

Chucky: A Succinct Cuckoo Filter for LSM-Tree

SIGMOD 2021
Short Talk

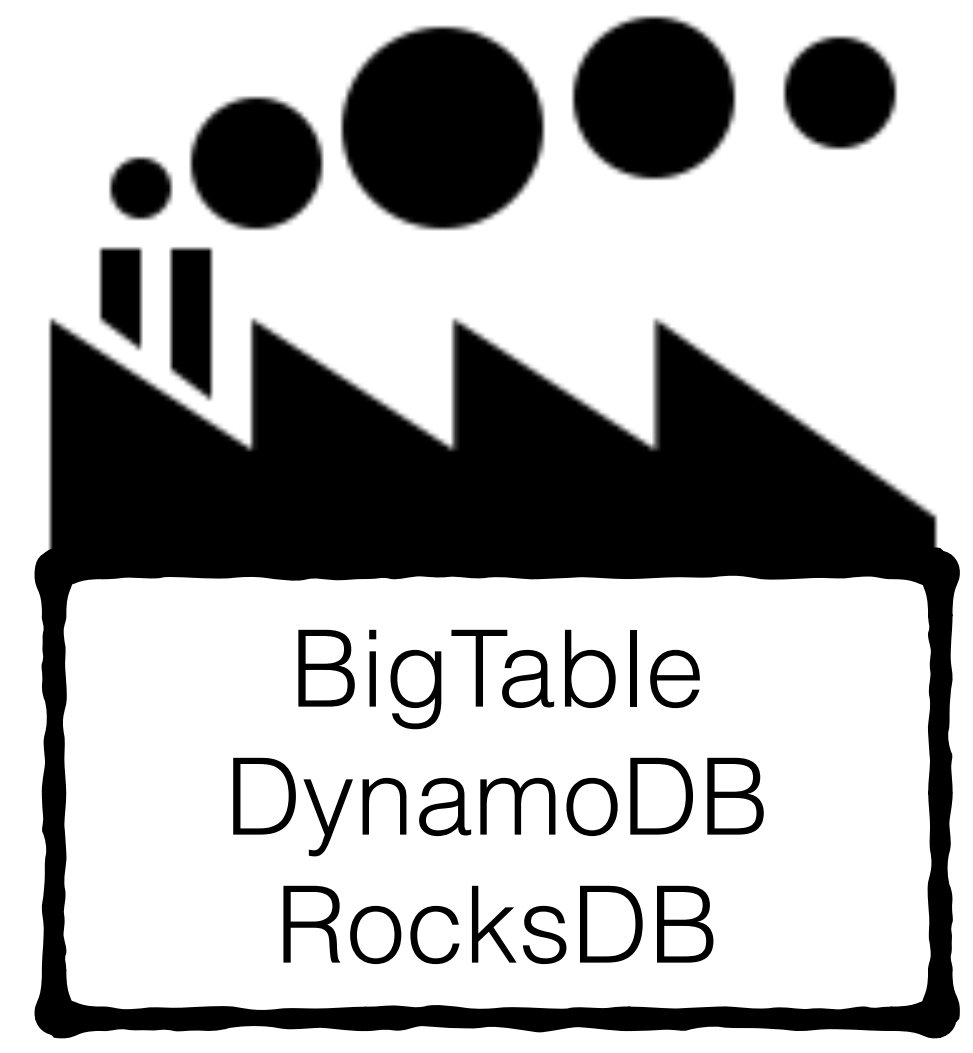
Niv Dayan, Moshe Twitto



PLiOPS
EXTREME DATA PROCESSOR



LSM-Tree

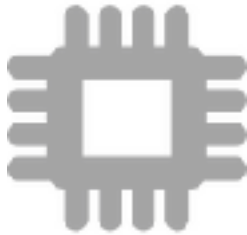


BigTable
DynamoDB
RocksDB

key-value pairs



buffer



level 1 →

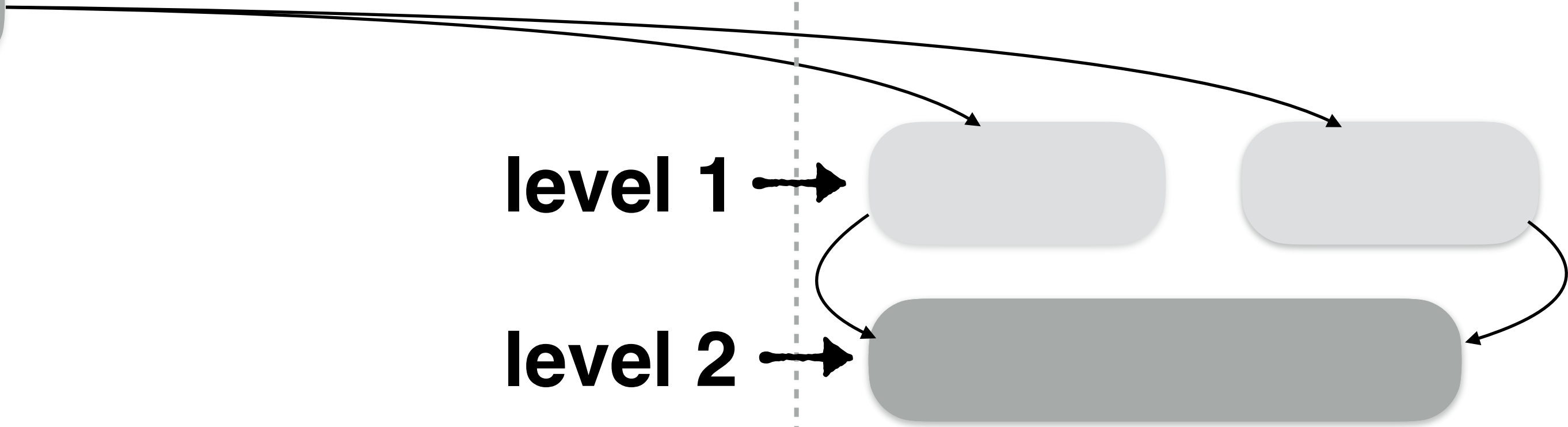


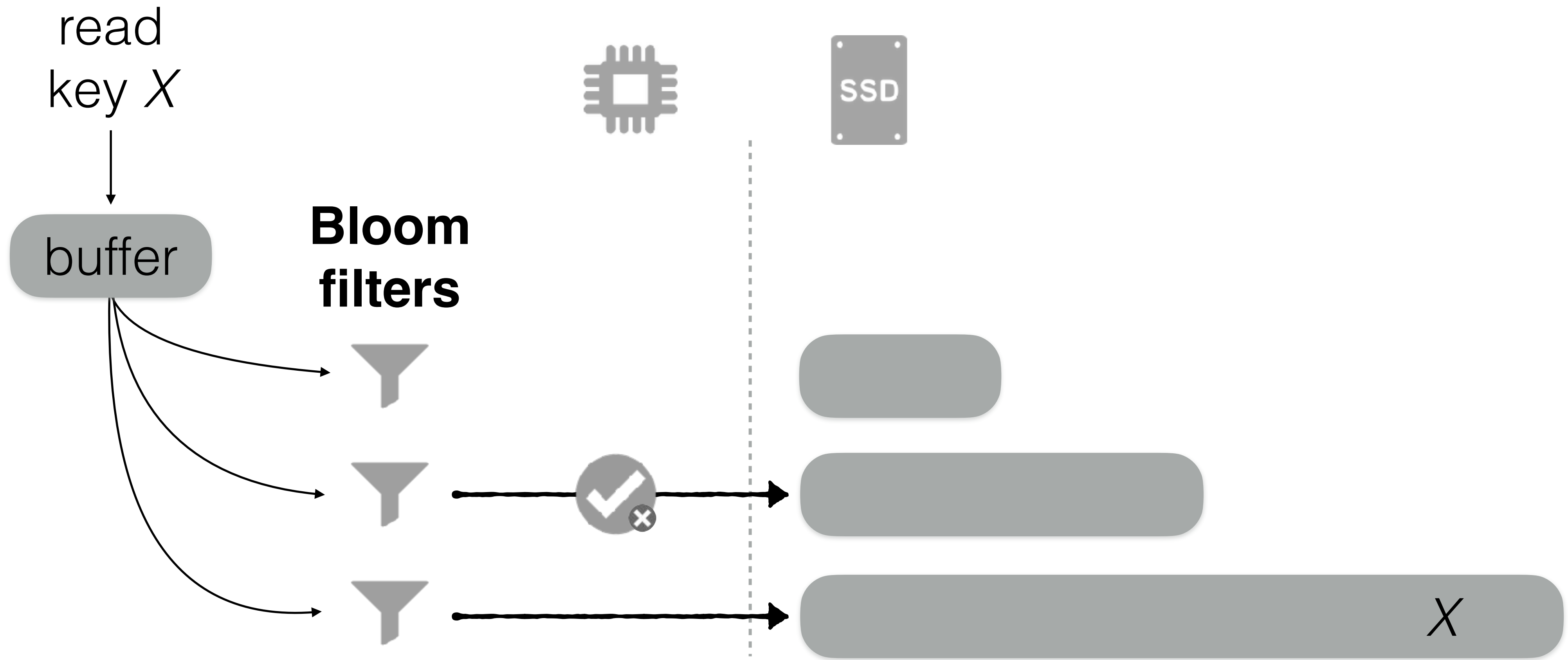
sort-merge

level 2 →

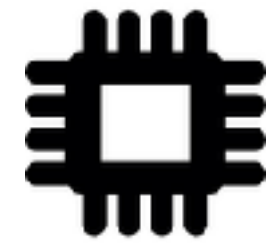


level 3 →





≈ 100 ns



≈ 10 μs

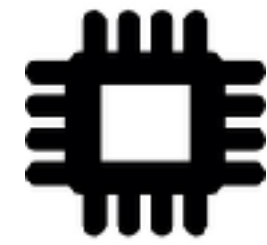


(optane)





$\approx 100 \text{ ns}$



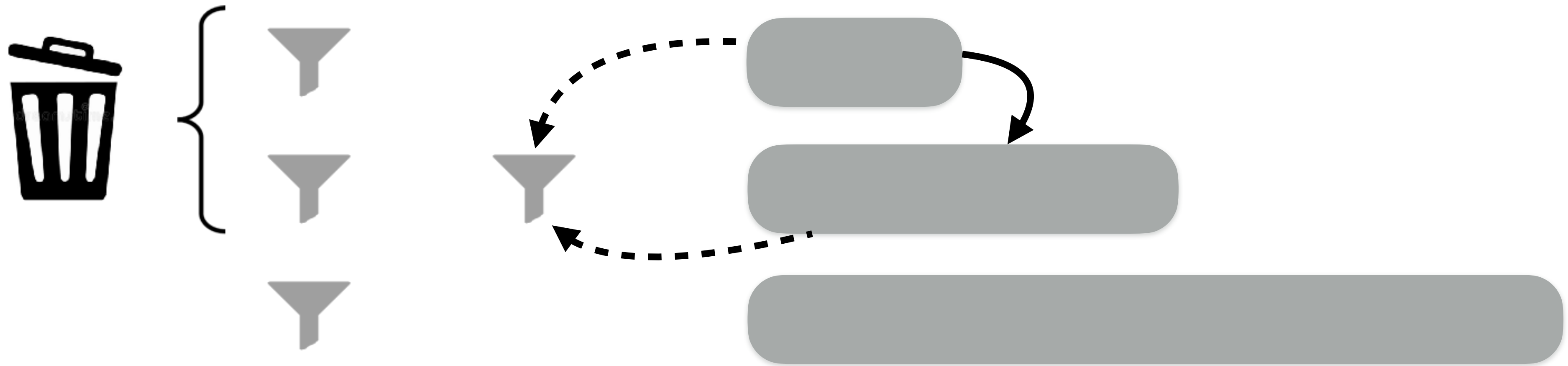
$\approx 10 \mu\text{s}$



(optane)

