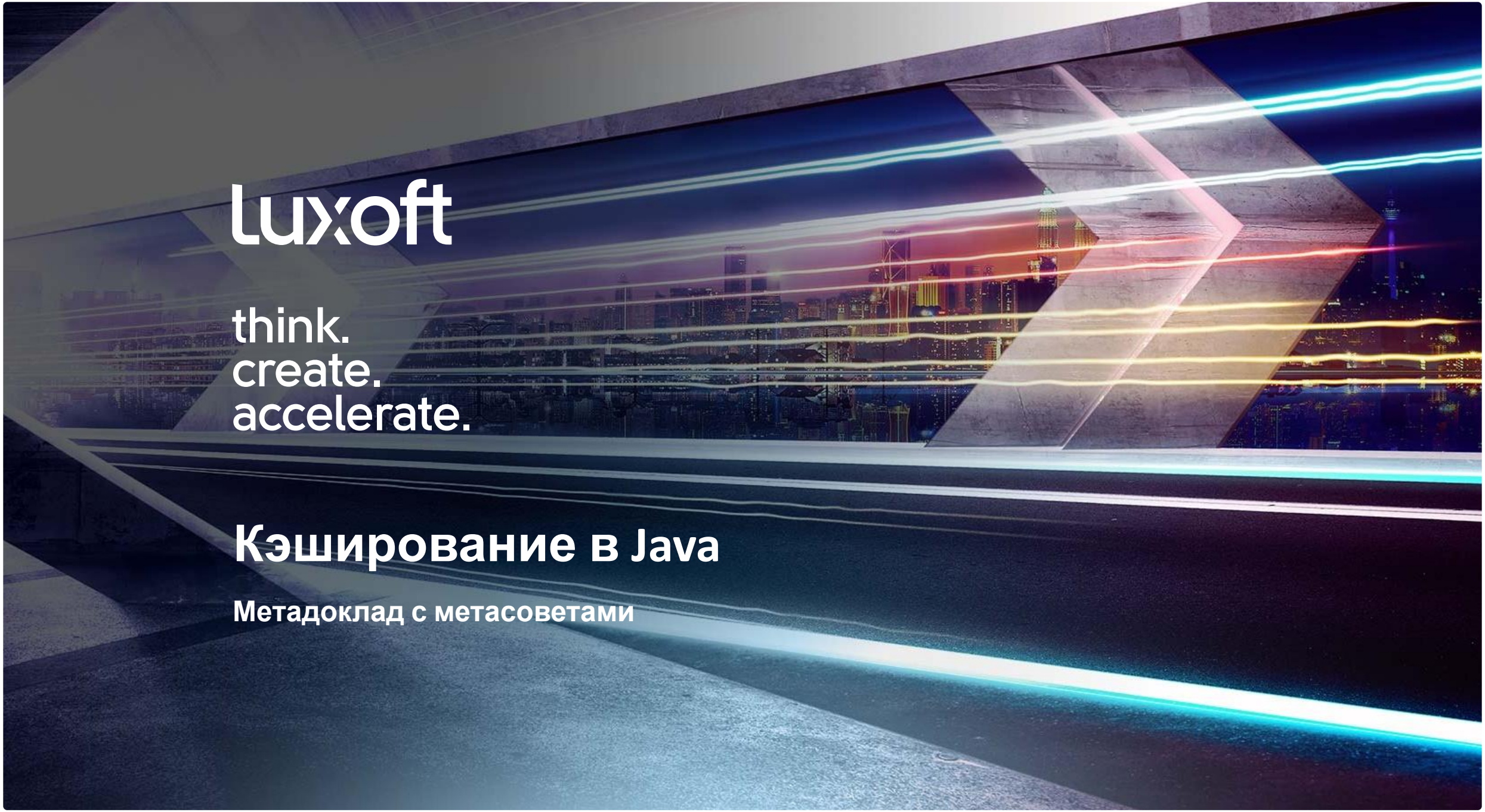
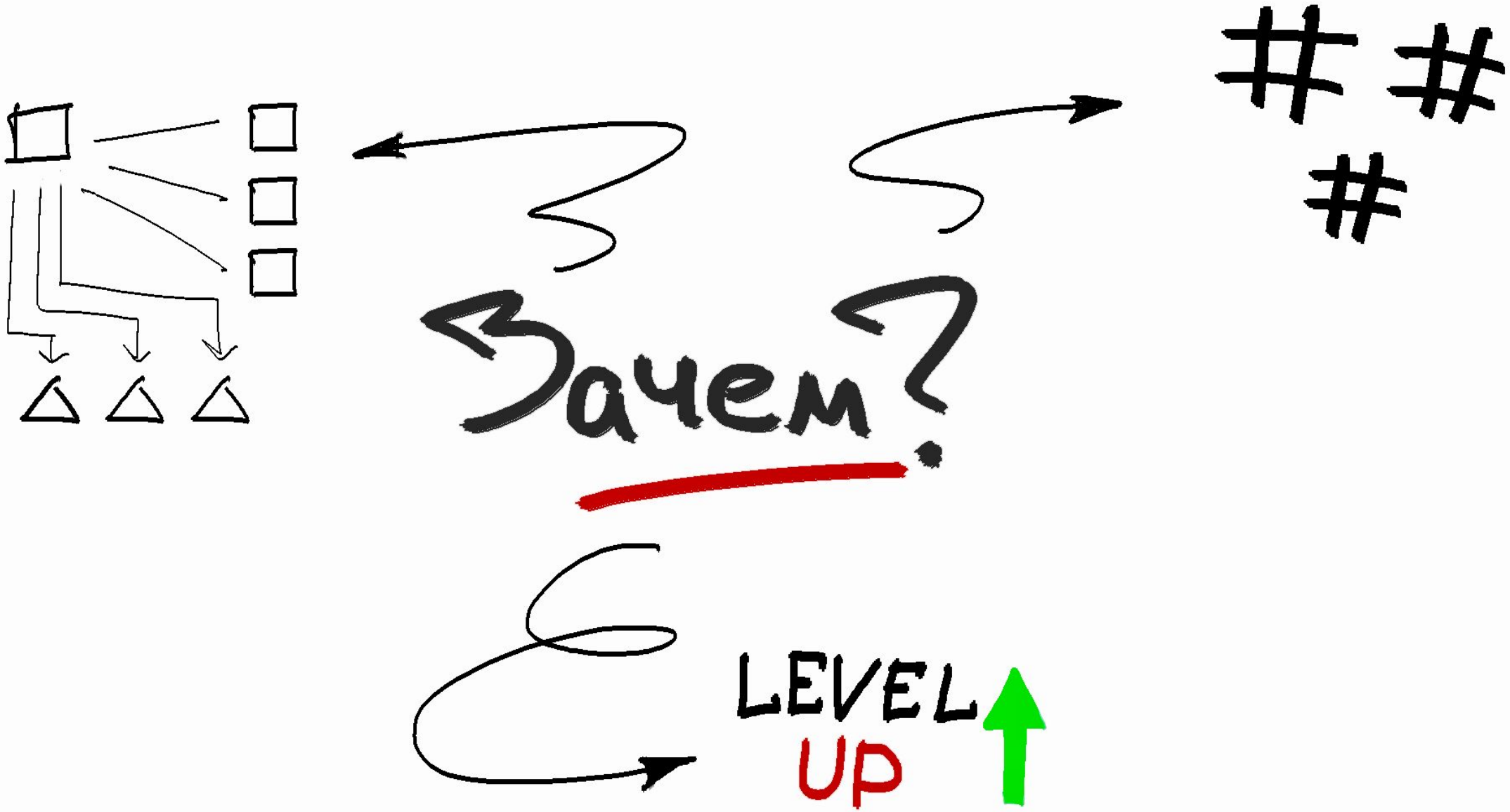


