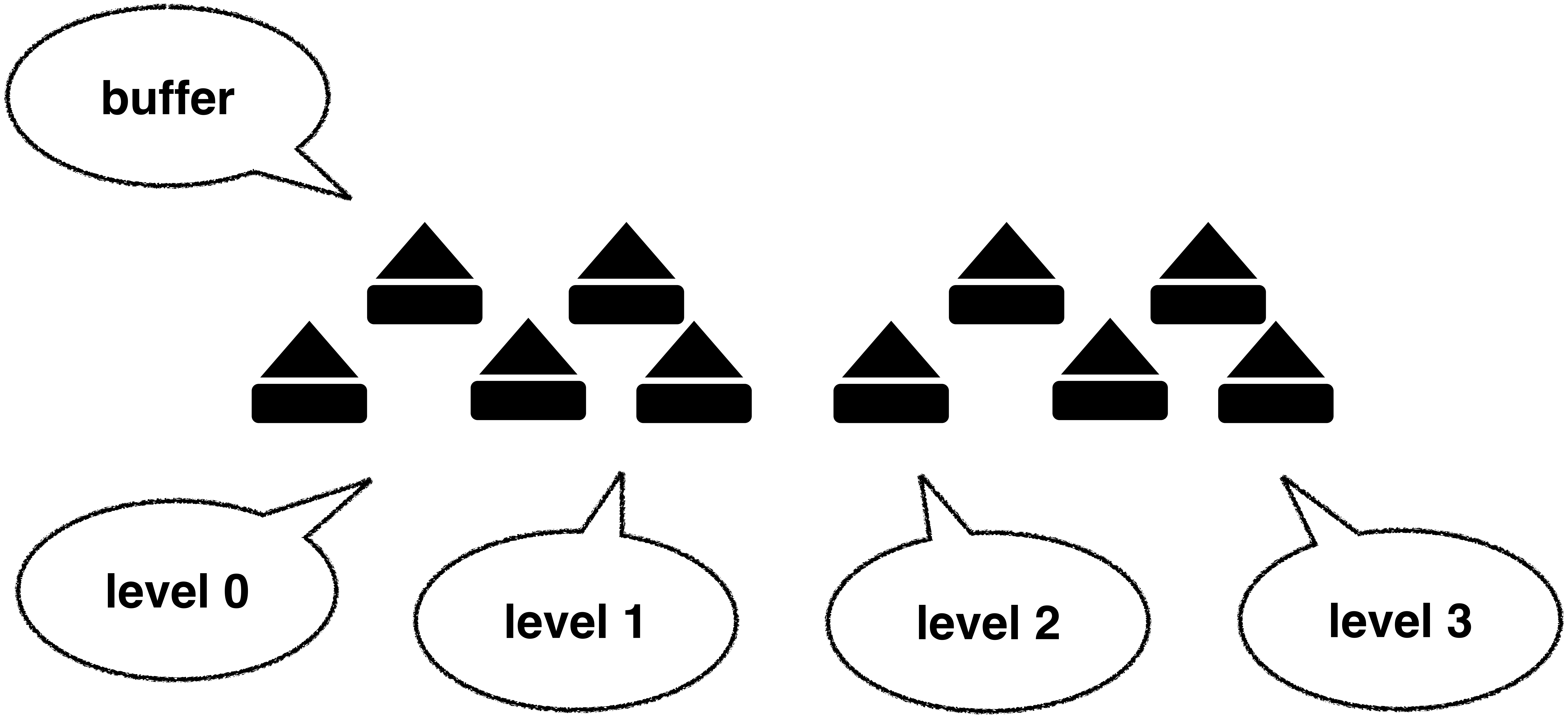


NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN

MACHINE LEARNING, SQL, CRYPTO, SCIENCE

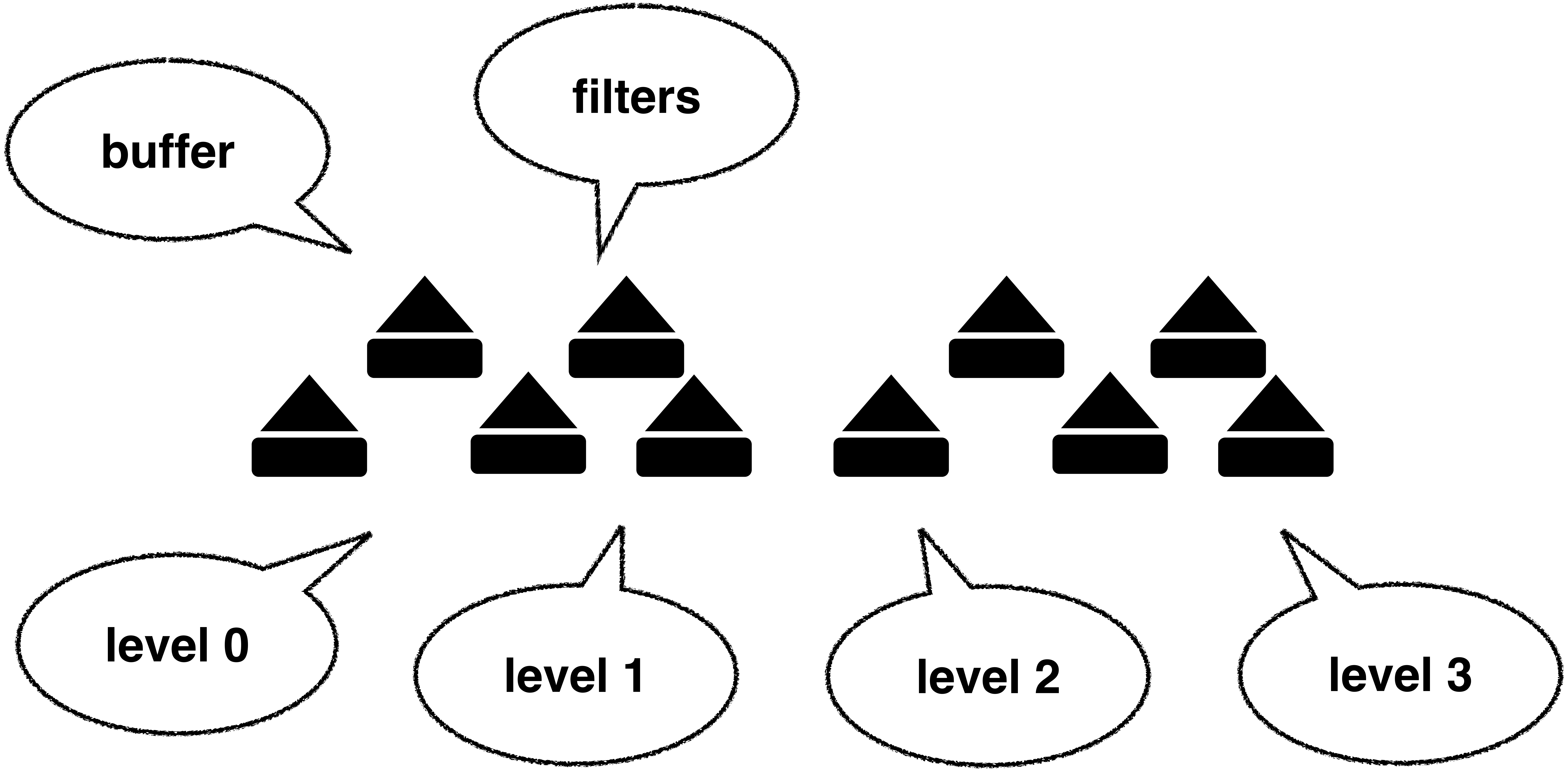


NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN

MACHINE LEARNING, SQL, CRYPTO, SCIENCE



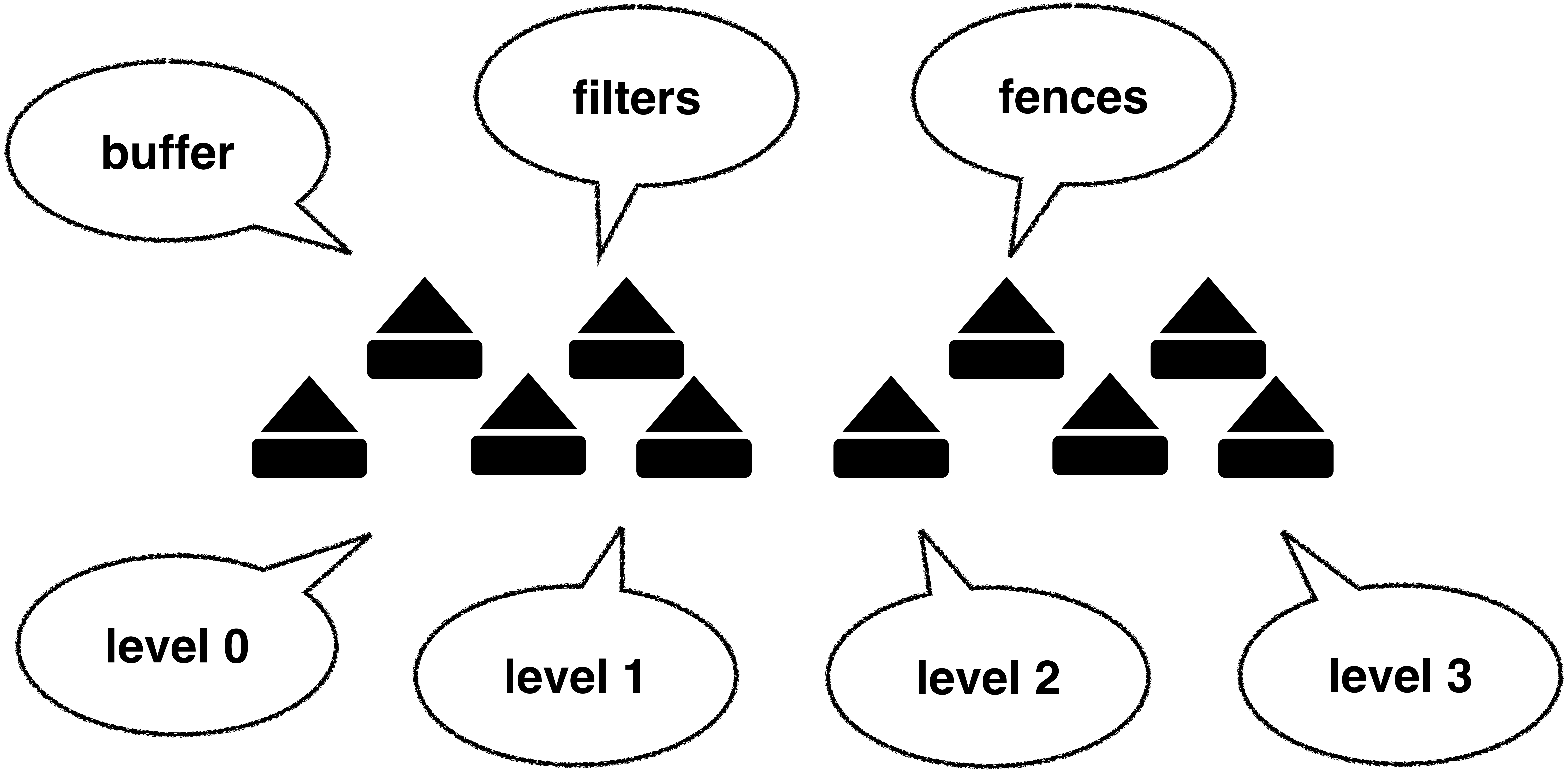
DASlab @ Harvard SEAS

NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN

MACHINE LEARNING, SQL, CRYPTO, SCIENCE



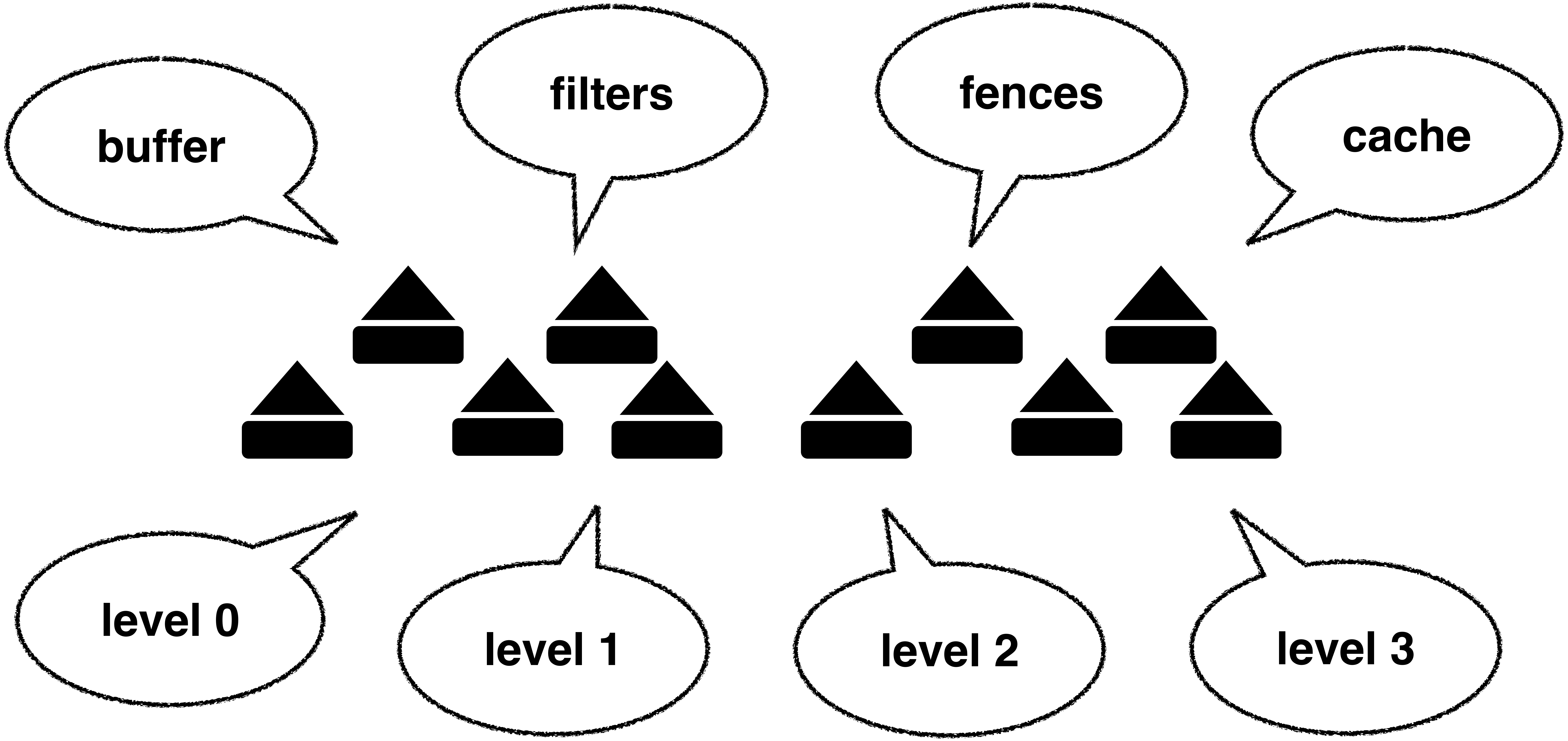
DASlab @ Harvard SEAS

NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN

MACHINE LEARNING, SQL, CRYPTO, SCIENCE



NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN

MACHINE LEARNING, SQL, CRYPTO, SCIENCE

DASlab @ Harvard SEAS

NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN
MACHINE LEARNING, SQL, CRYPTO, SCIENCE

DASlab @ Harvard SEAS

NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN

MACHINE LEARNING, SQL, CRYPTO, SCIENCE

buffer

filters

fences

cache

diverse data structures

interactions

hardware

level 0

level 1

level 2

level 3

DASlab @ Harvard SEAS

NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN

MACHINE LEARNING, SQL, CRYPTO, SCIENCE

NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN

MACHINE LEARNING, SQL, CRYPTO, SCIENCE

buffer

filters

fences

cache

diverse data structures | interactions | hardware | parallelism | robustness cloud cost SLAs

level 0

level 1

level 2

level 3

DASlab @ Harvard SEAS

NoSQL systems are the backbone of the BigData and AI era

LSM-tree
KV-stores

FACEBOOK, AMAZON, GOOGLE, TWITTER, LINKEDIN

MACHINE LEARNING, SQL, CRYPTO, SCIENCE

buffer **filters** **fences** **cache**

diverse data structures	interactions	hardware	parallelism	robustness cloud cost SLAs