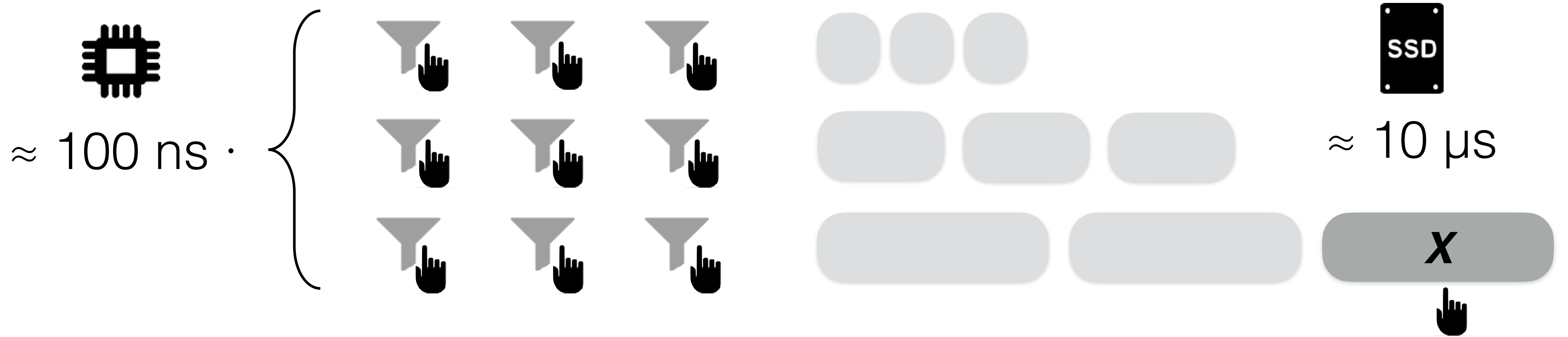


Problem 1: **reconstruction cost**

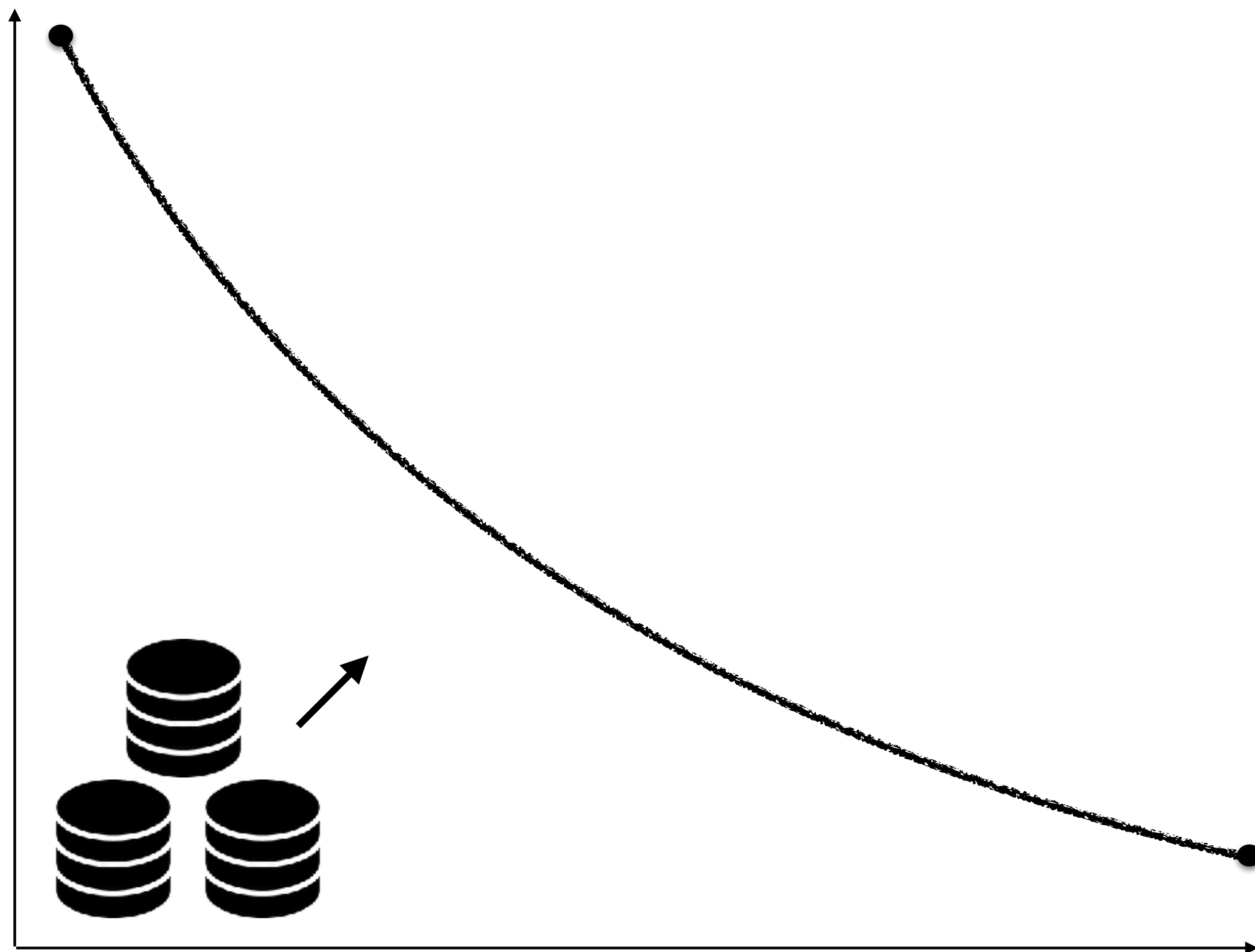


Problem 1: reconstruction cost

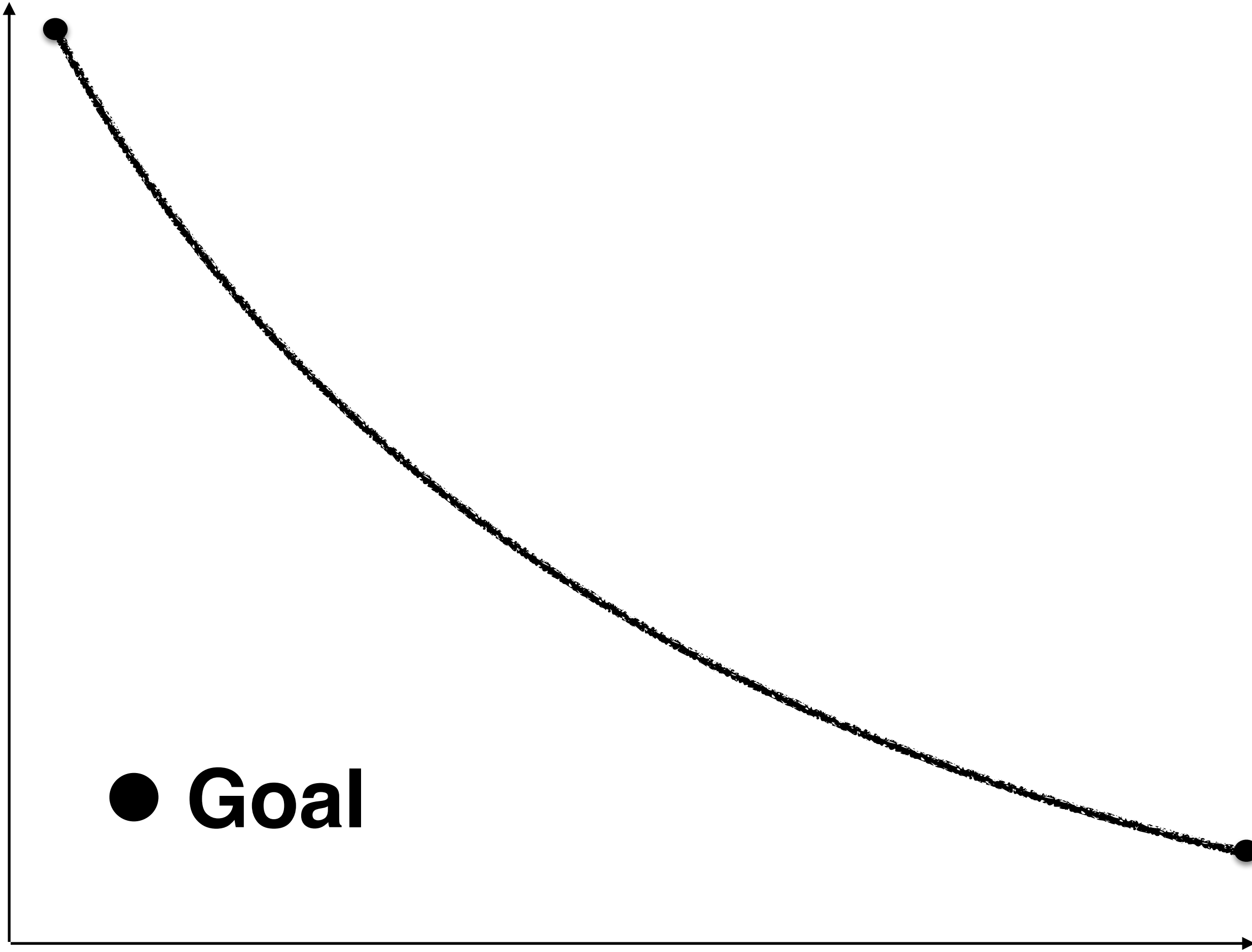
Problem 2: **query cost**