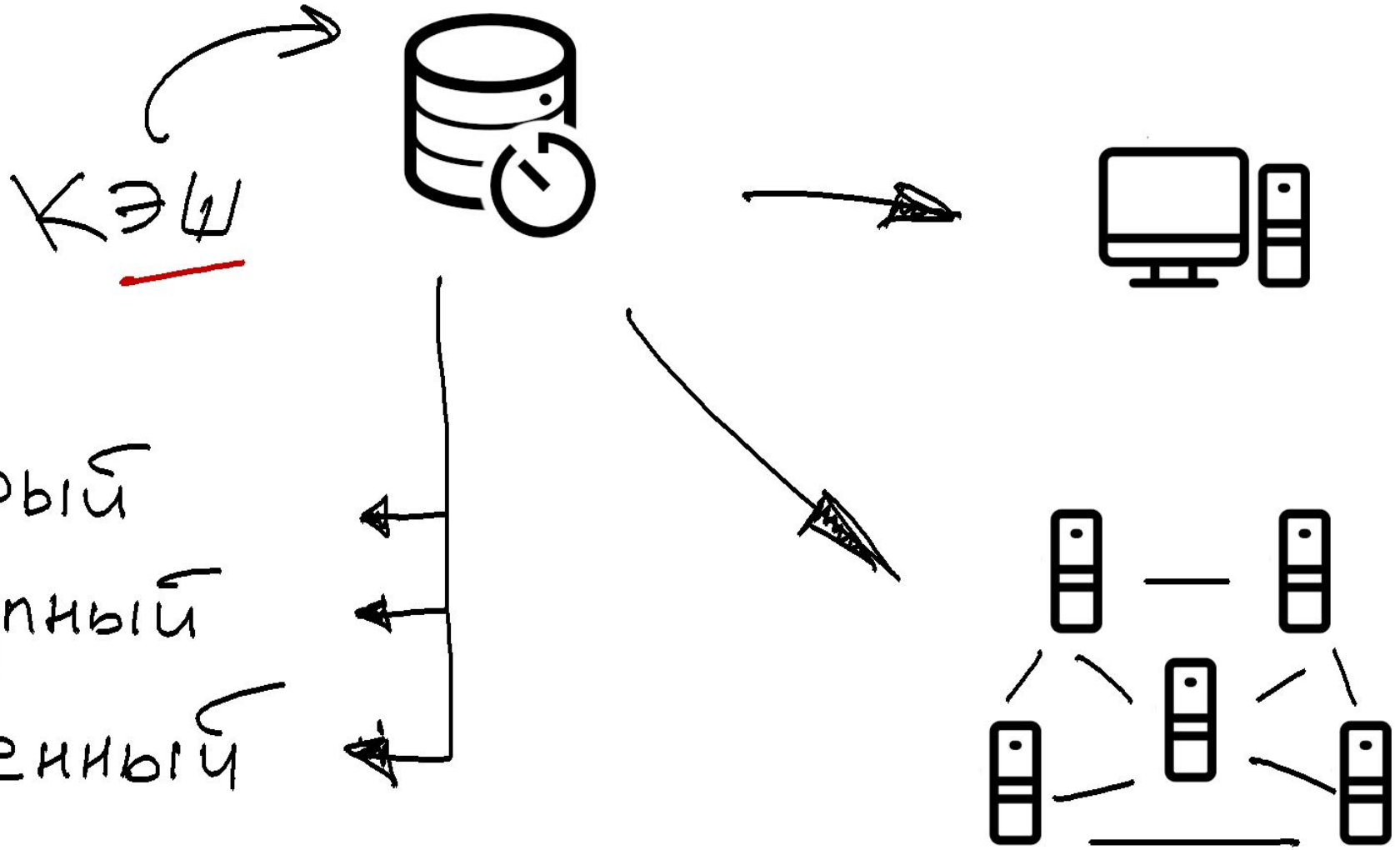
Кэширование в Java

Метадоклад с метасоветами

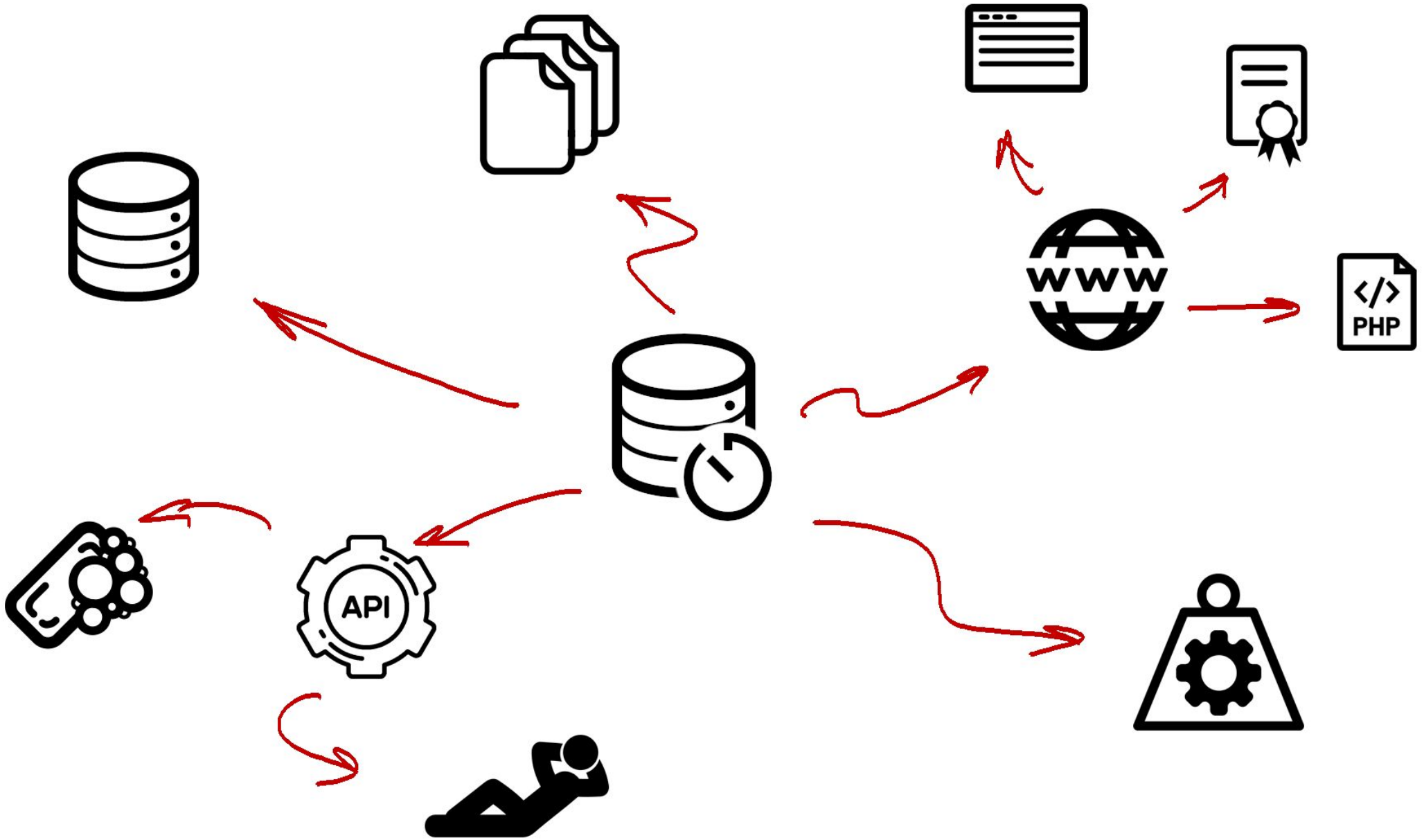

Почему?


Зачем?
LEVEL
UP


ЧТО ТАКОЕ
КЭШ?


КЭШ
• быстрый
• доступный
• временный


Что?