There exist numerous variations of NoSQL KV-stores

LSM-tree variants, B-trees (MongoDB), Hash-index (Microsoft)

diverse data structures

interactions

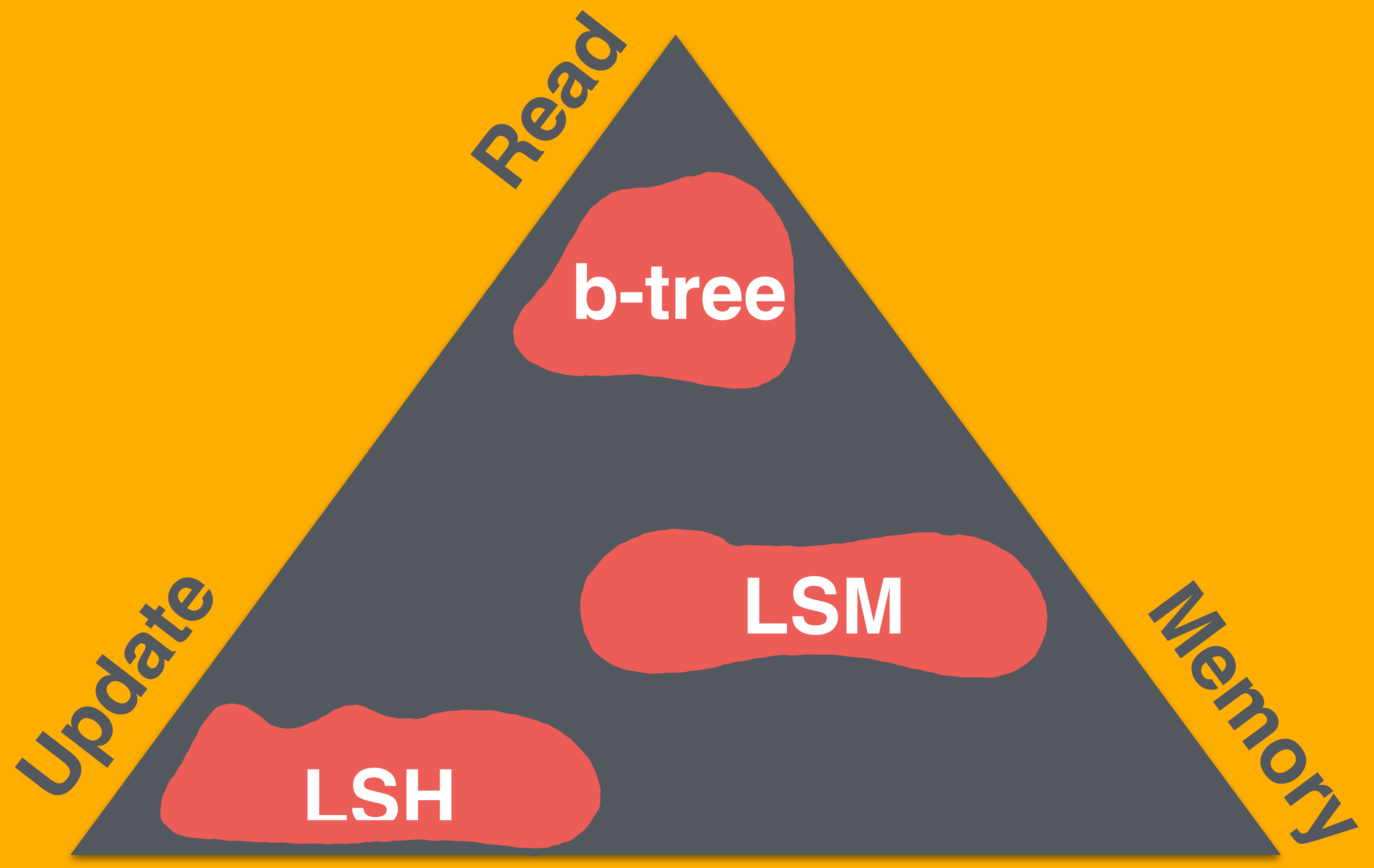
hardware

parallelism

robustness
cloud cost
SLAs

There exist numerous variations of NoSQL KV-stores

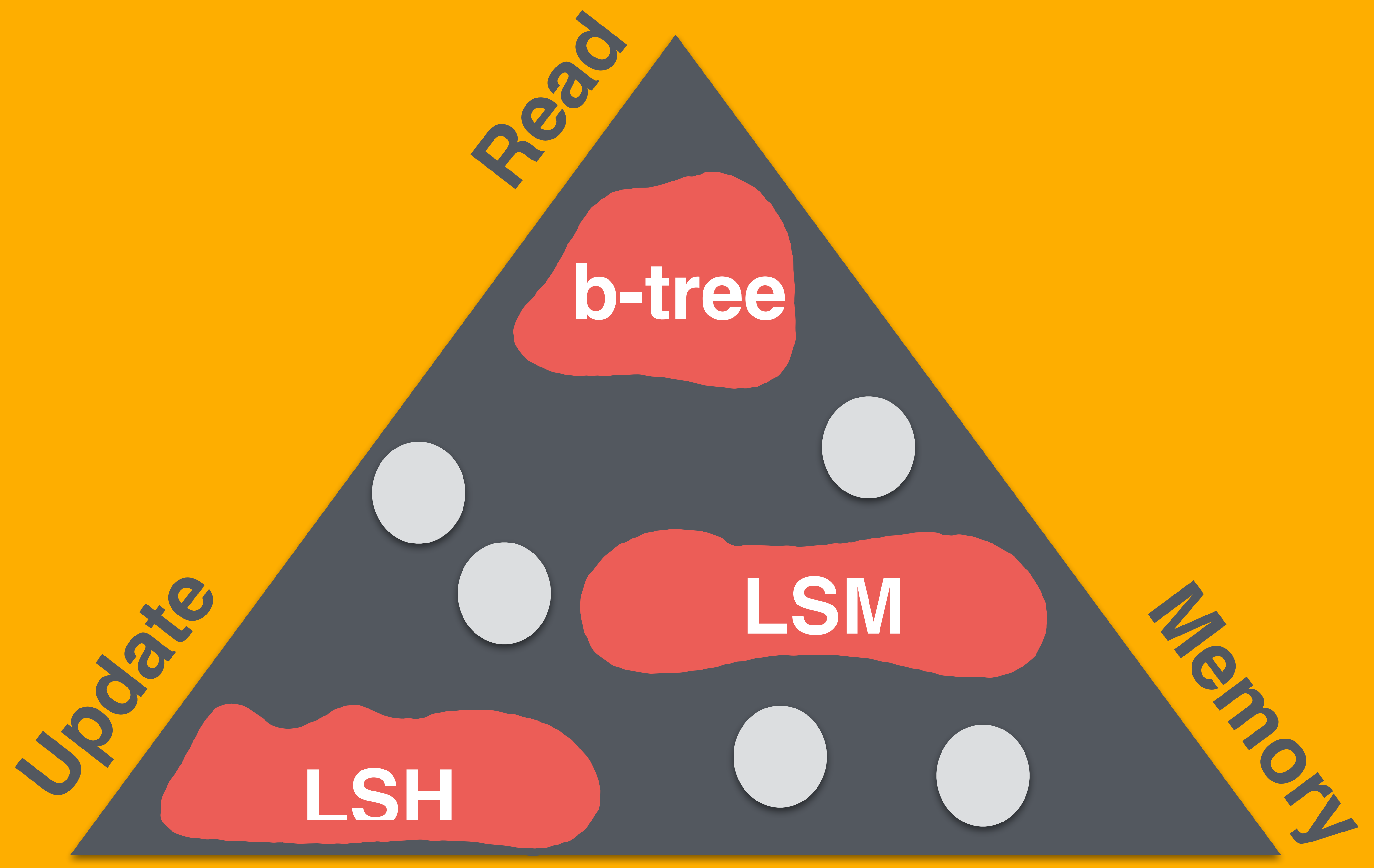
LSM-tree variants, B-trees (MongoDB), Hash-index (Microsoft)



diverse data structures	interactions	hardware	parallelism	robustness cloud cost SLAs