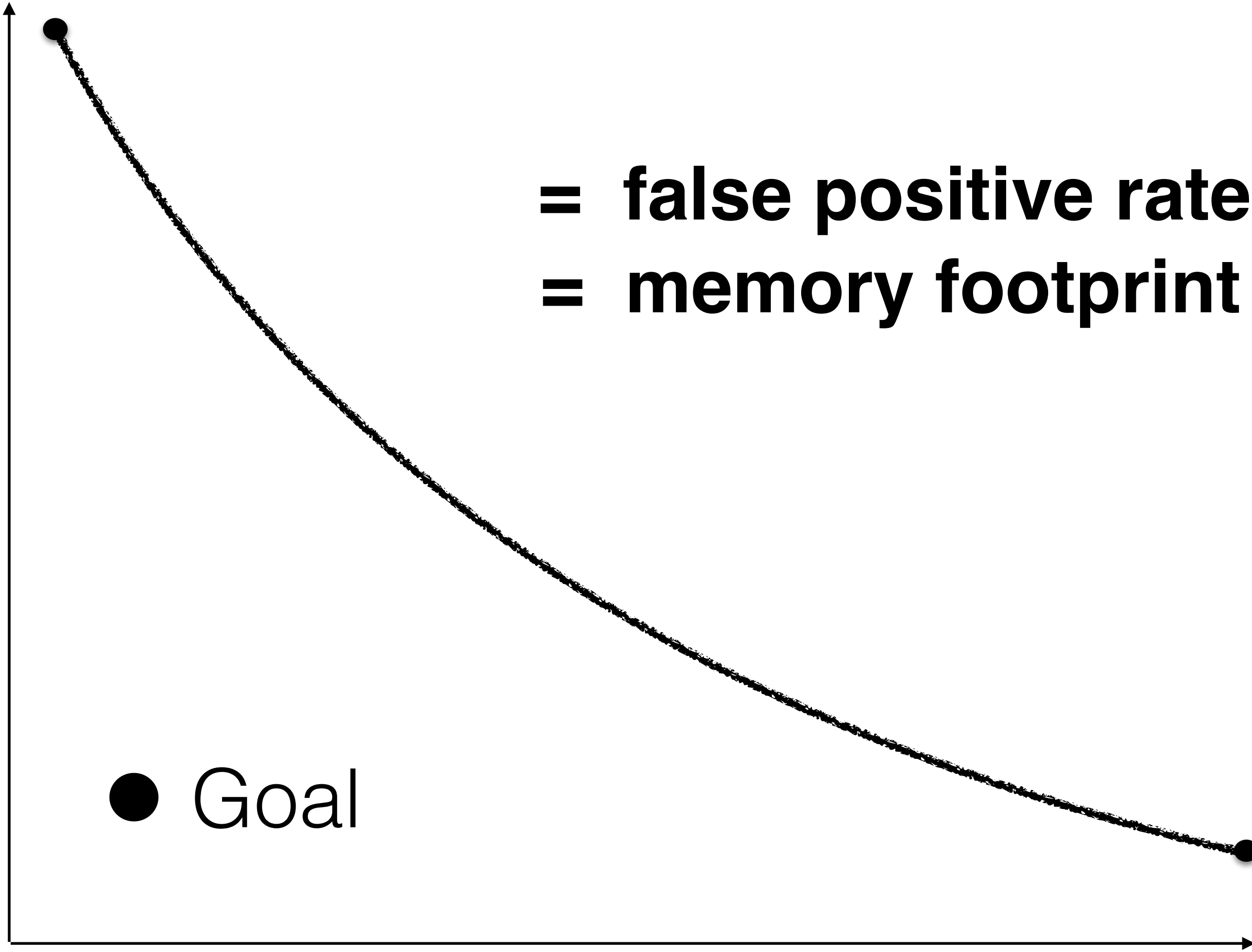


query
cost



reconstruction cost



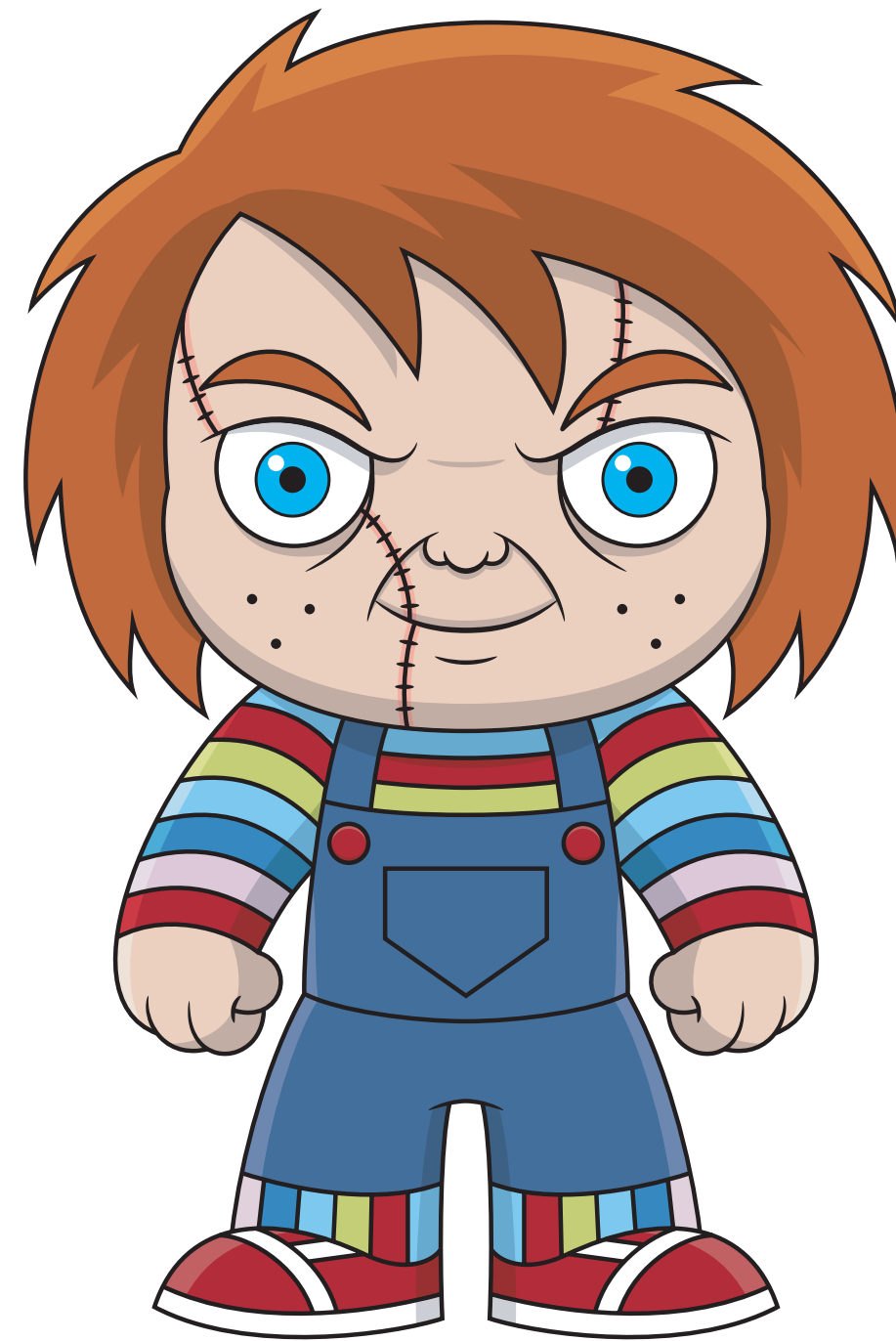


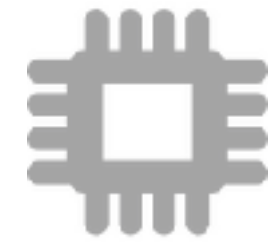
= false positive rate
= memory footprint