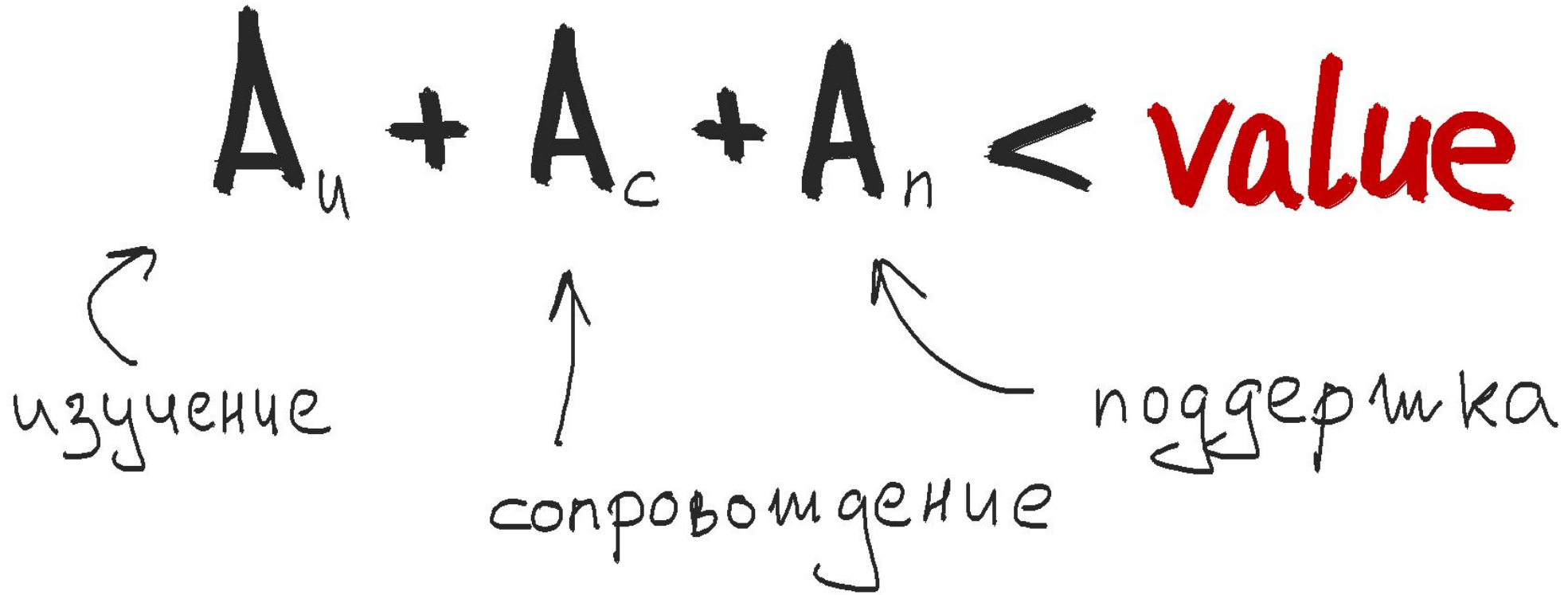

www
PHP
API

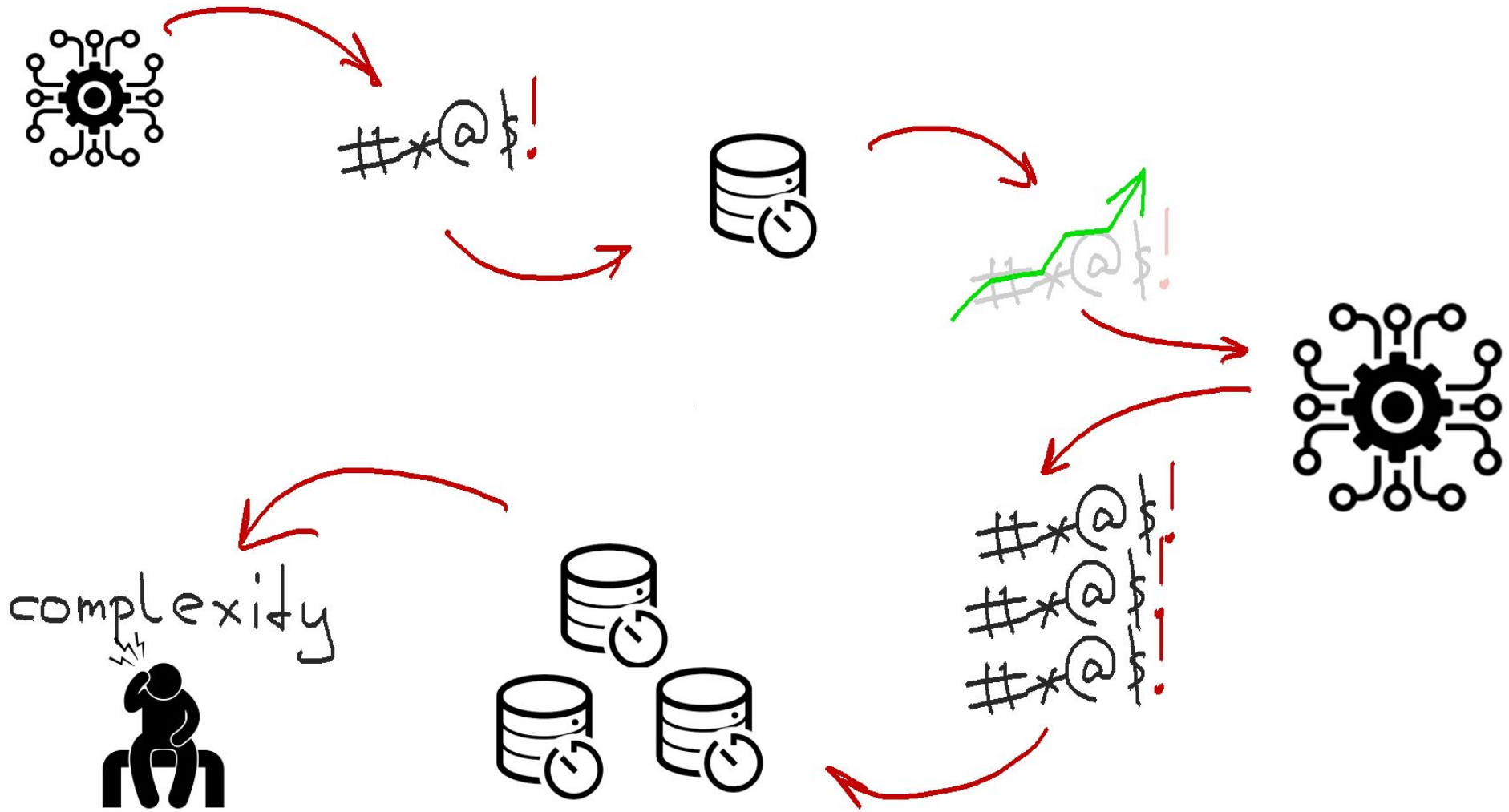
Luxoft

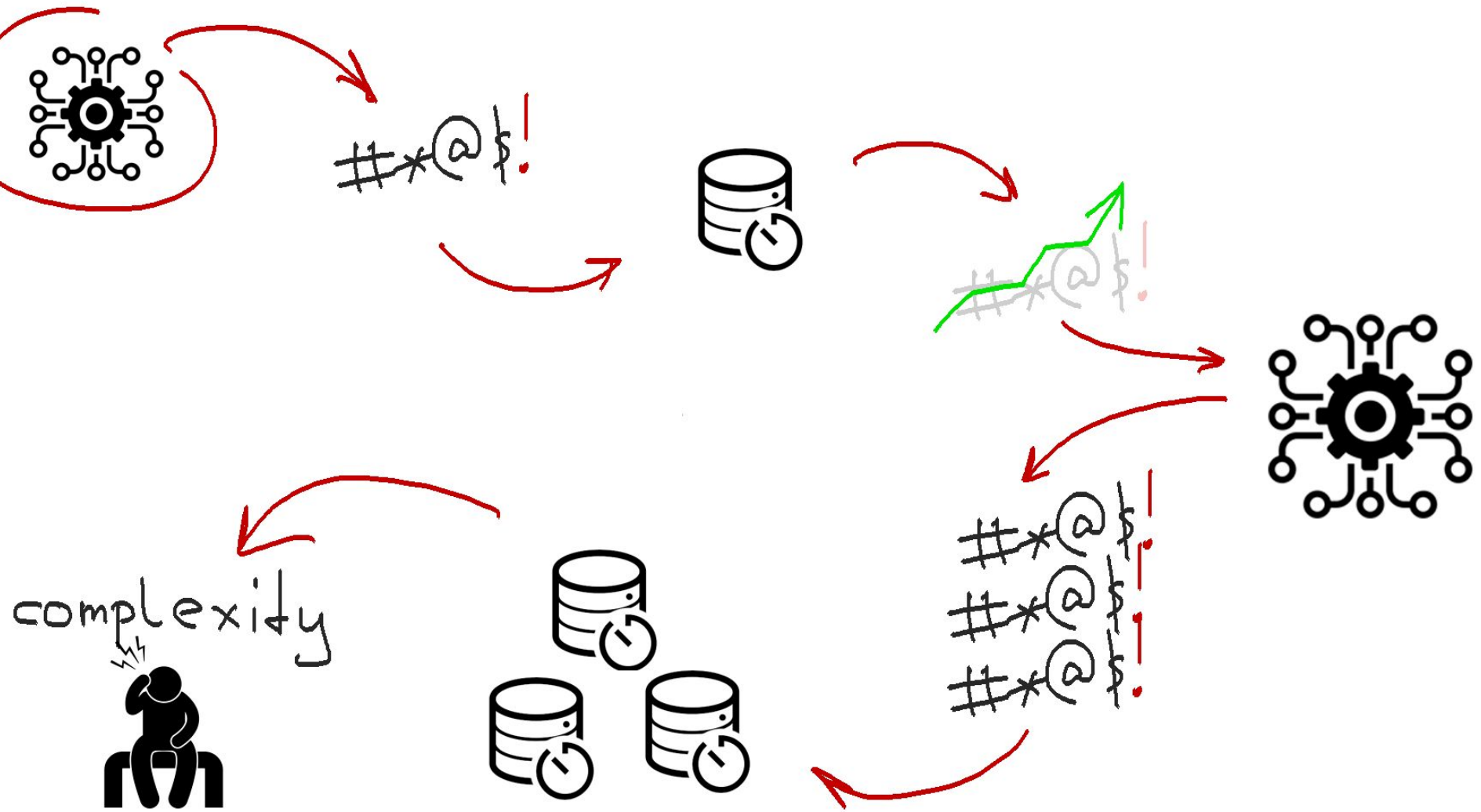
А надо ли?

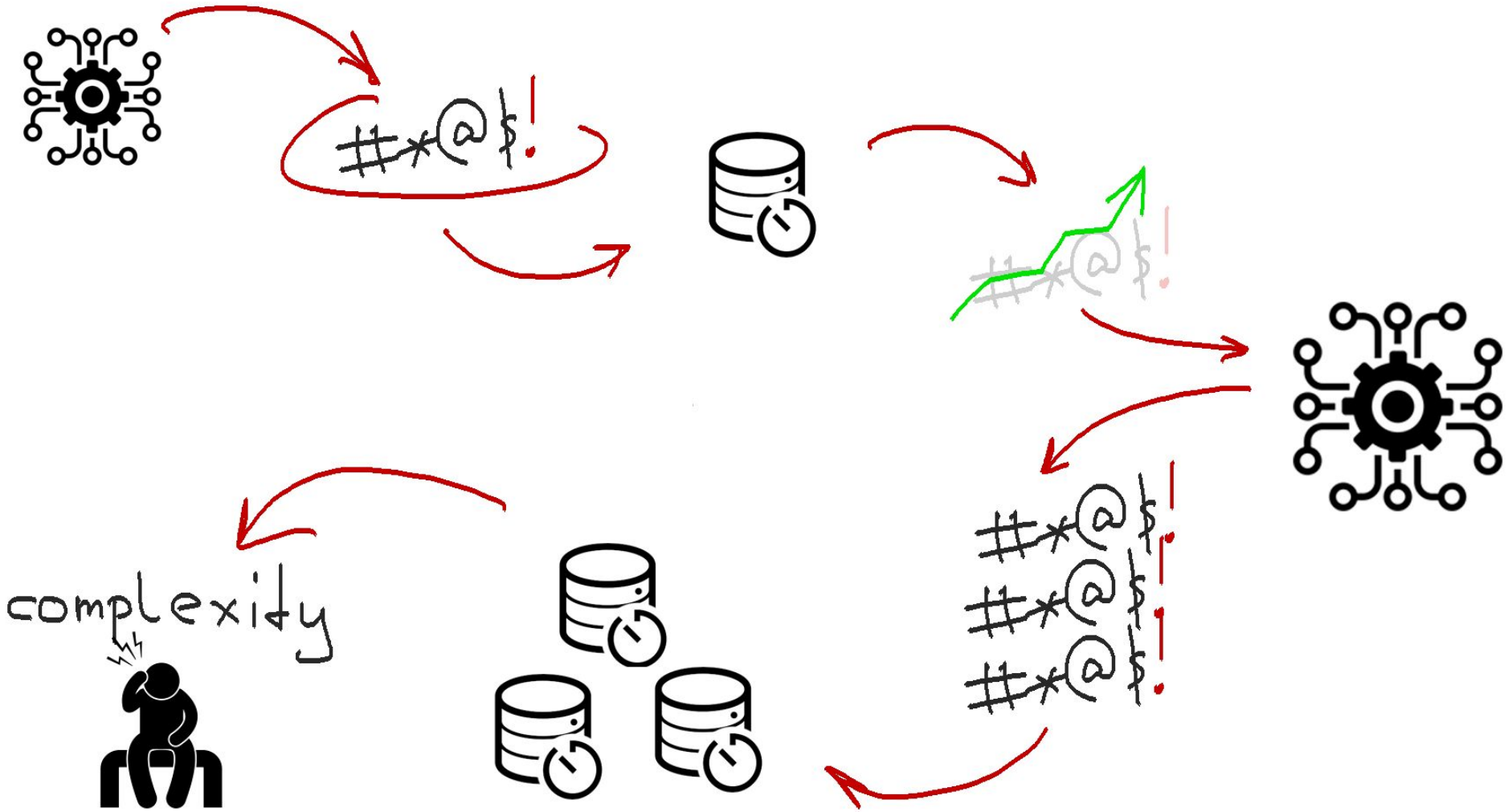
...premature optimization
is the root of all evil
(or at least most of it)
in programming.