There exist numerous variations of NoSQL KV-stores

LSM-tree variants, B-trees (MongoDB), Hash-index (Microsoft)



Constant and increasing efforts for new system designs as applications & hardware change

diverse data structures	interactions	hardware	parallelism	robustness cloud cost SLAs

There exist numerous variations of NoSQL KV-stores

LSM-tree variants, B-trees (MongoDB), Hash-index (Microsoft)

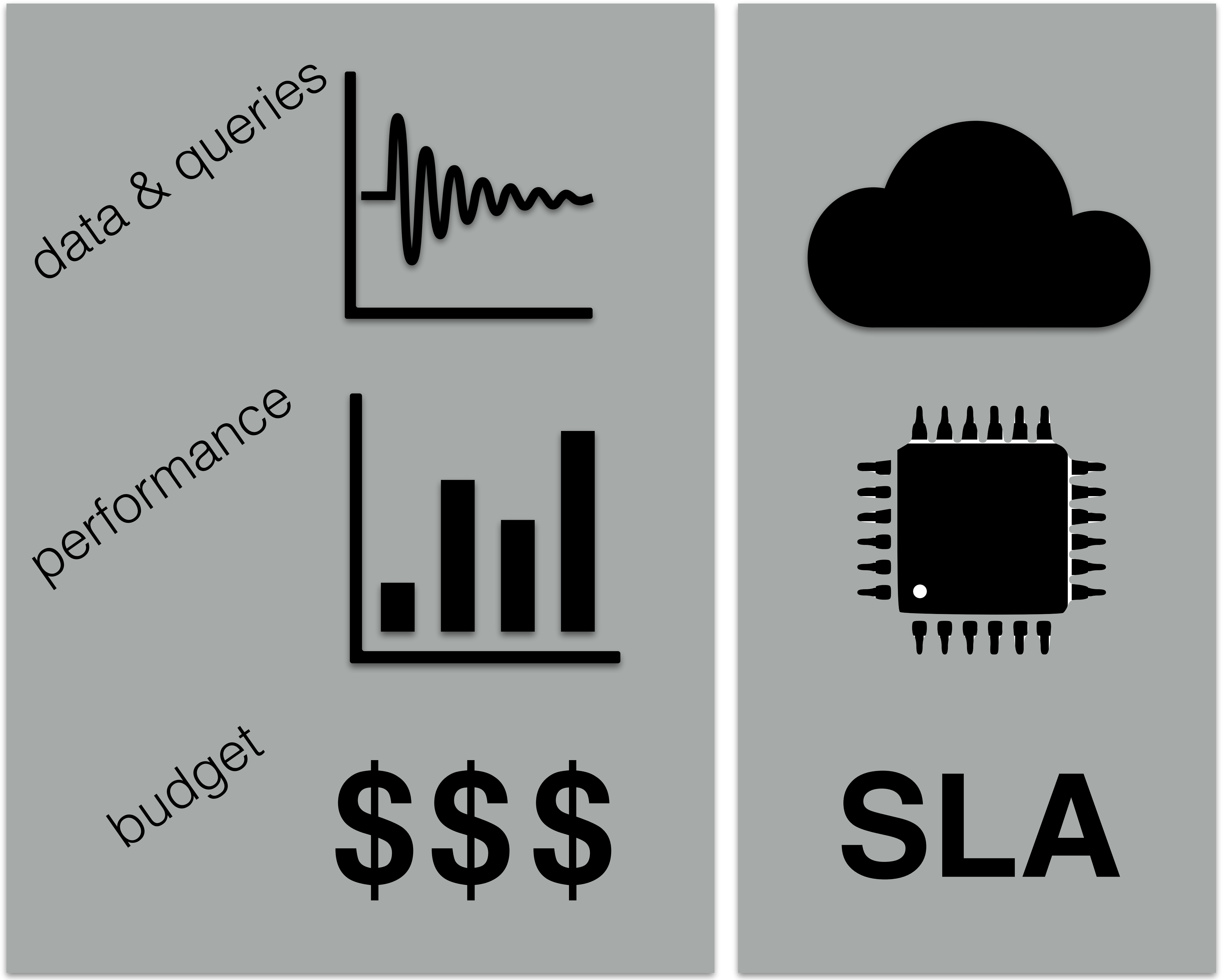
diverse data structures | interactions | hardware | parallelism | robustness cloud cost SLAs

Requirements/Goals

diverse data structures | interactions | hardware | parallelism | robustness cloud cost SLAs

Requirements/Goals

Context

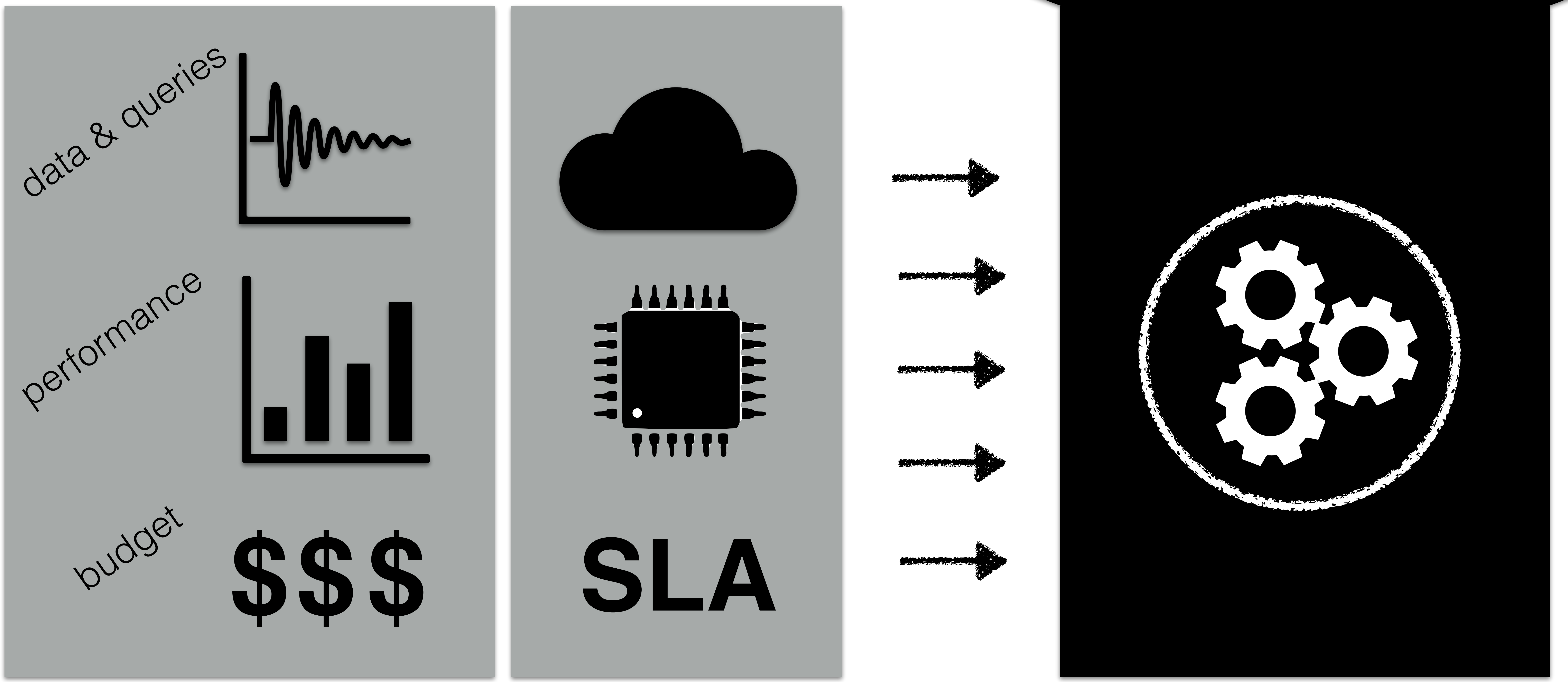


DASlab @ Harvard SEAS

diverse data structures | interactions | hardware | parallelism | robustness cloud cost SLAs