● Goal



Chucky: huffman coded key-value store





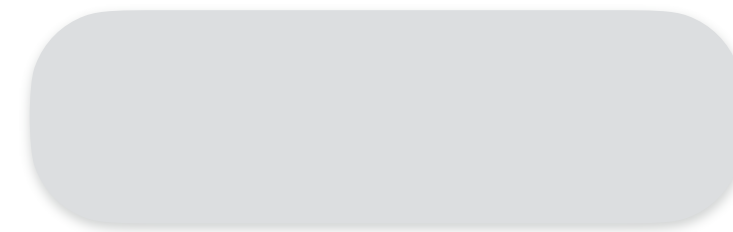
hash table

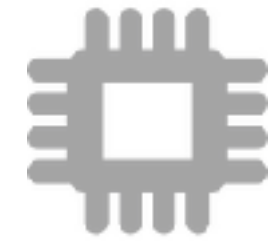


key

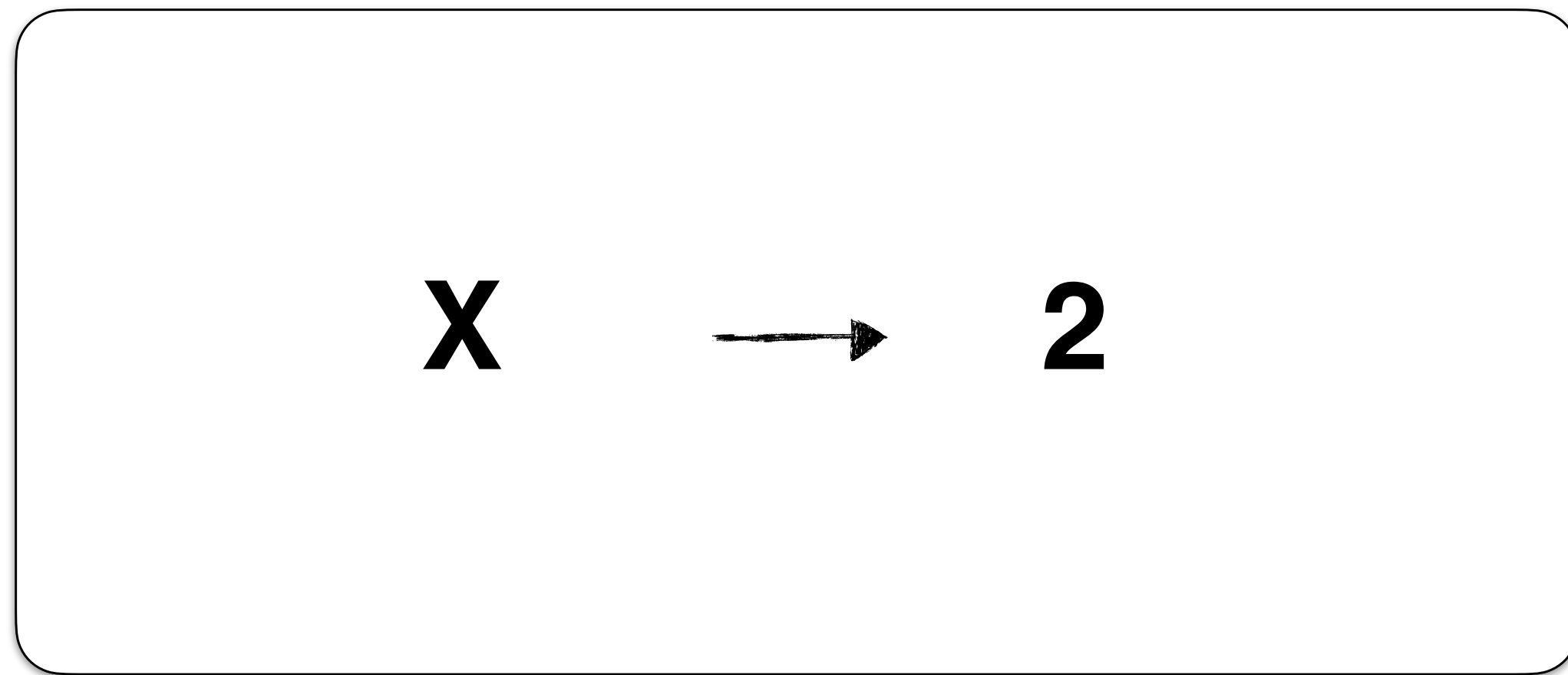


level

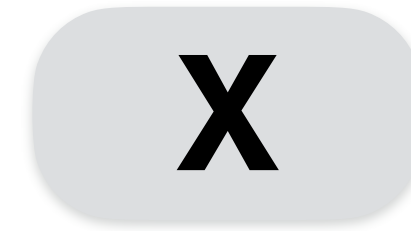




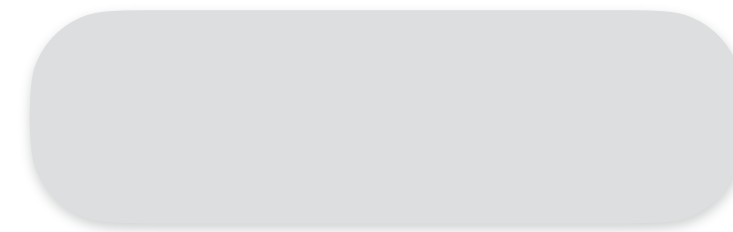
level ID



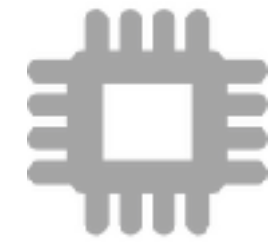
1



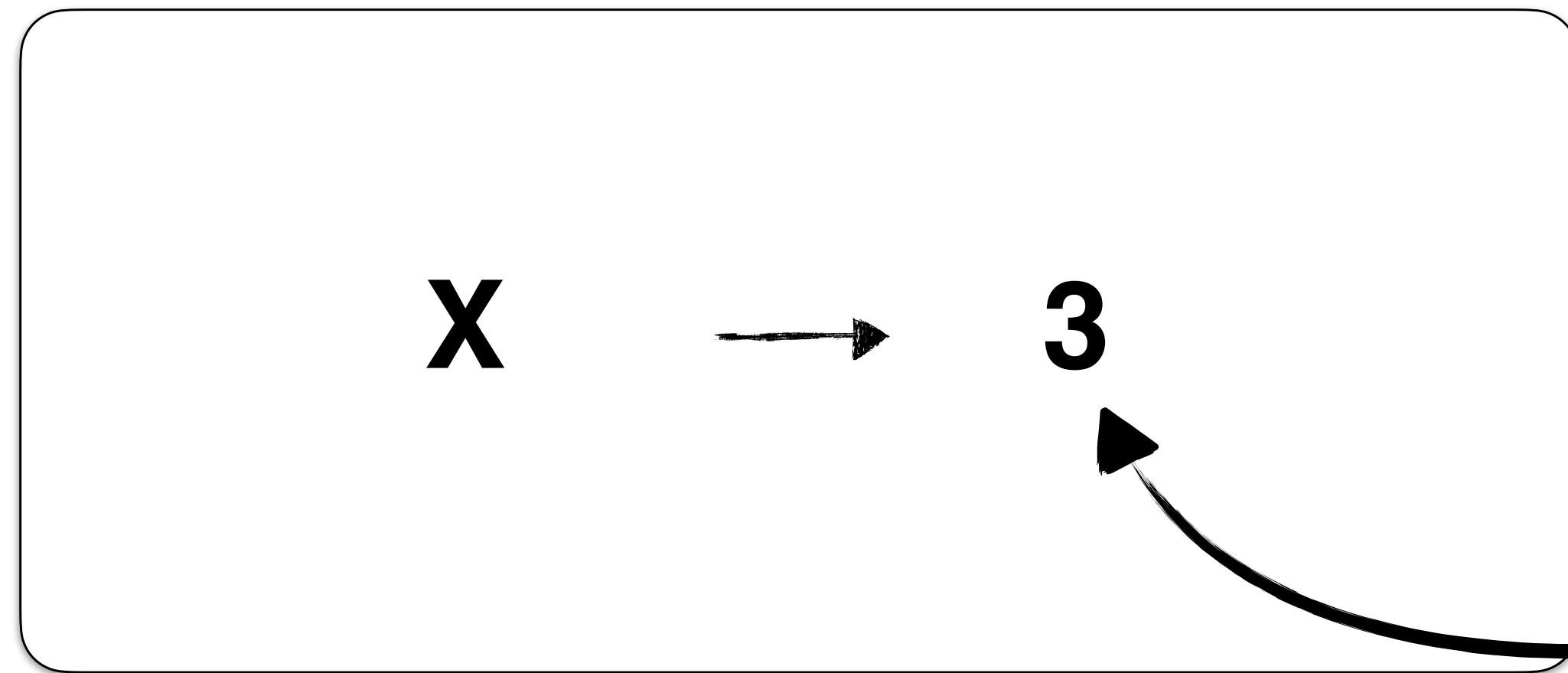
2



3



level ID



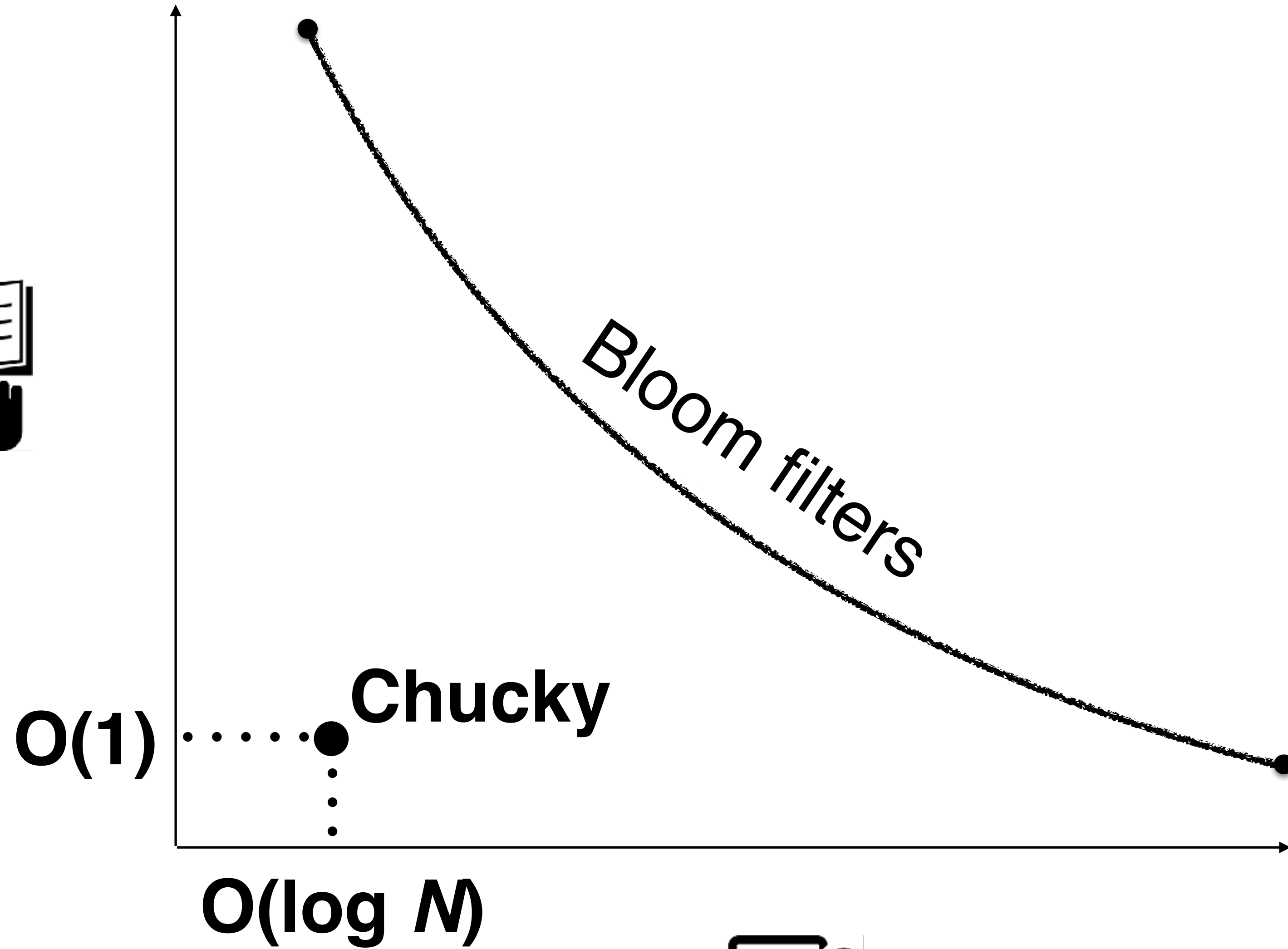
1

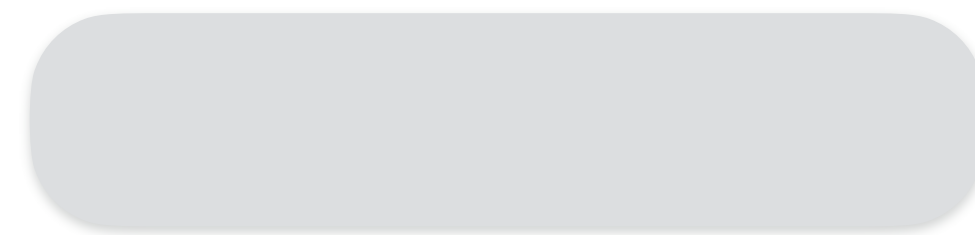
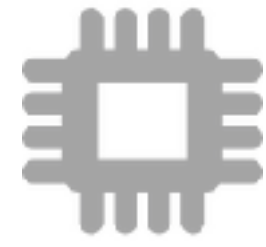
2