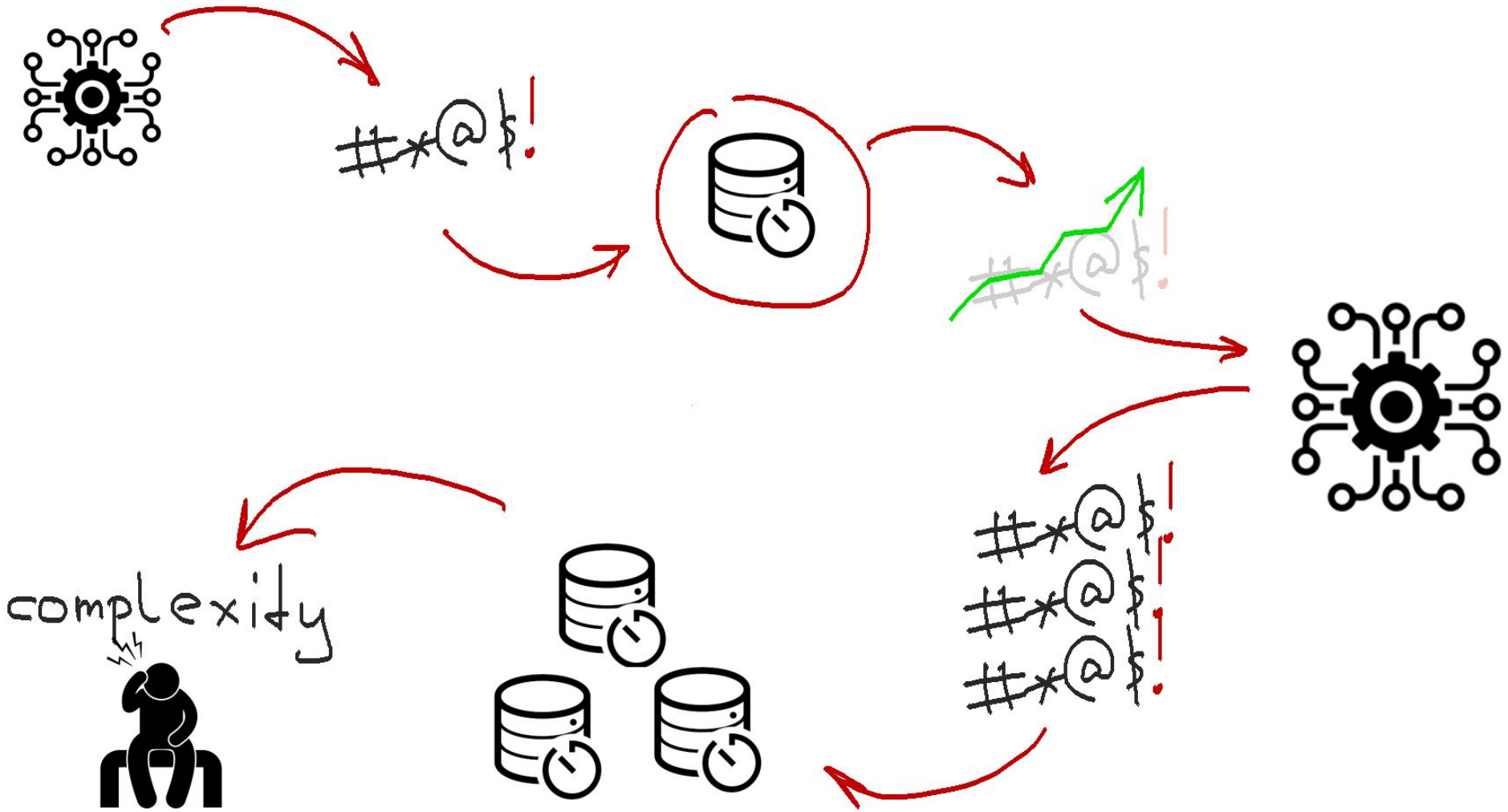
D. Knuth

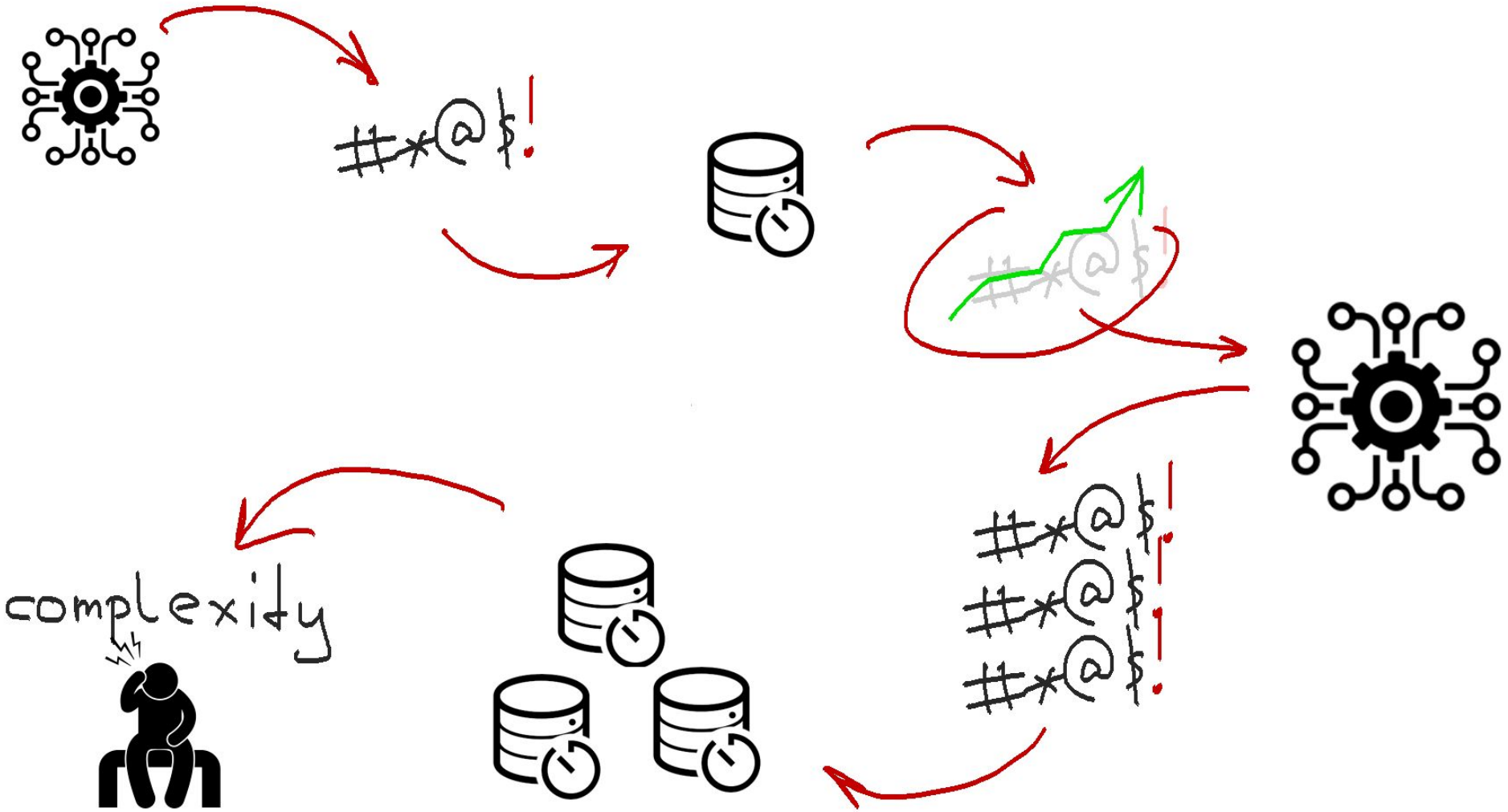

$A_u + A_c + A_n < value$
изучение
сопровождение
поддержка

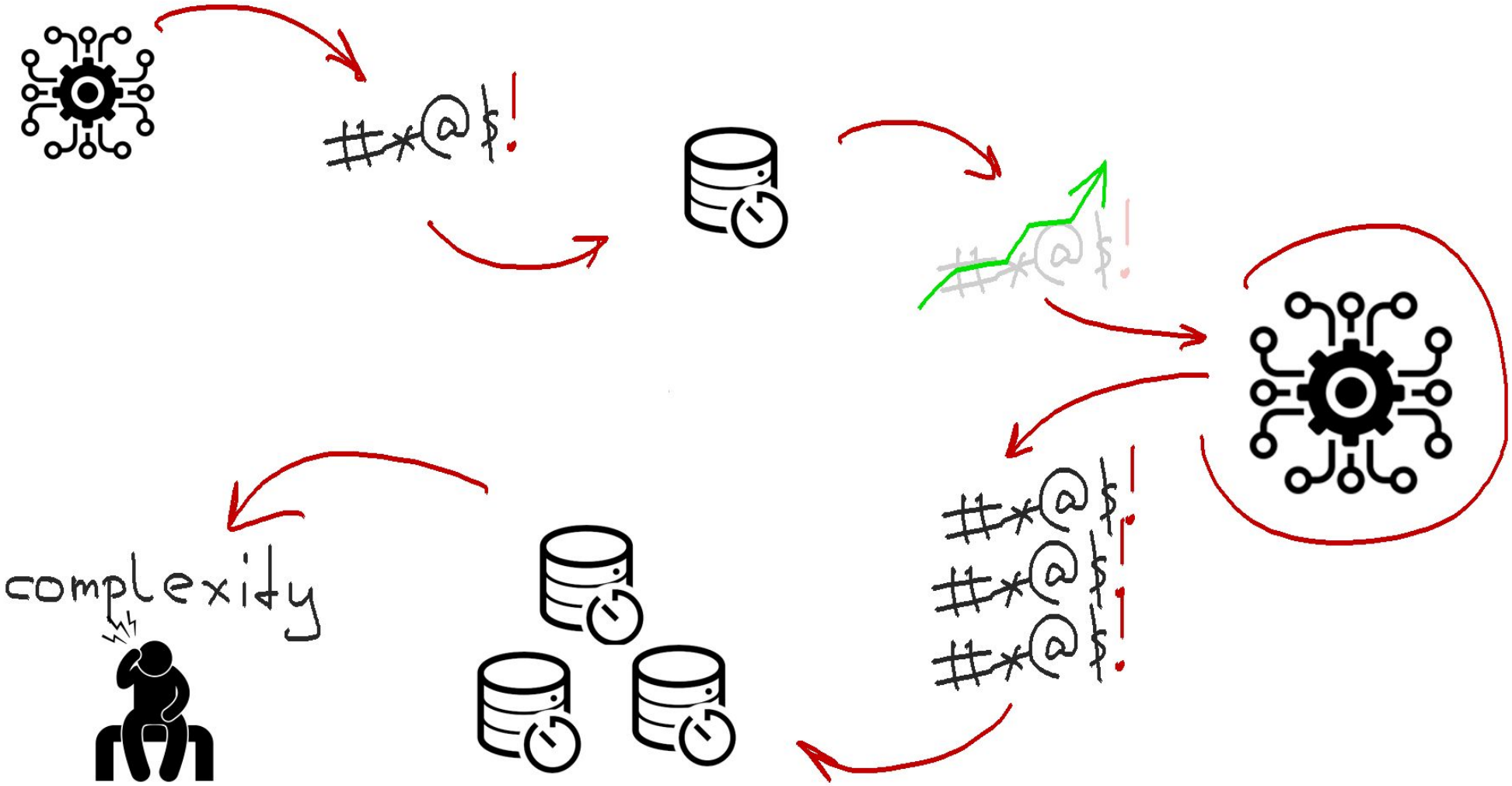

#*@$!
#*@$!
complexity
#*@$!
#*@$!
#*@$!