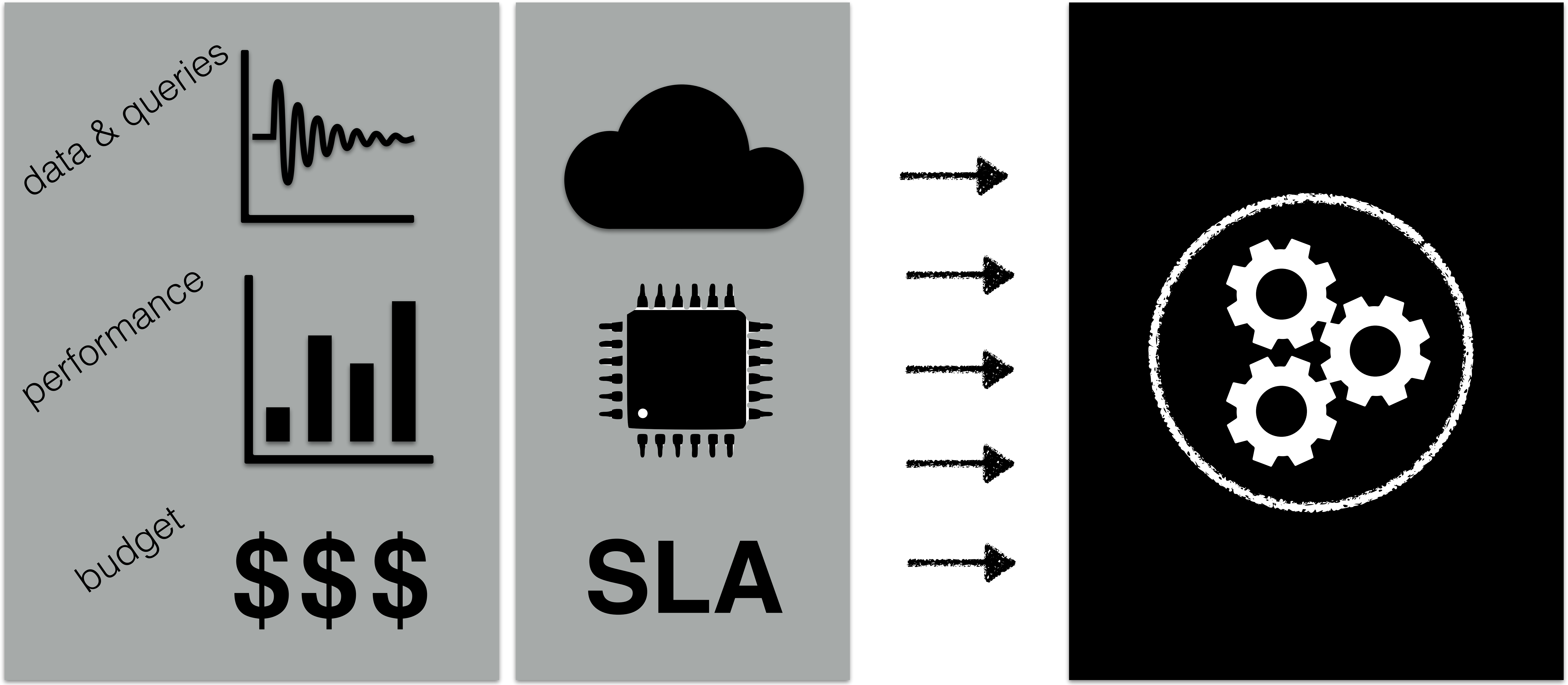
Requirements/Goals

Context



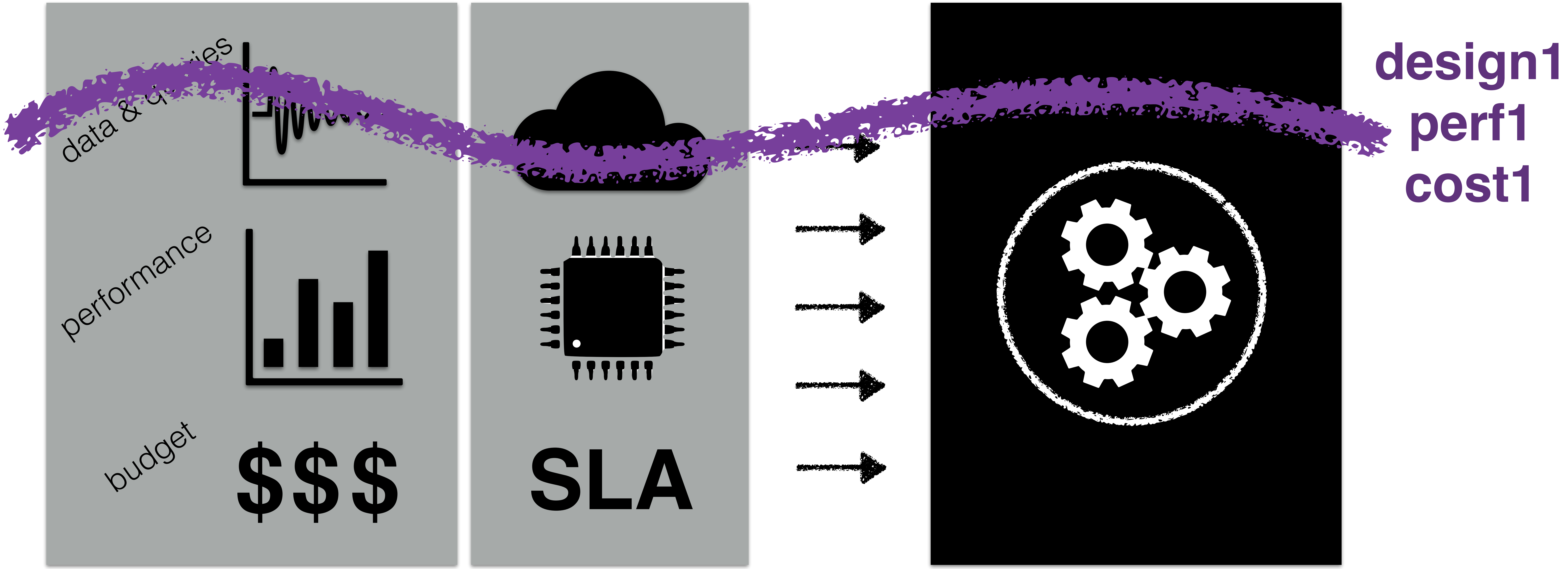
DASlab @ Harvard SEAS


diverse
data structures
interactions
hardware
parallelism
robustness
cloud cost
SLAs
Requirements/Goals
Context
data & queries
performance
budget
$$$
SLA
best
DATA
SYSTEM
design & code
DASlab
@ Harvard SEAS



what-if reasoning

DASlab
@ Harvard SEAS



what-if reasoning

DASlab @ Harvard SEAS

what-if reasoning

AUTO 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*]

DASlab @ Harvard SEAS

IS THERE A CALCULUS OF DATA SYSTEMS?

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*] the **grammar** of data systems design