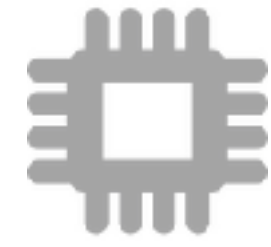



3

update

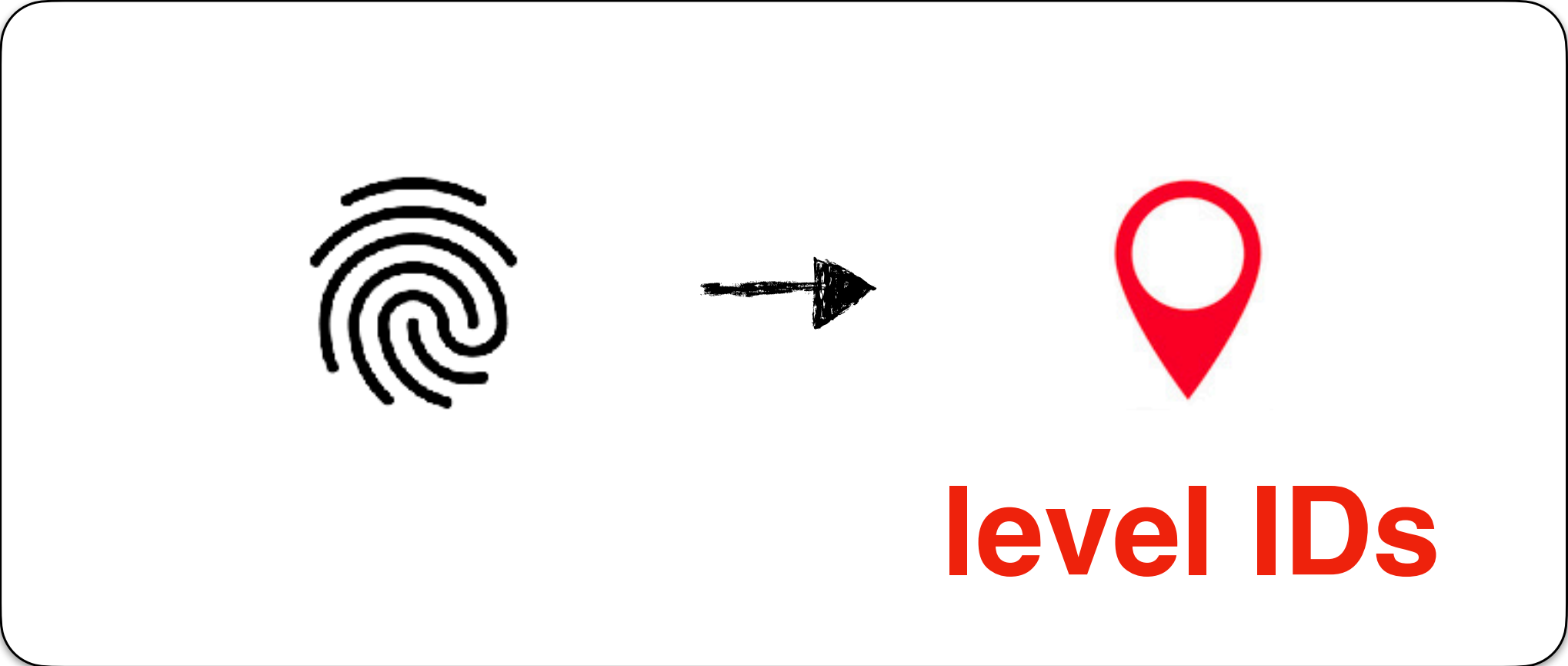






hash() = 





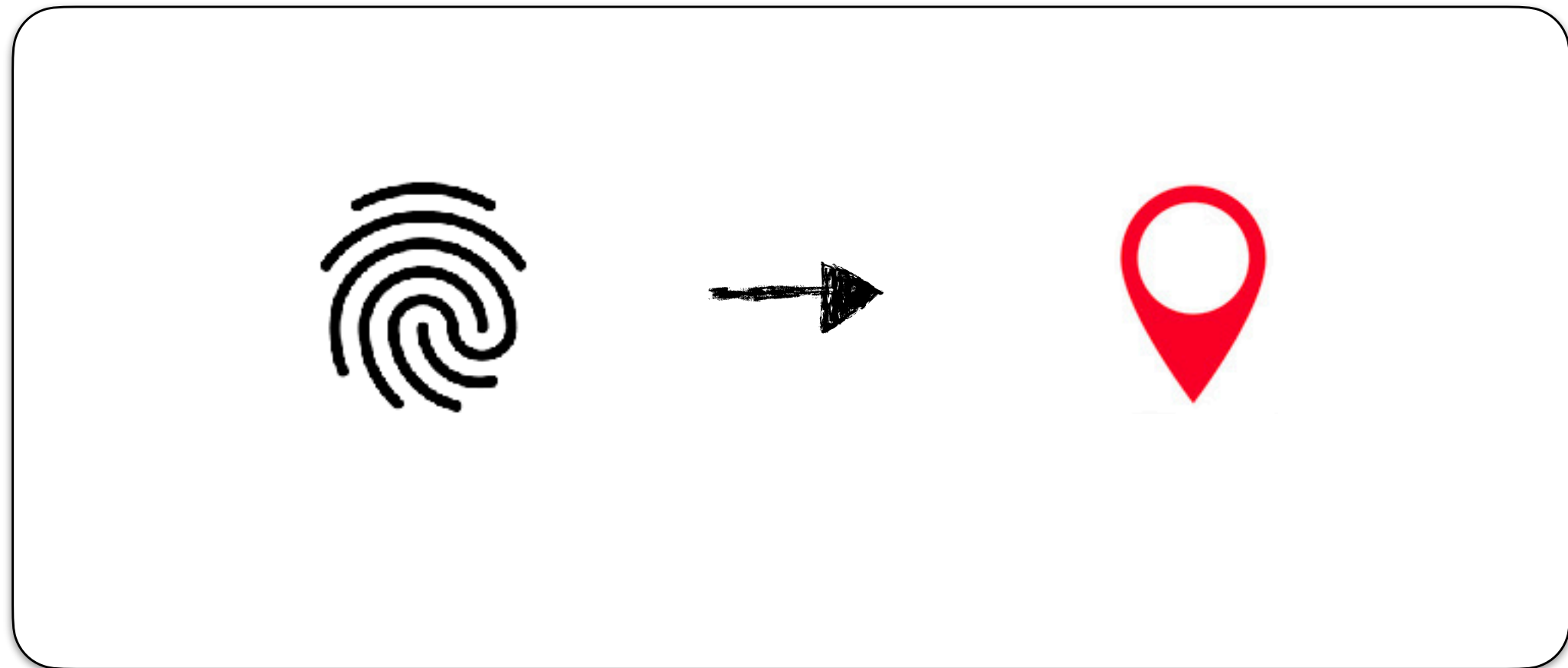
integer
encoding

01

10

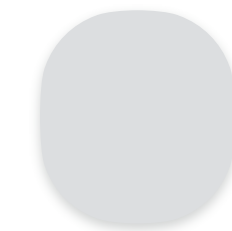
11





integer
encoding

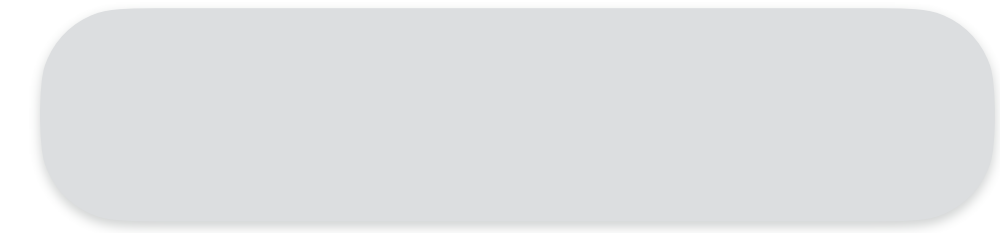
001



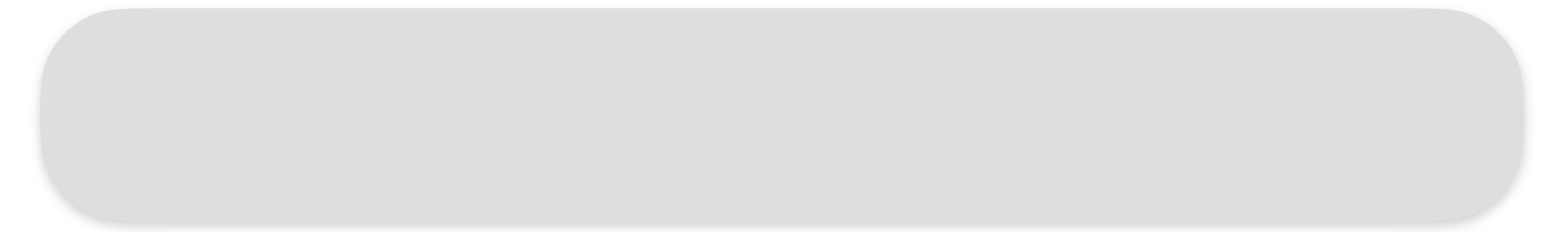
010