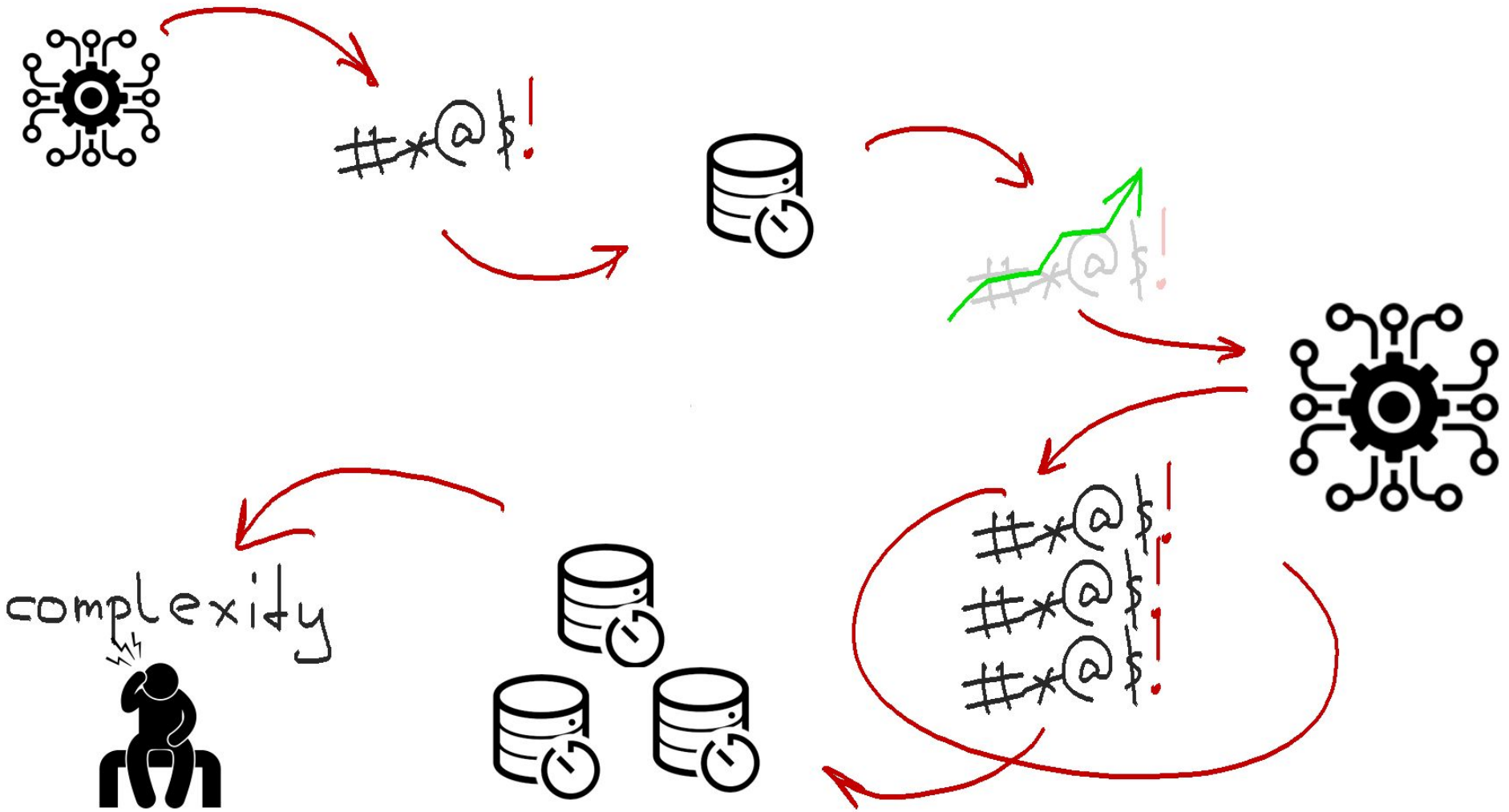

#*@$!
#*@$!
complexity

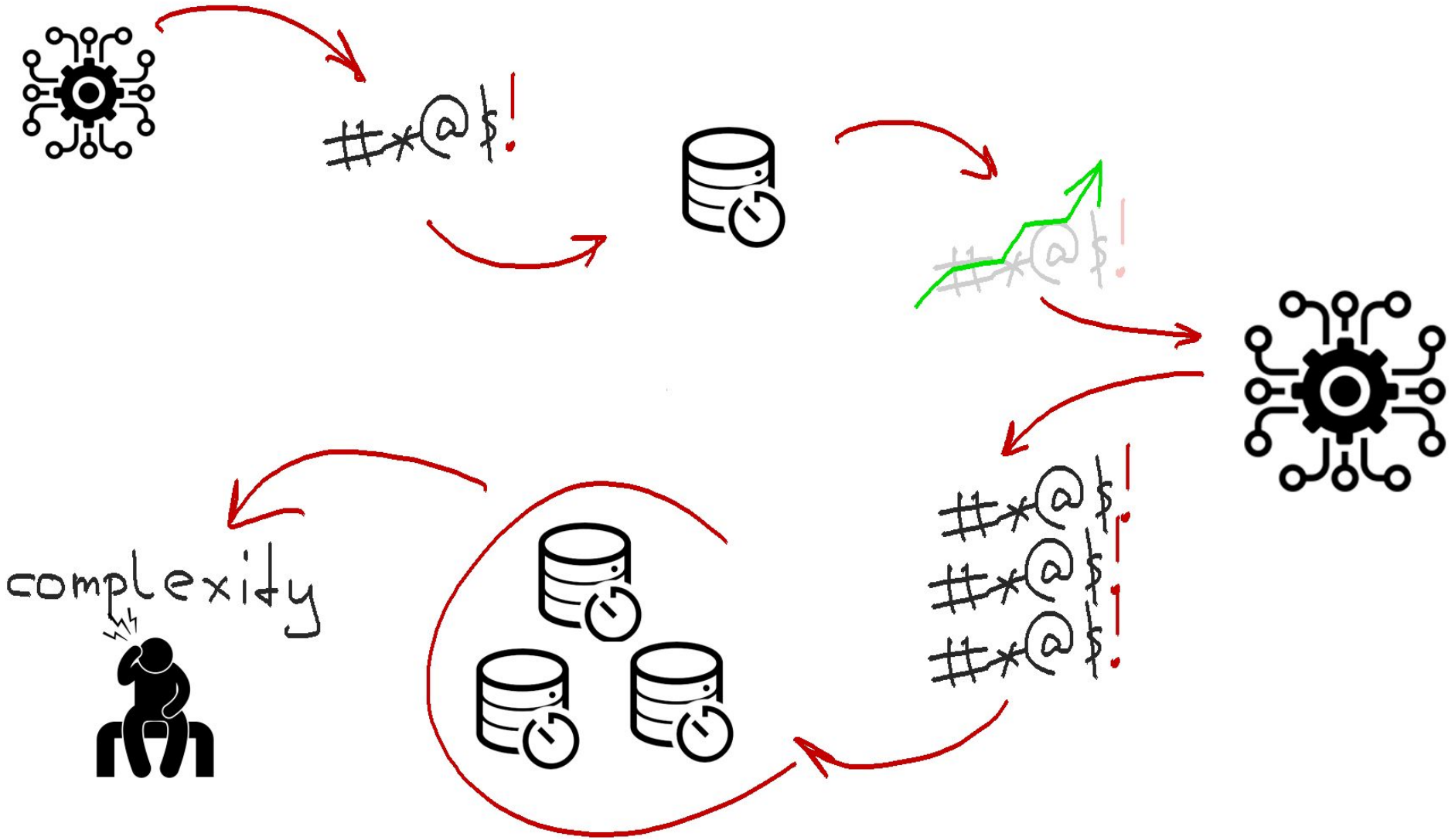

#*@$!
#*@$!
complexity
#*@$!
#*@$!
#*@$!

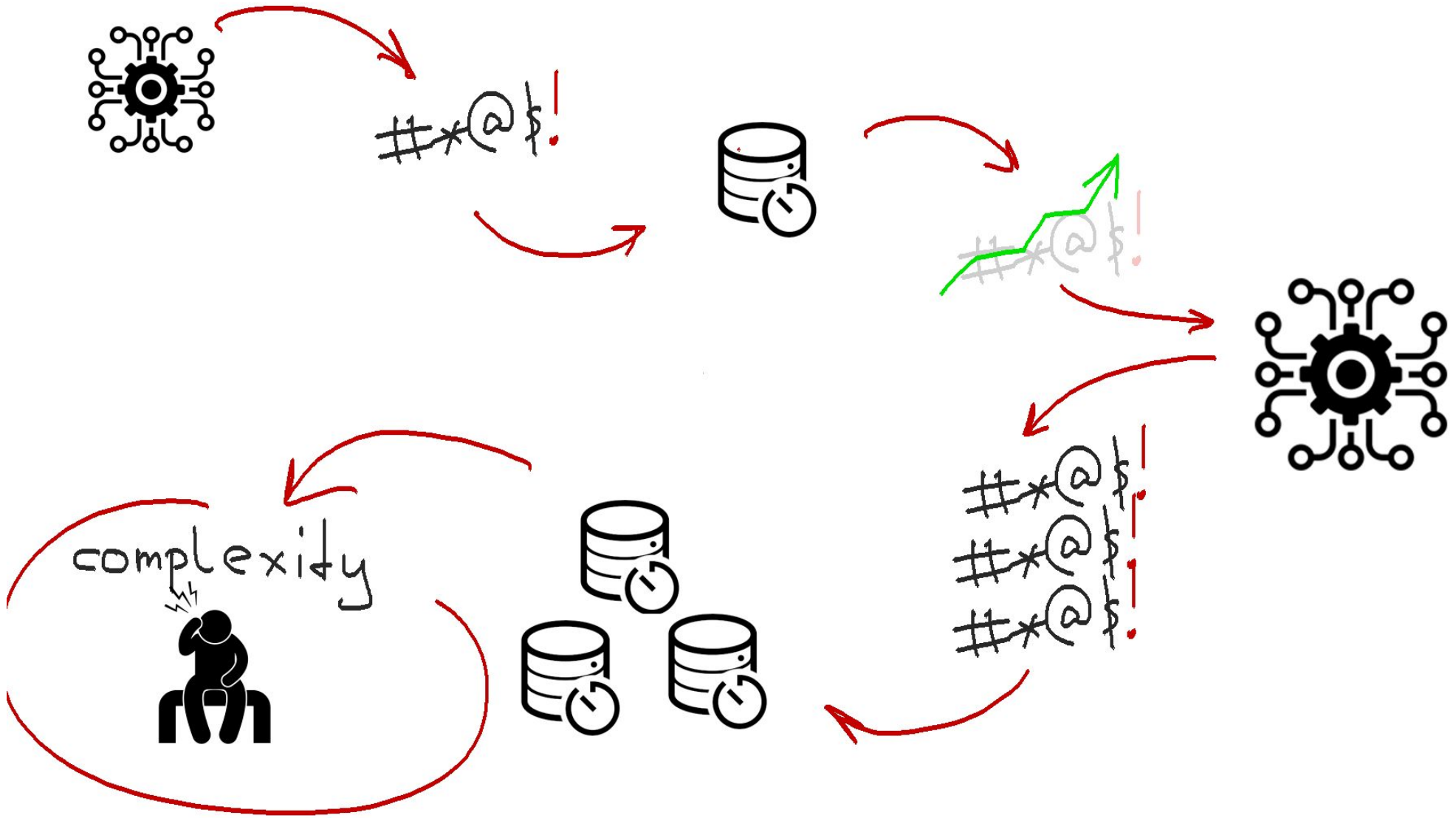