Nikos Kazantzakis, philosopher

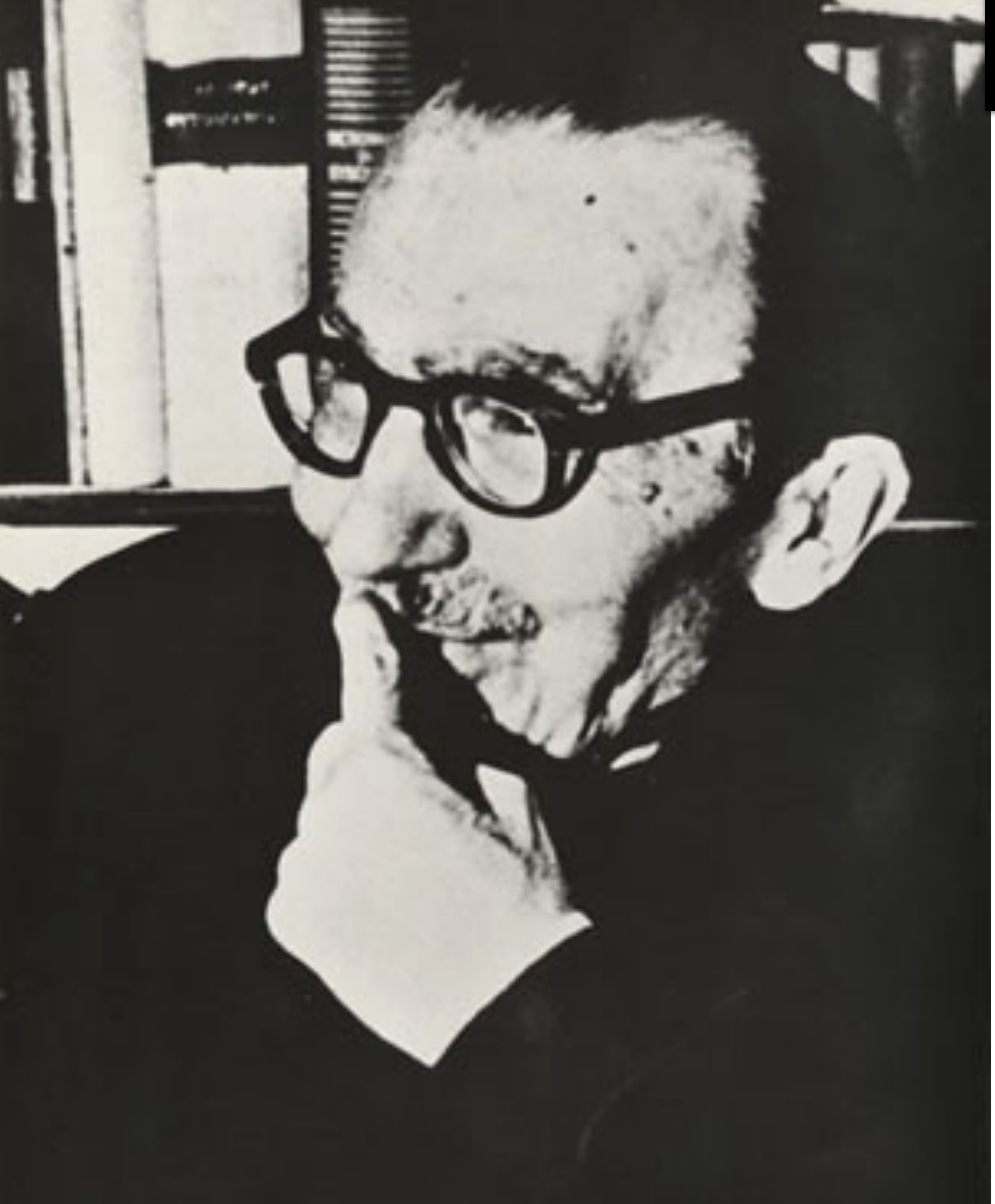
action is for nothing holy

hope the most of

fear form

am free

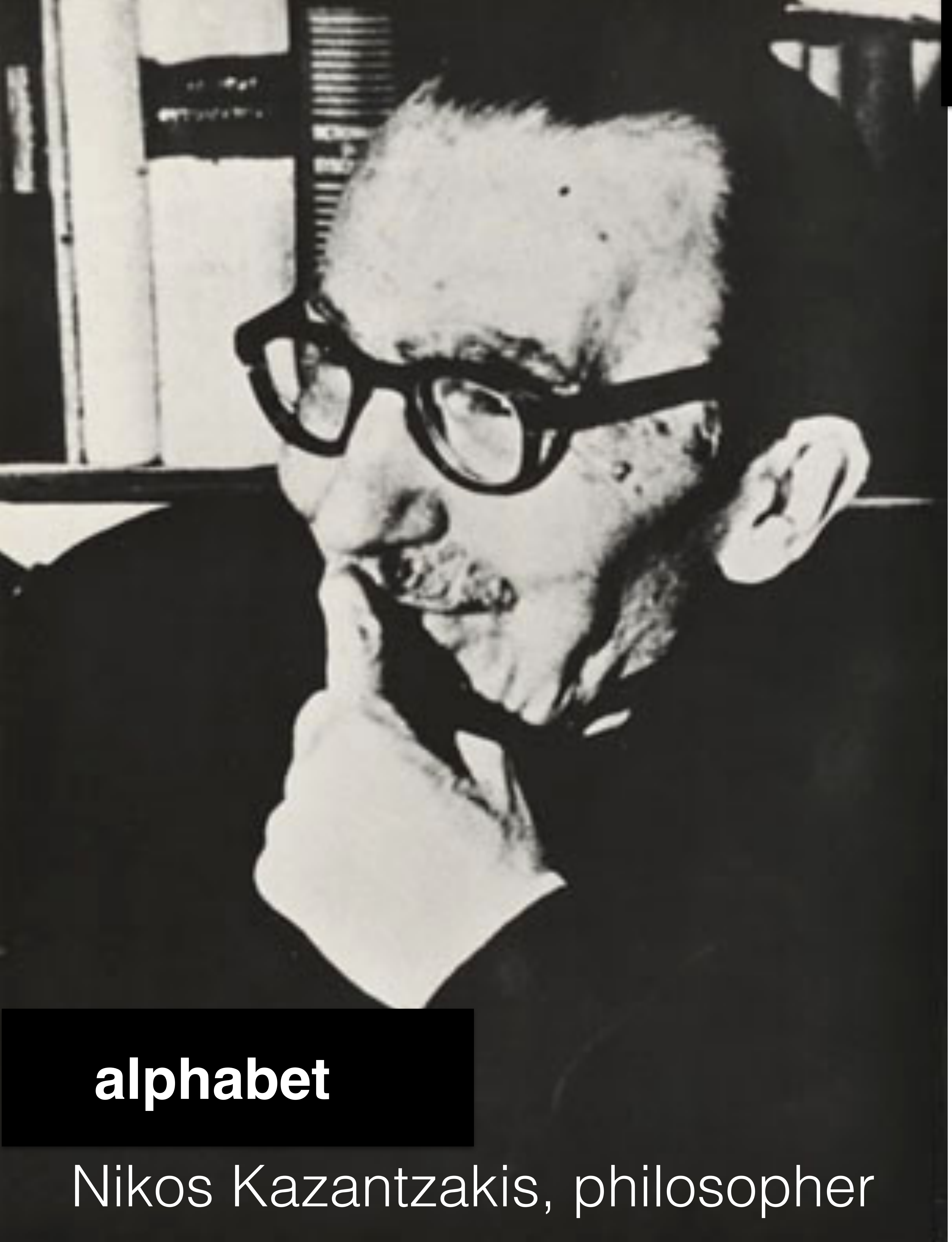
ultimate I theory



Nikos Kazantzakis, philosopher

action is the ultimate most holy form of theory

I hope for nothing
I fear nothing
I am free



alphabet

Nikos Kazantzakis, philosopher

action is

the most holy of

form

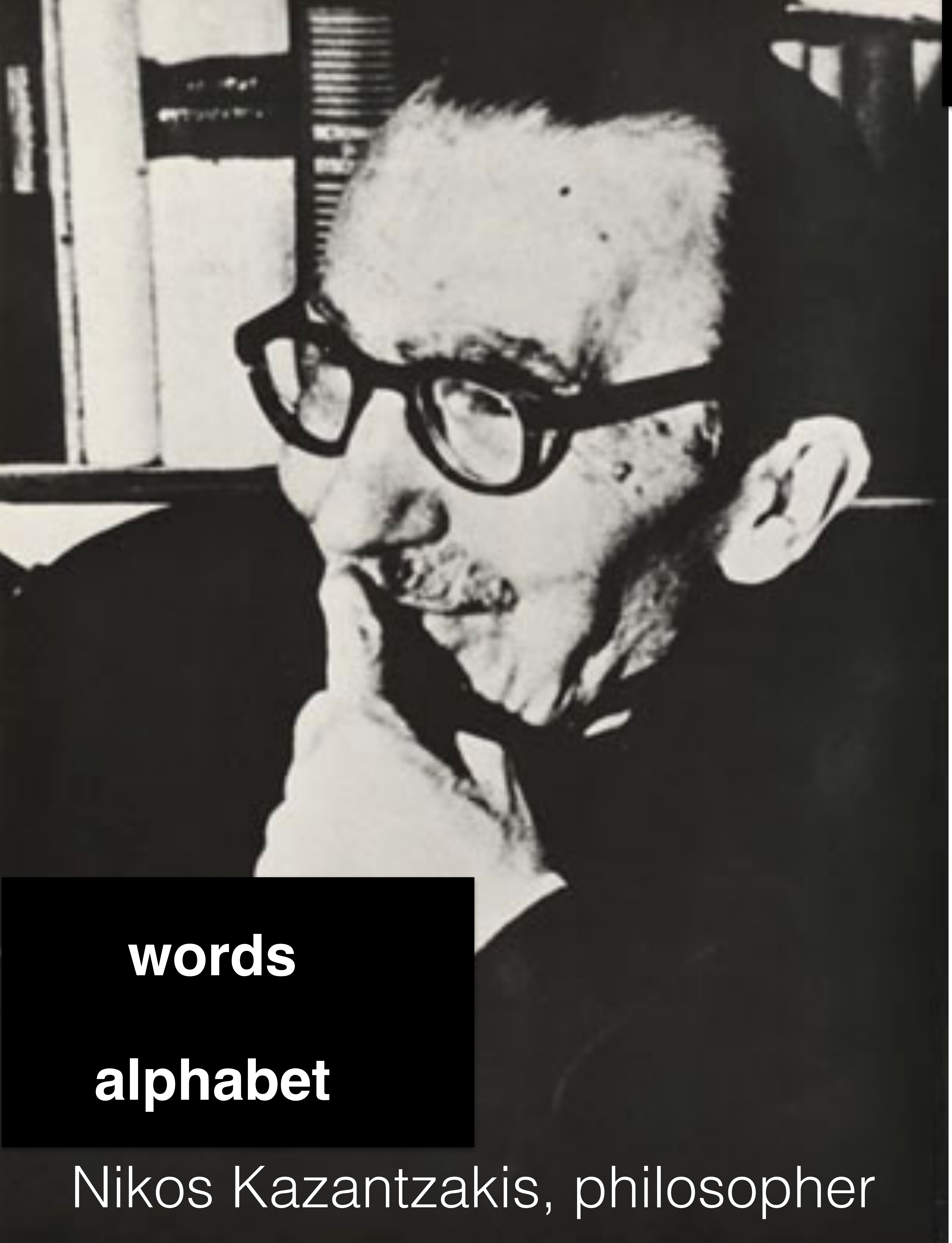
ultimate theory

I hope for nothing

I fear nothing

I am free

the **grammar** of data systems design



Nikos Kazantzakis, philosopher

action is

the most holy

of

form

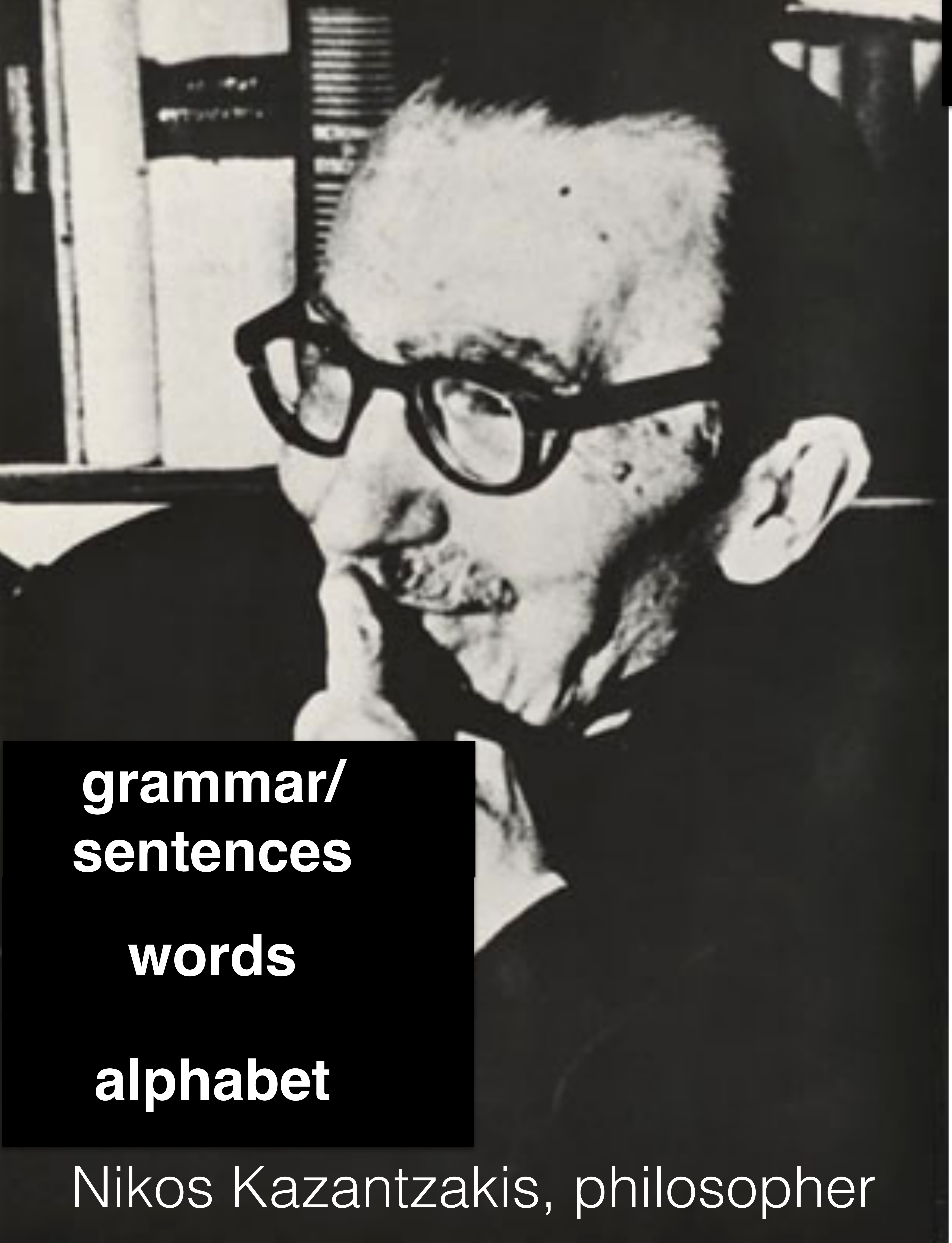
ultimate theory

I hope for nothing

I fear nothing

I am free

the **grammar** of data systems design



Nikos Kazantzakis, philosopher

action is

the most holy

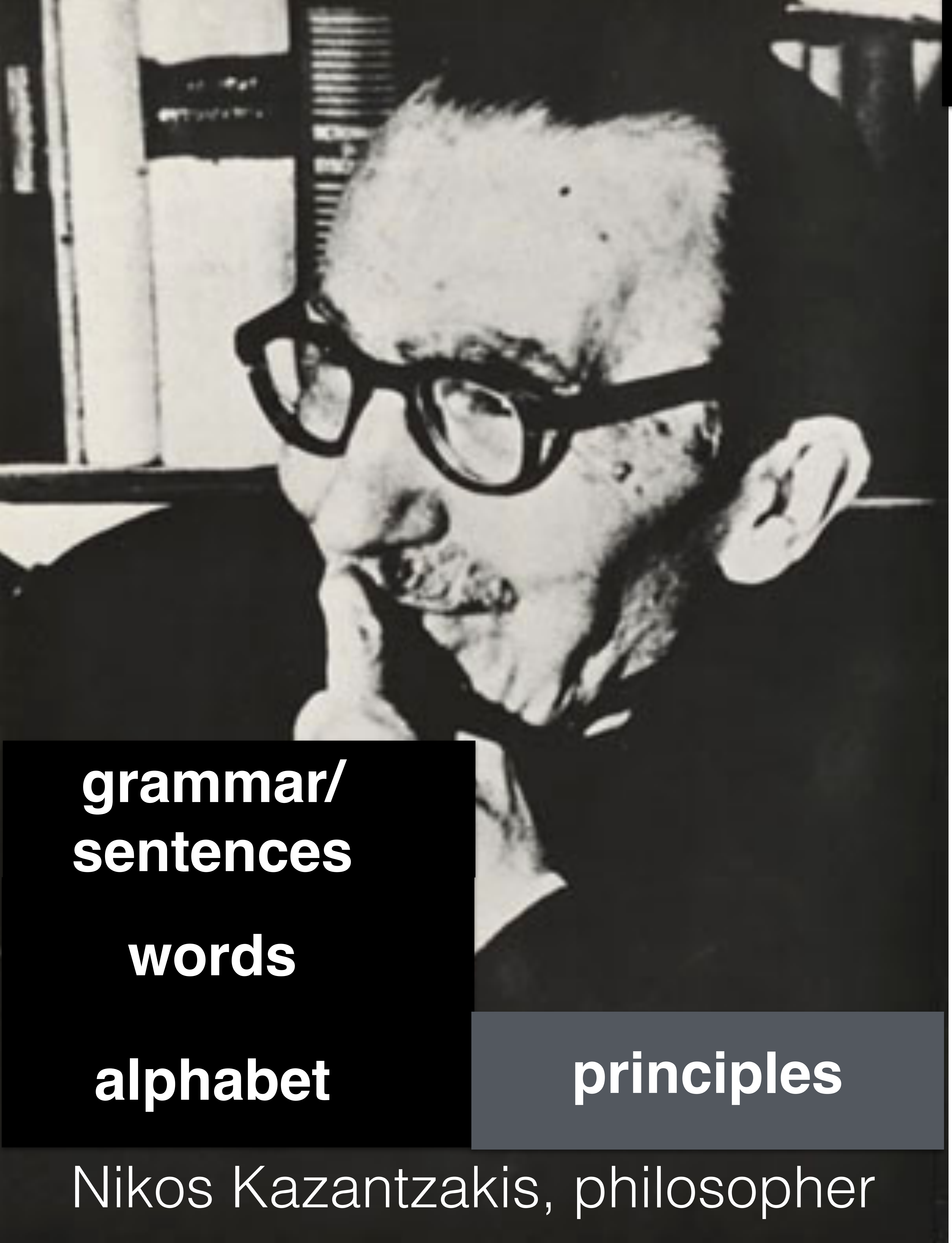
form of

ultimate theory

I hope for nothing

I fear nothing

I am free



grammar/ sentences

words

alphabet

principles

Nikos Kazantzakis, philosopher

action is

the most holy

form of

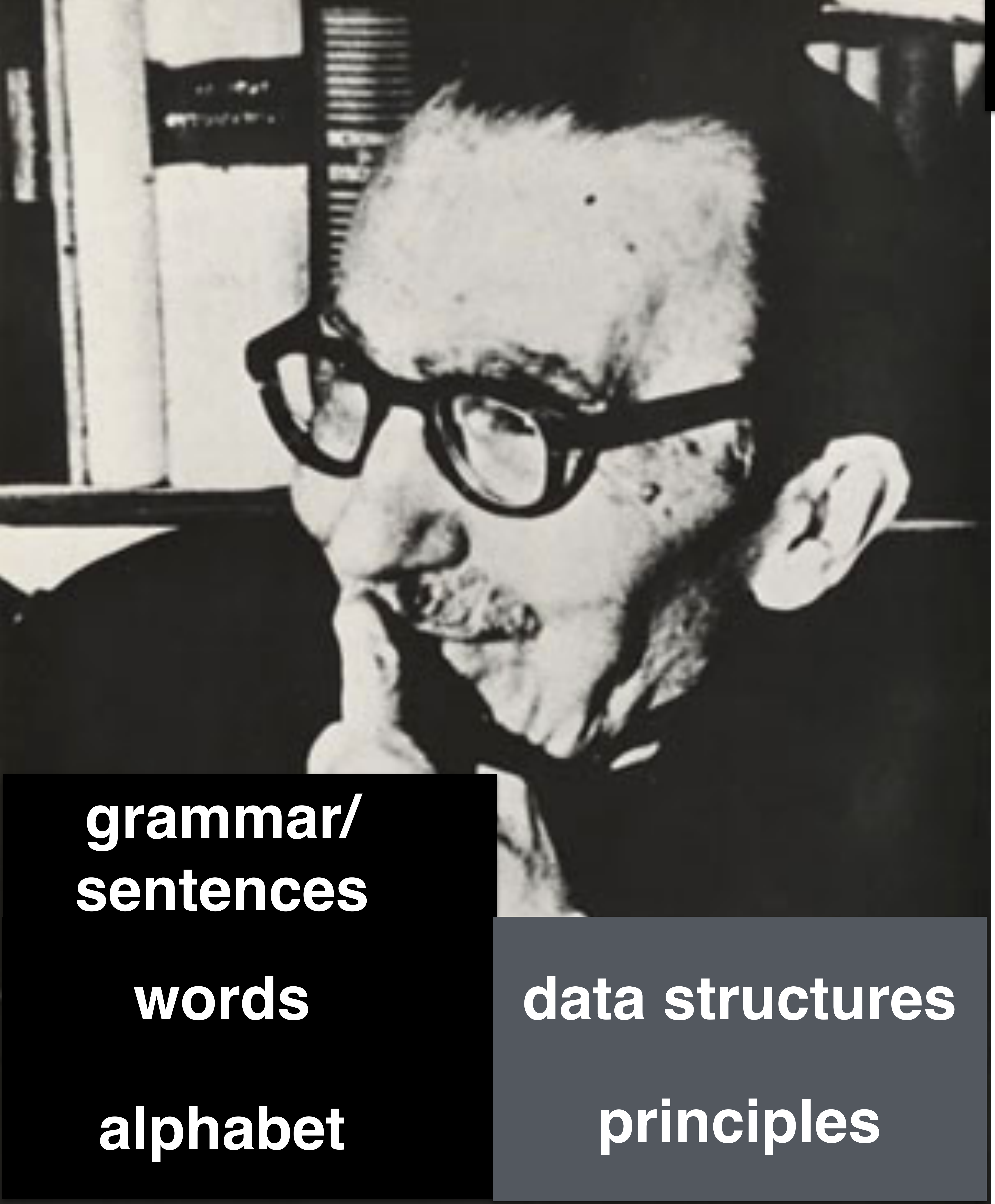
ultimate theory

I hope for nothing

I fear nothing

I am free

the **grammar** of data systems design



Nikos Kazantzakis, philosopher

action is
the most holy
form of
ultimate theory