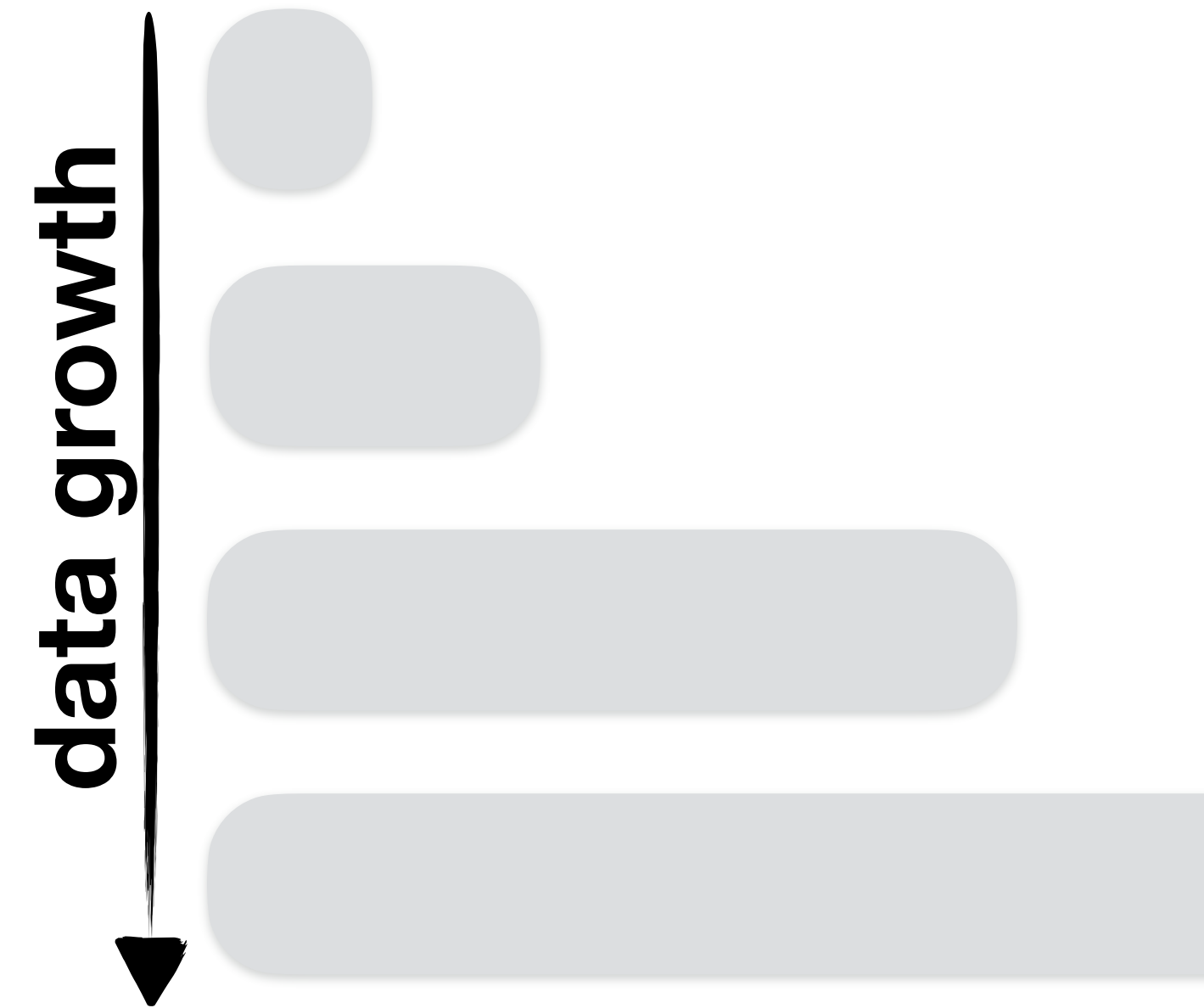
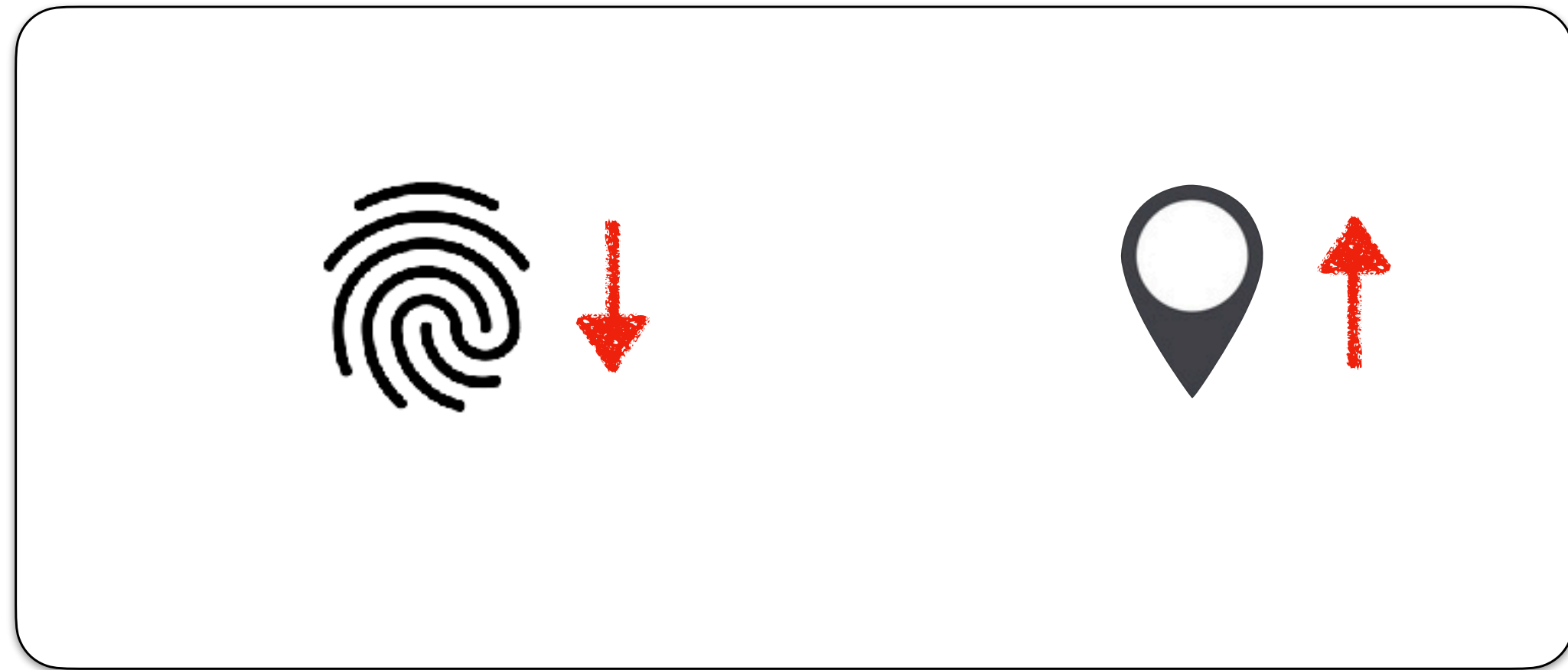
011




100



M bits / entry







1
2
2
3
3
3
3

1

2

3



most entries



smaller



001

01

1





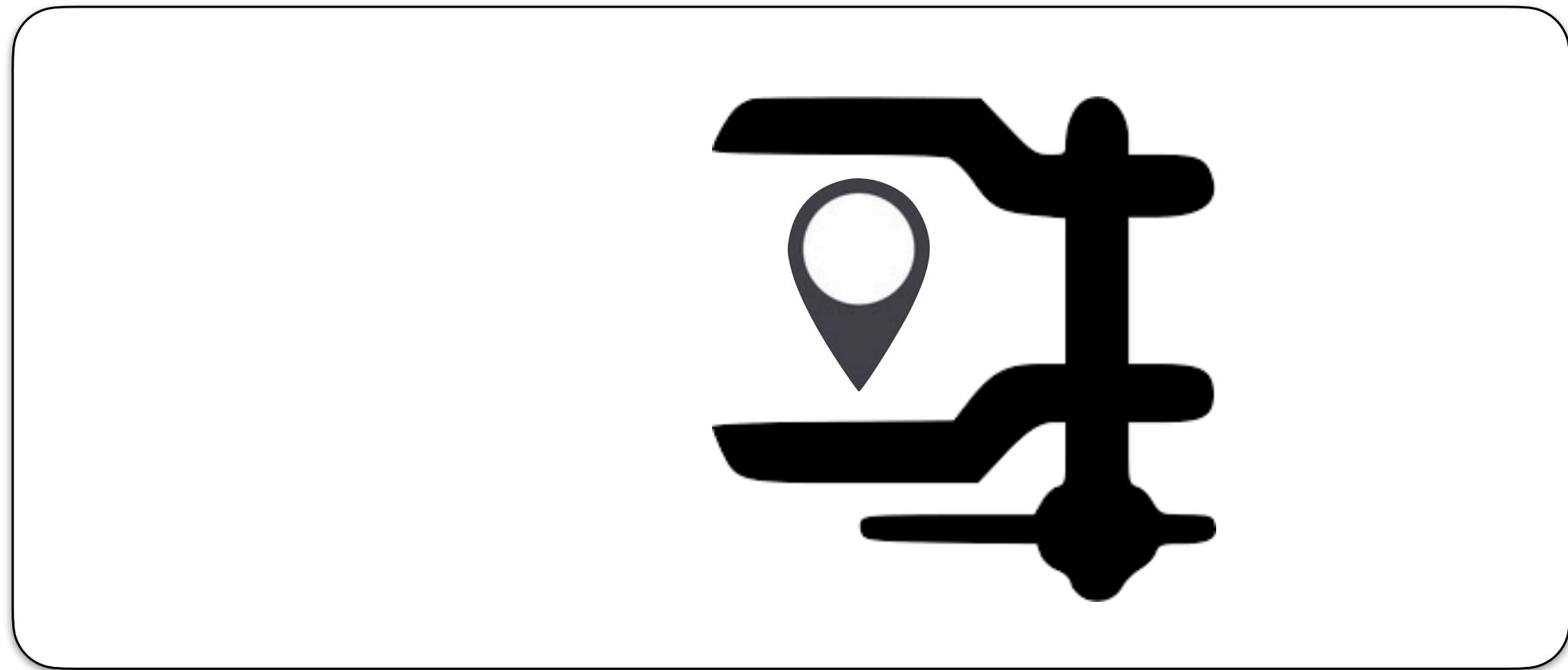
less common ↑

001

01

1





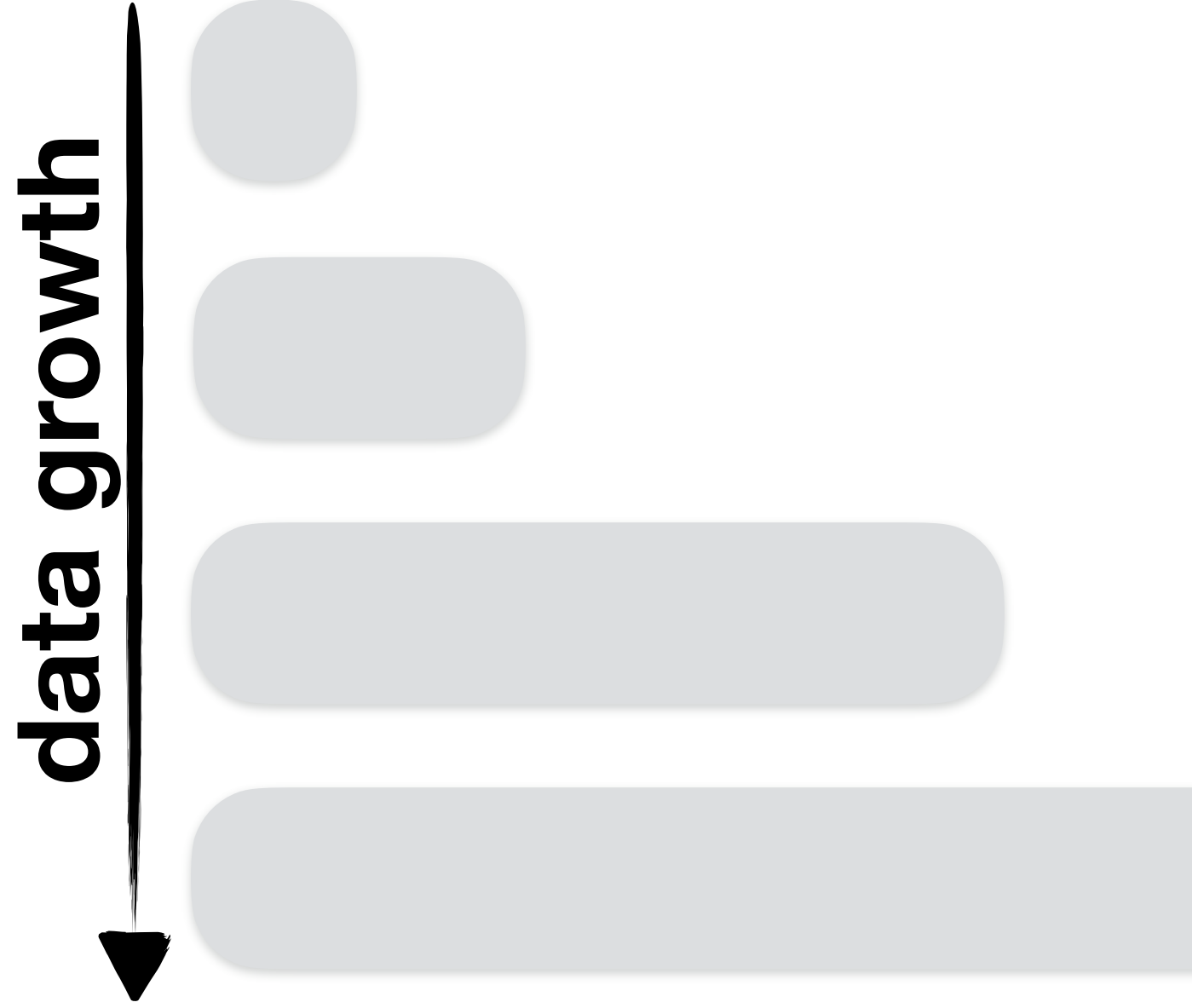
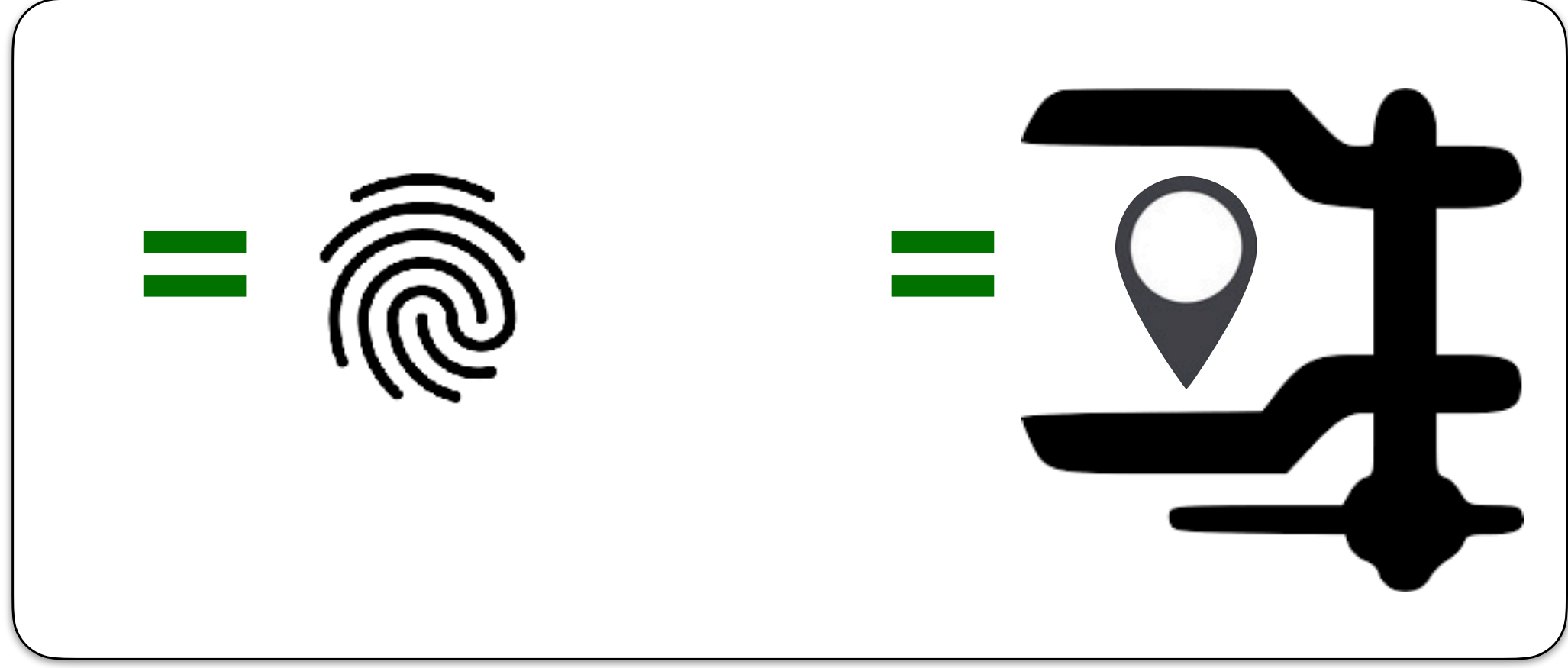
001