#*@$!
#*@$!
complexity
#*@$!
#*@$!
#*@$!

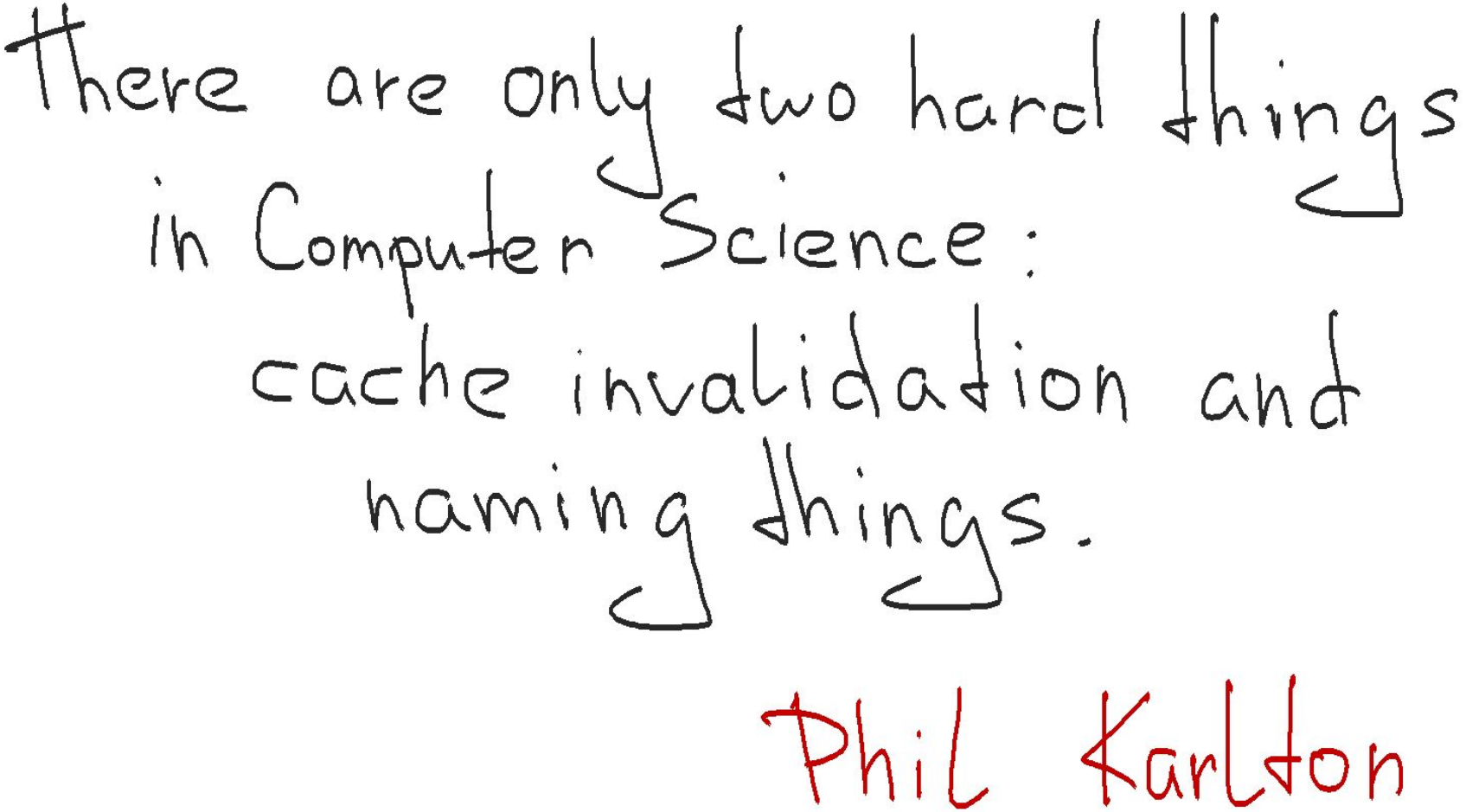

#*@$!
complexity


#*@$!
#*@$!
complexity


#*@$!
#*@$!
complexity


#*@$!
#*@$!
complexity
#*@$!
#*@$!
#*@$!