I hope for nothing
I fear nothing
I am free

the **grammar** of data systems design

grammar/ sentences	**interactions**
words	**data structures**
alphabet	**principles**

Nikos Kazantzakis, philosopher

action is
the most holy
form of
ultimate theory

I hope for nothing
I fear nothing
I am free

the **grammar** of data systems design

grammar/ sentences	**interactions**
words	**data structures**
alphabet	**principles**

Nikos Kazantzakis, philosopher

action is
the most holy
of
form
ultimate theory

NEW

I hope for nothing
I fear nothing
I am free

the **grammar** of data systems design

grammar/ sentences	**interactions**
words	**data structures**
alphabet	**principles**

Nikos Kazantzakis, philosopher

action is
the most holy of
form
theory

which are “all”
possible *data systems*
we may ever invent?

I hope for nothing
I fear nothing
I am free

Trillions of possible data structures

Data Calculator @SIGMOD 2018

Trillions of possible data structures

Data Calculator @SIGMOD 2018

New NoSQL systems: 1000x faster

Cosine @PVLDB 2022 and Limousine @SIGMOD 2024

Trillions of possible data structures

Data Calculator @SIGMOD 2018

New NoSQL systems: 1000x faster

Cosine @PVLDB 2022 and Limousine @SIGMOD 2024

Synthesized statistics, 10x faster ML

Data Canopy @SIGMOD 2017