01

1



average code length < 2 bits





001

01

1

underflow



1

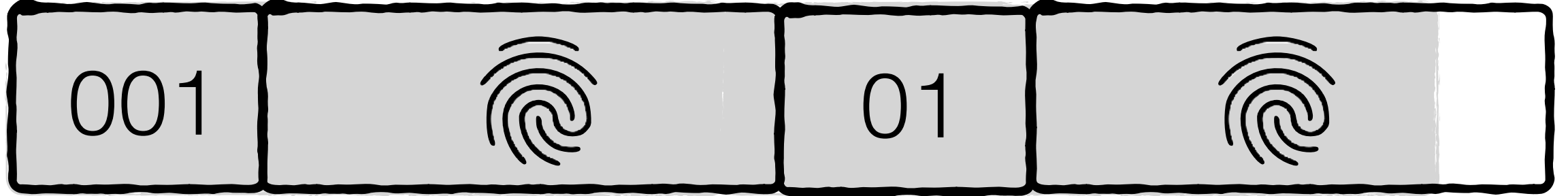
2

3

001

01

1



overflow

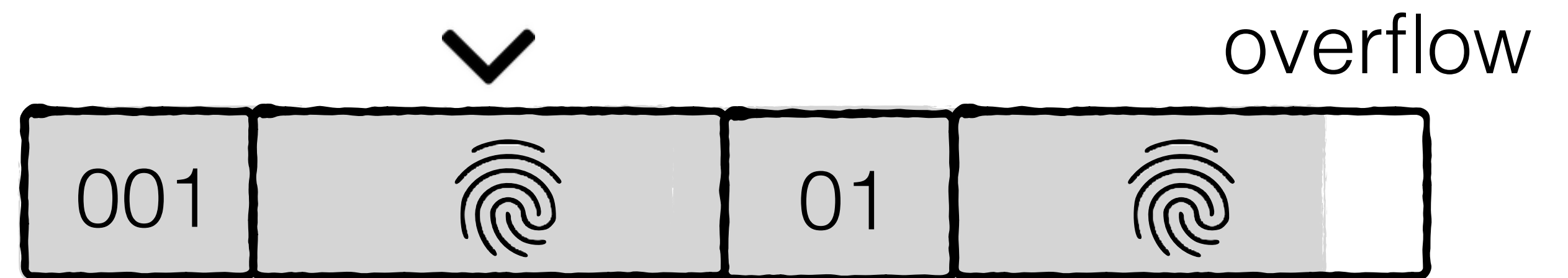
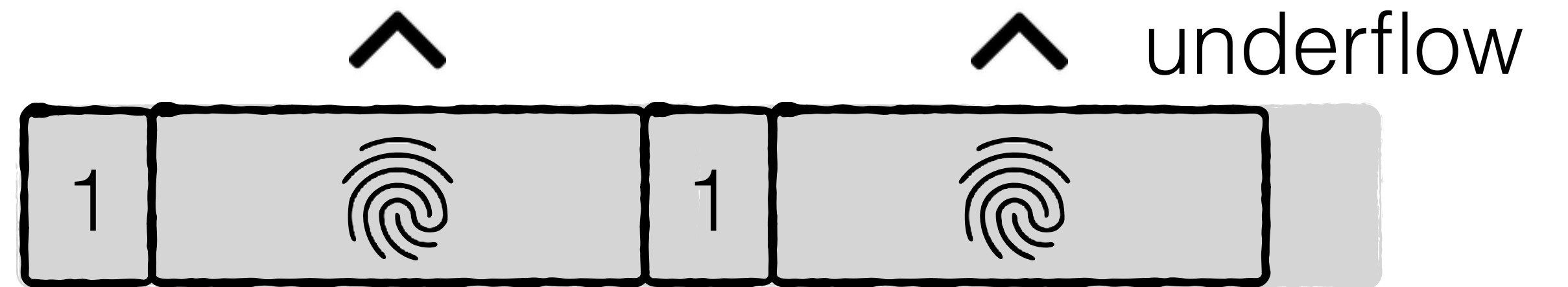
001



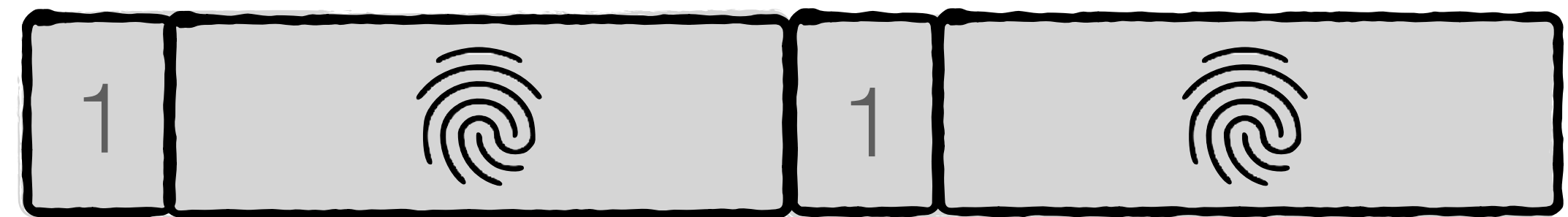
01



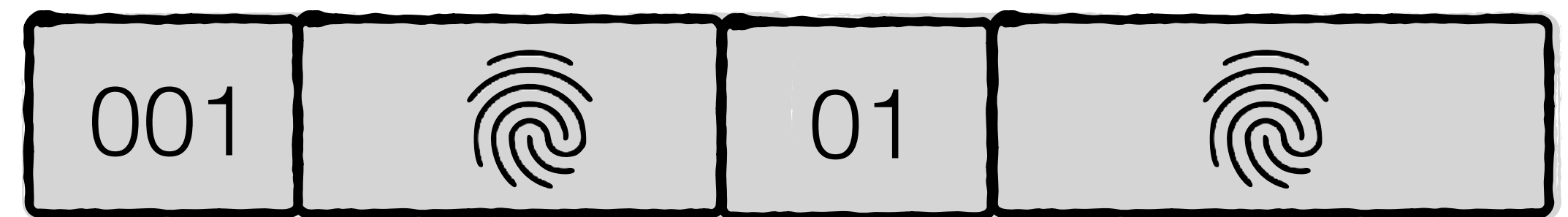
**larger fingerprints
for larger levels**