#*@$!
#*@$!
#*@$!
#*@$!
#*@$!
complexity


There are only two hard things
in Computer Science:
cache invalidation and
naming things.
Phil Karlton

Сложности:

* инвалидация
* дебаг
* производительность
* выход из строя


С чего
начать?

1-й этап

1. Поможет ли кэширование снизить затраты на улучшение
   - производительности
   - масштабируемости

2 Как неактуальные данные повлияют на работу системы/пользователя?

3 Инвалидация
- → общая?
- → частные случаи?
- → как?

4 Не изменятся ли потребности в обозримом будущем?

5 Где должно происходить кэширование?

6 Кто должен за него отвечать?

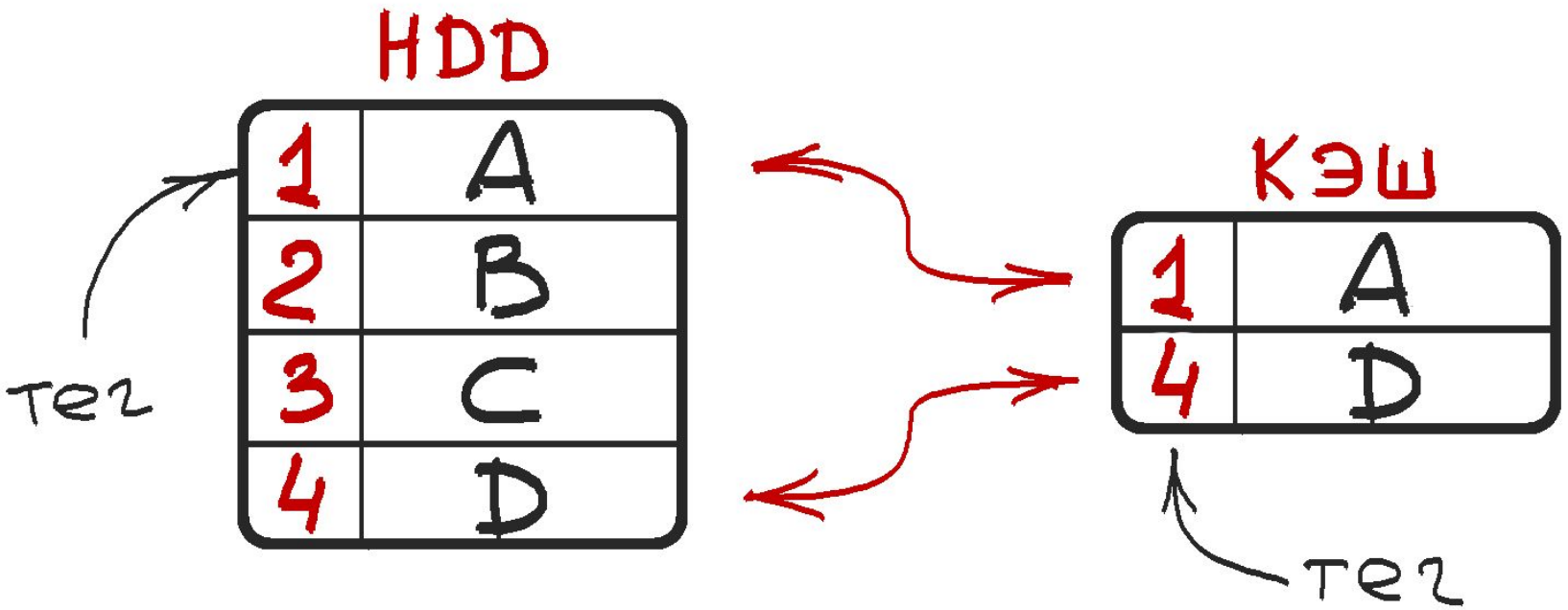
2-й этап

1. Доступный функционал
   - full-fledged
   - light weight
2. Open source или commercial?
3. Тип кэша? In-process, distributed?
4. Производительность? Надежность?

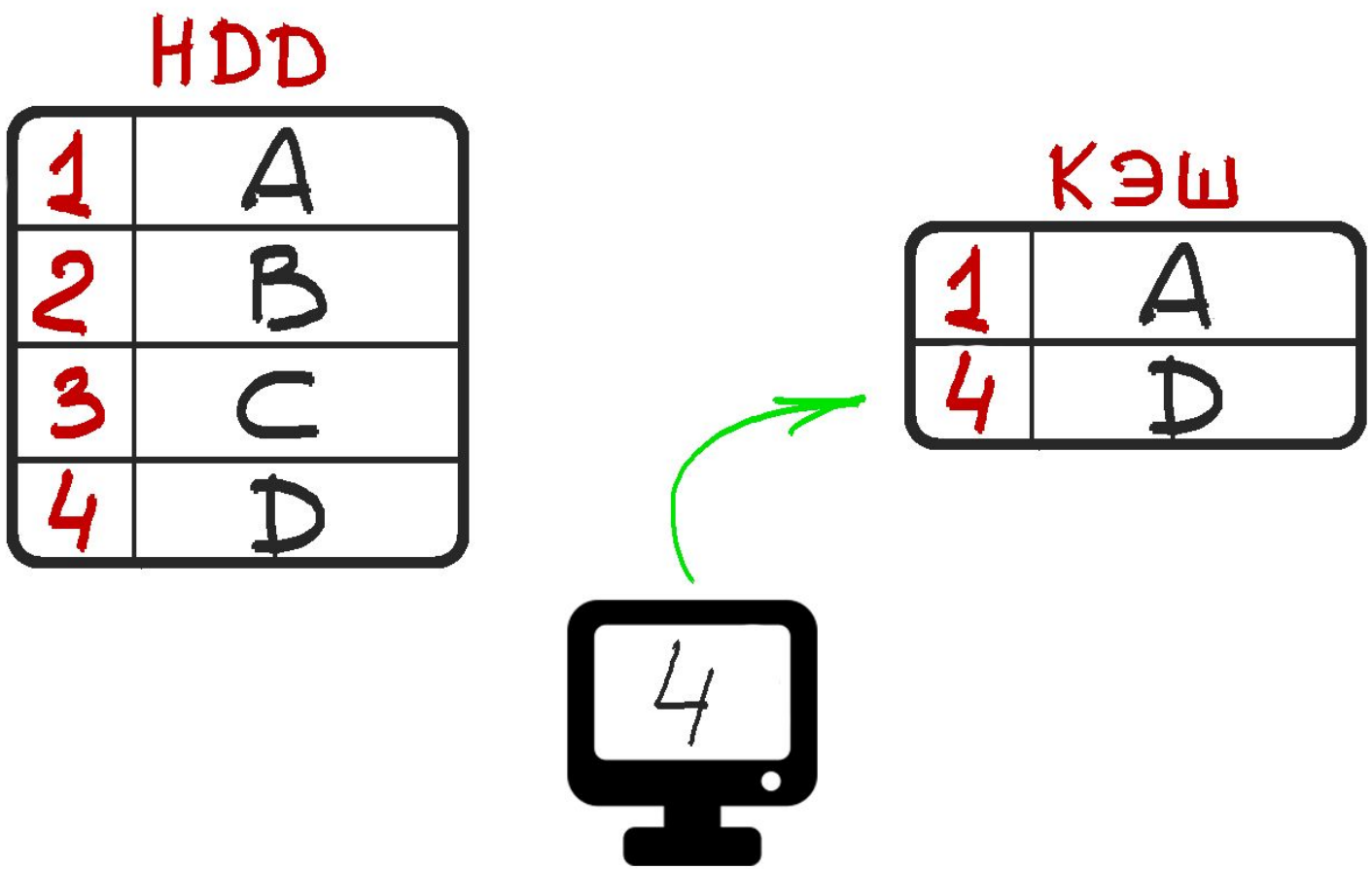
и т.д.