Trillions of possible data structures

Data Calculator @SIGMOD 2018

New NoSQL systems: 1000x faster

Cosine @PVLDB 2022 and Limousine @SIGMOD 2024

Synthesized statistics, 10x faster ML

Data Canopy @SIGMOD 2017

10x faster Neural Networks

MotherNets @MLSys 2020, and M2 @MLSys 2023

Trillions of possible data structures

Data Calculator @SIGMOD 2018

New NoSQL systems: 1000x faster

Cosine @PVLDB 2022 and Limousine @SIGMOD 2024

Synthesized statistics, 10x faster ML

Data Canopy @SIGMOD 2017

10x faster Neural Networks

MotherNets @MLSys 2020, and M2 @MLSys 2023

10x faster Image AI

Image Calculator, SIGMOD 2024

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

Check out: syllabus, preparation readings, project 0, systems project 1, online sections

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

Here is my data and inference requests.
Design and implement and implement an LLM for my budget?

Nvidia released a new GPU.
Should we invest in the new hardware for our cluster of Image AI systems?

We are preparing to release a new feature for our social network application.
Should we redesign and reimplement our underlying key-value store?

….

CS 265

Stratos Idreos

BIG DATA SYSTEMS

NoSQL | Neural Networks | Image AI | LLMs | Data Science