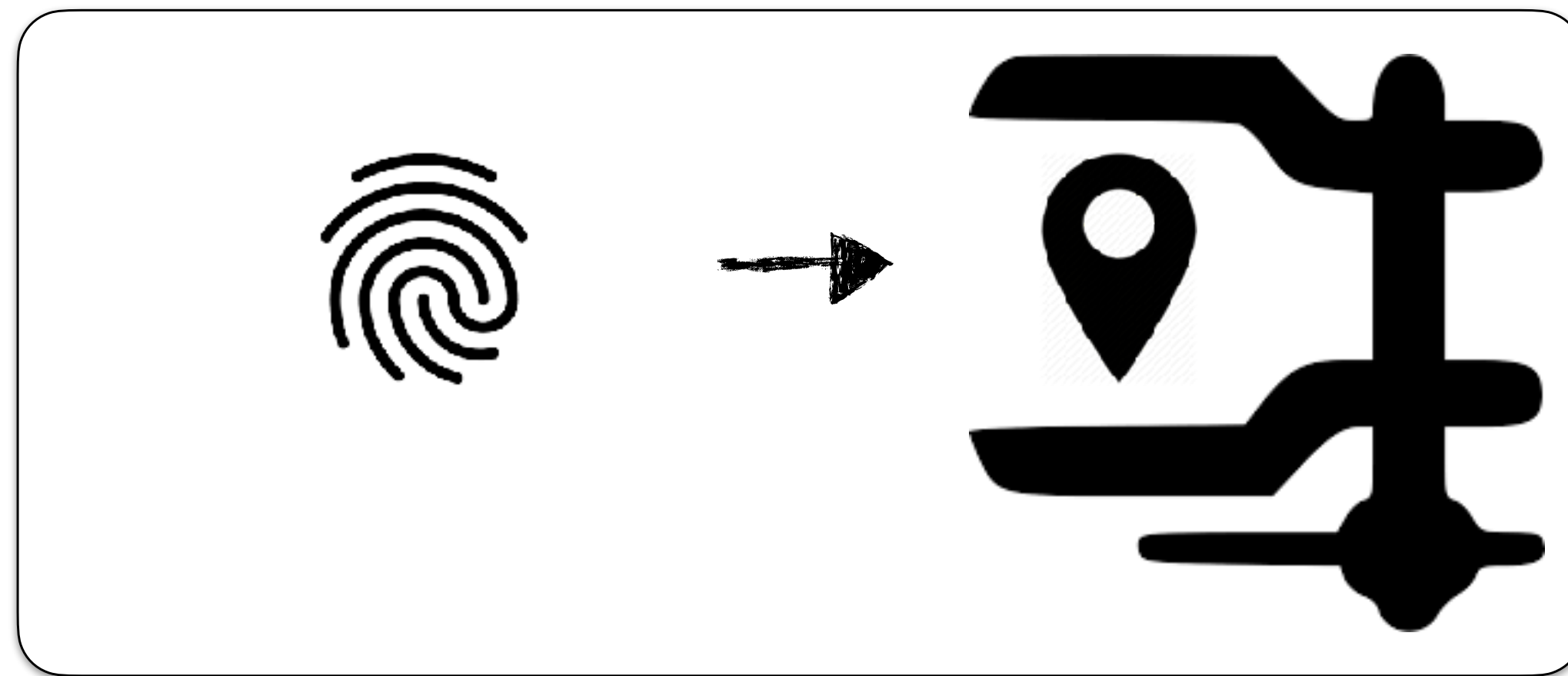
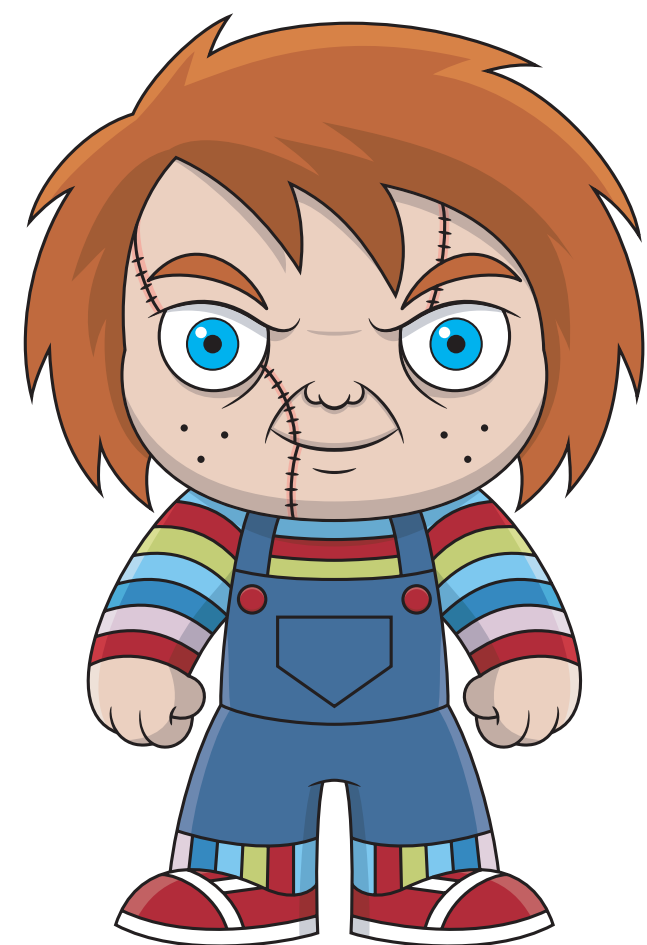
larger fingerprints
for larger levels



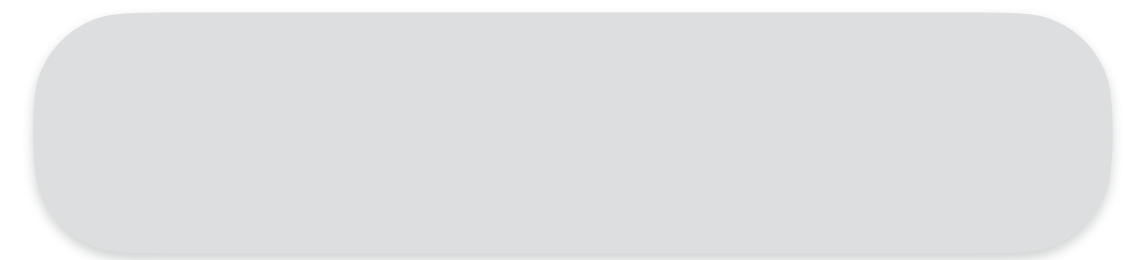
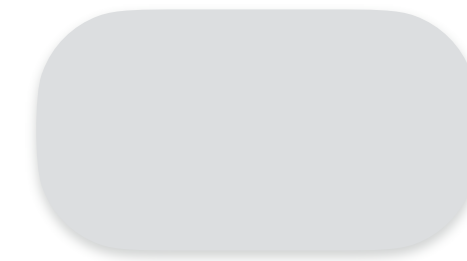
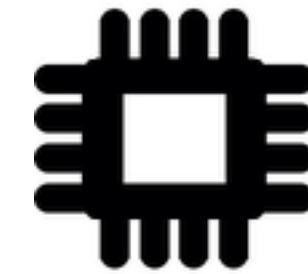
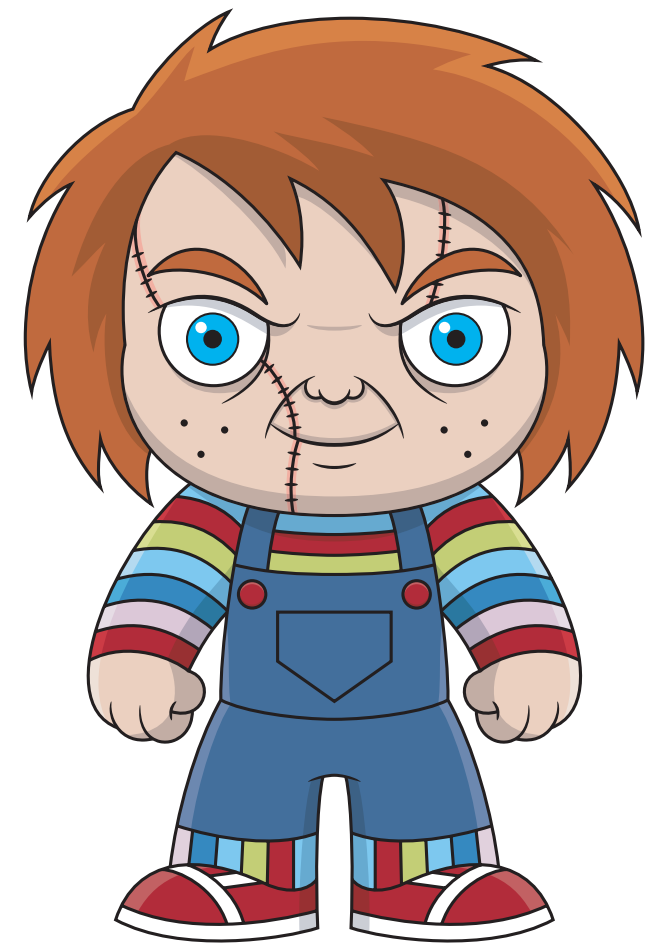
**more tricks in longer
presentation & paper**



Chucky

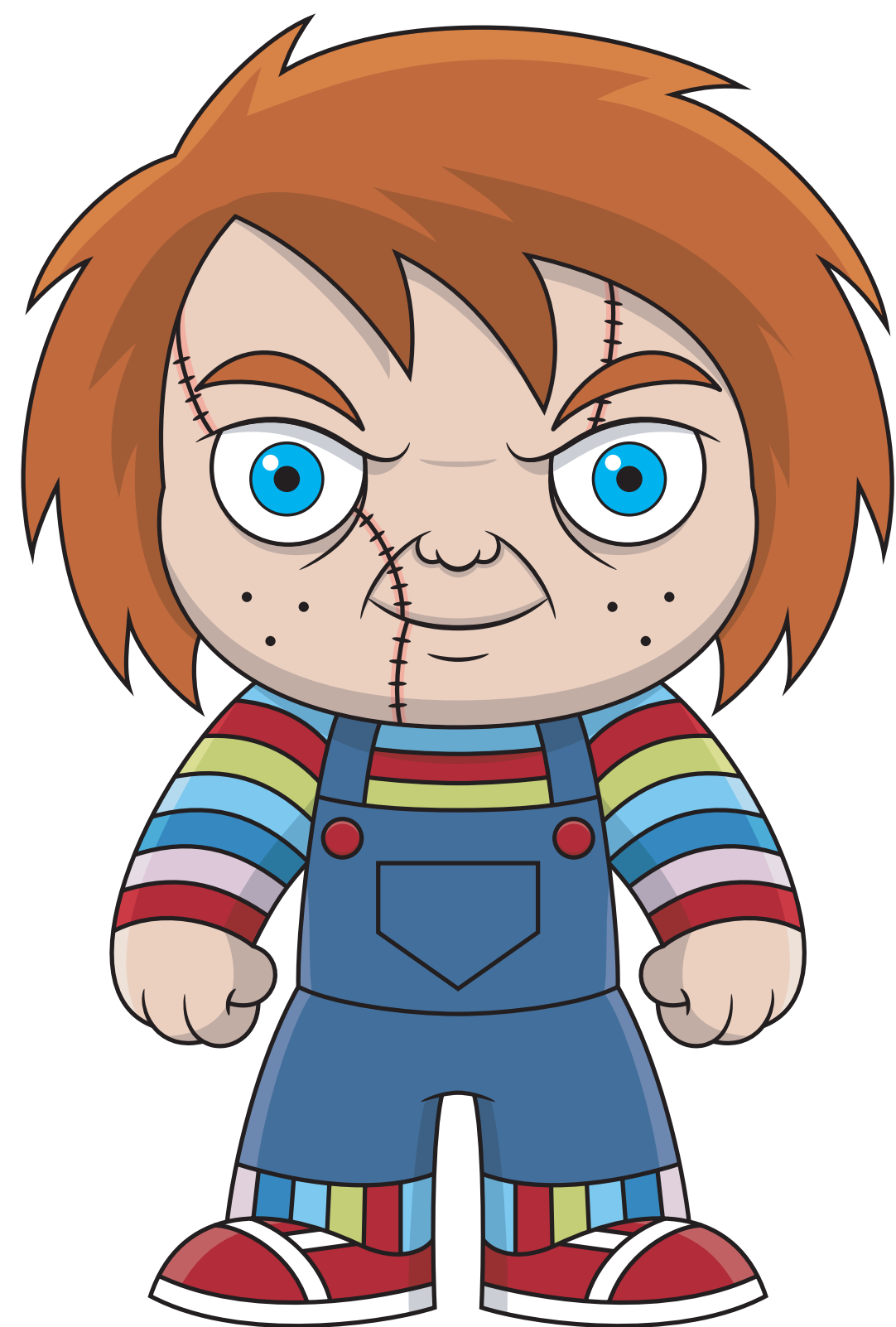


Chucky



=

=



Thank you!