Терминология

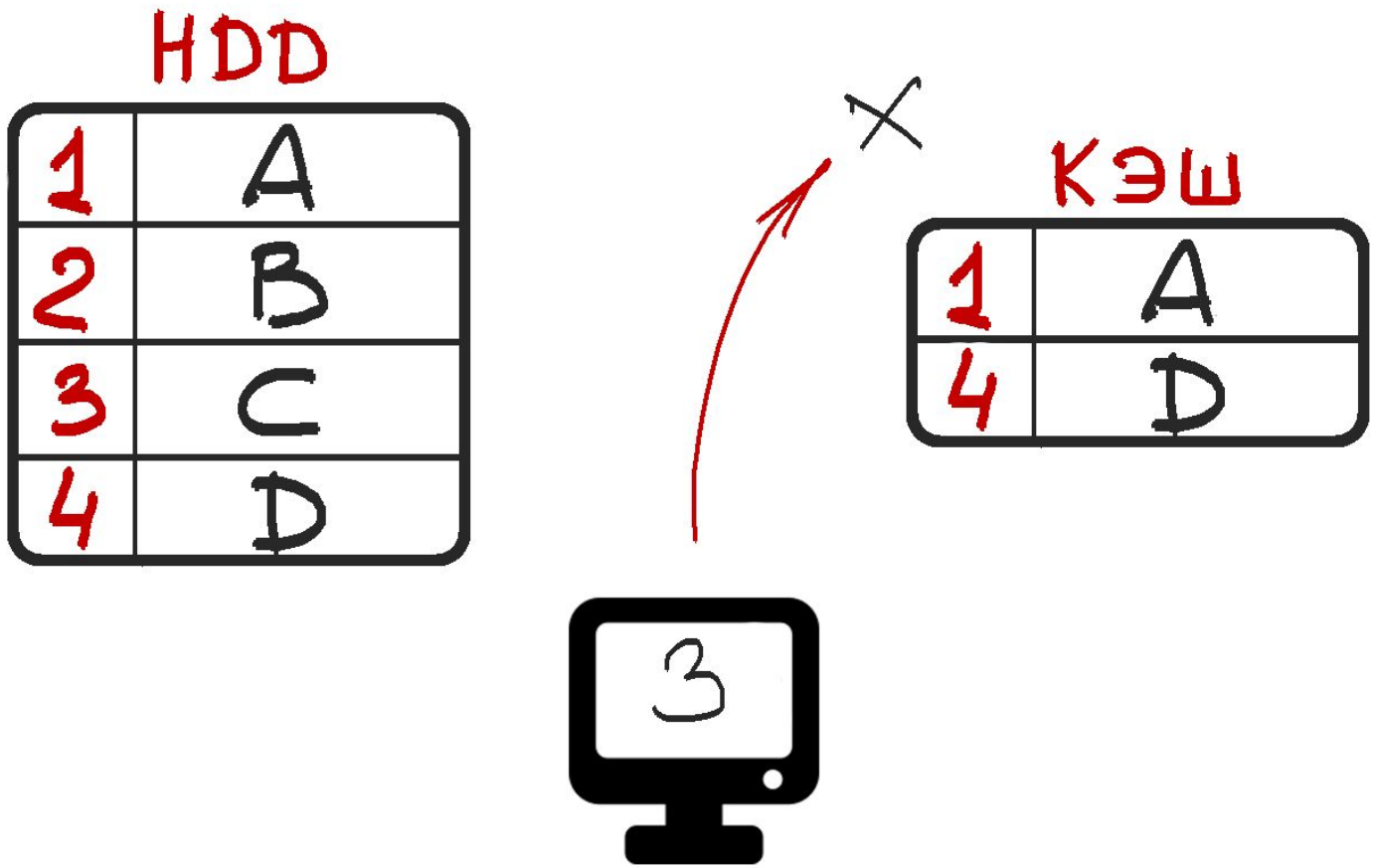
Tez



Попадание (hit)



Промах (miss)



Метрики

* cache hit percentage
* eviction count
* total load count
* average get time
* memory usage

и т.д.

Eviction policies

- → Least Recently Used (LRU)
- → Most Recently Used (MRU)
- → Least-Frequently Used (LFU)
- → Adaptive Replacement Cache

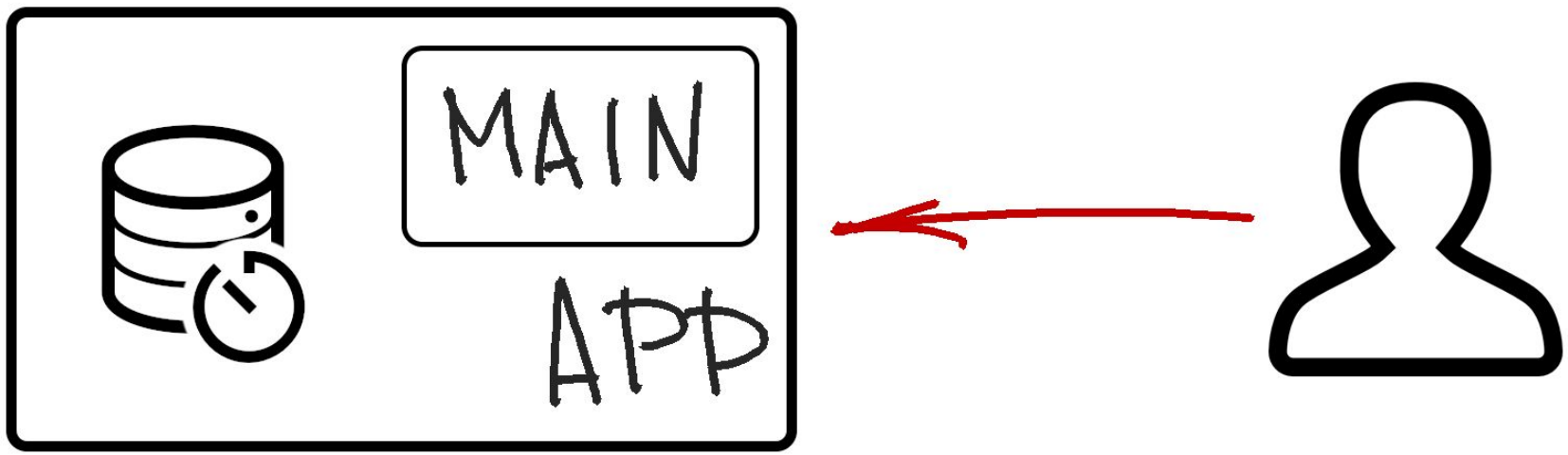
и т. д.

Writing policies

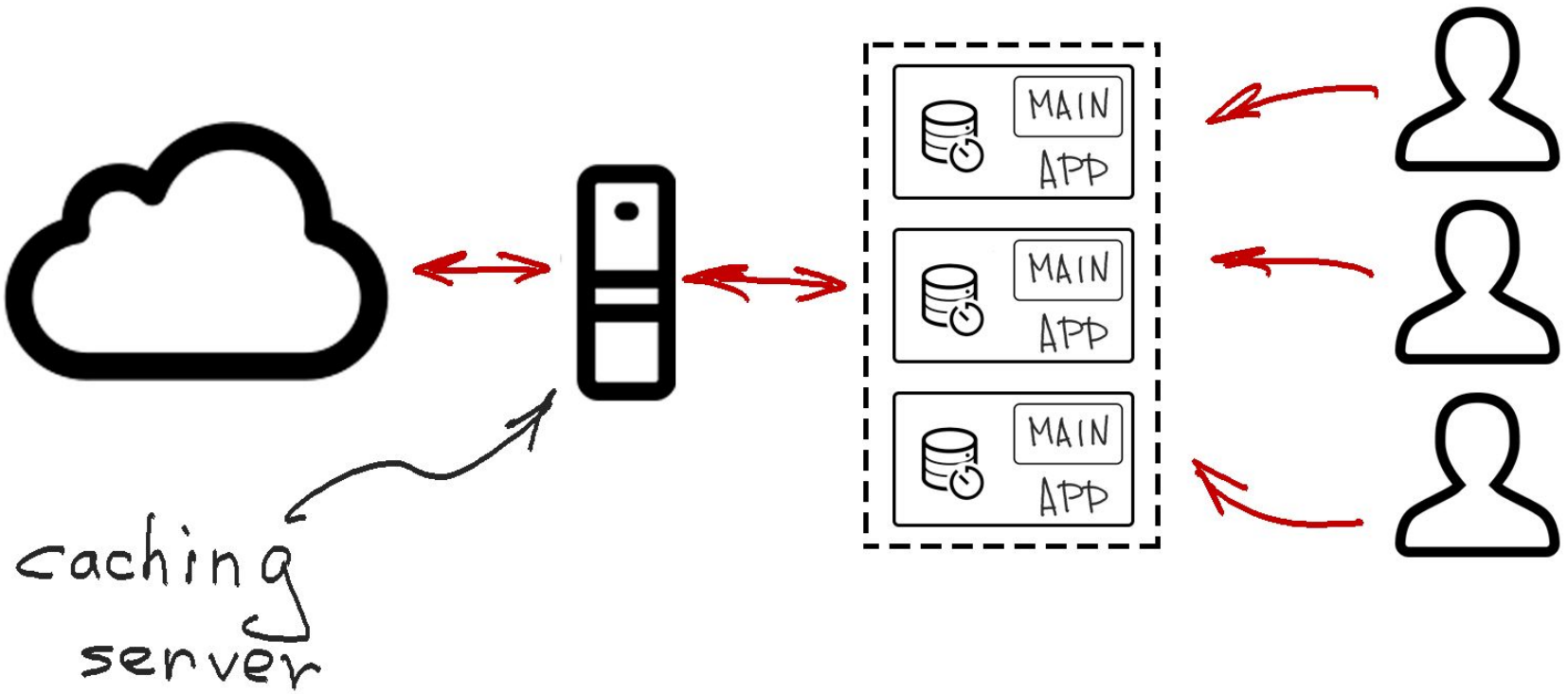
- сквозная запись (write-through)
- отложенная запись (write-back)

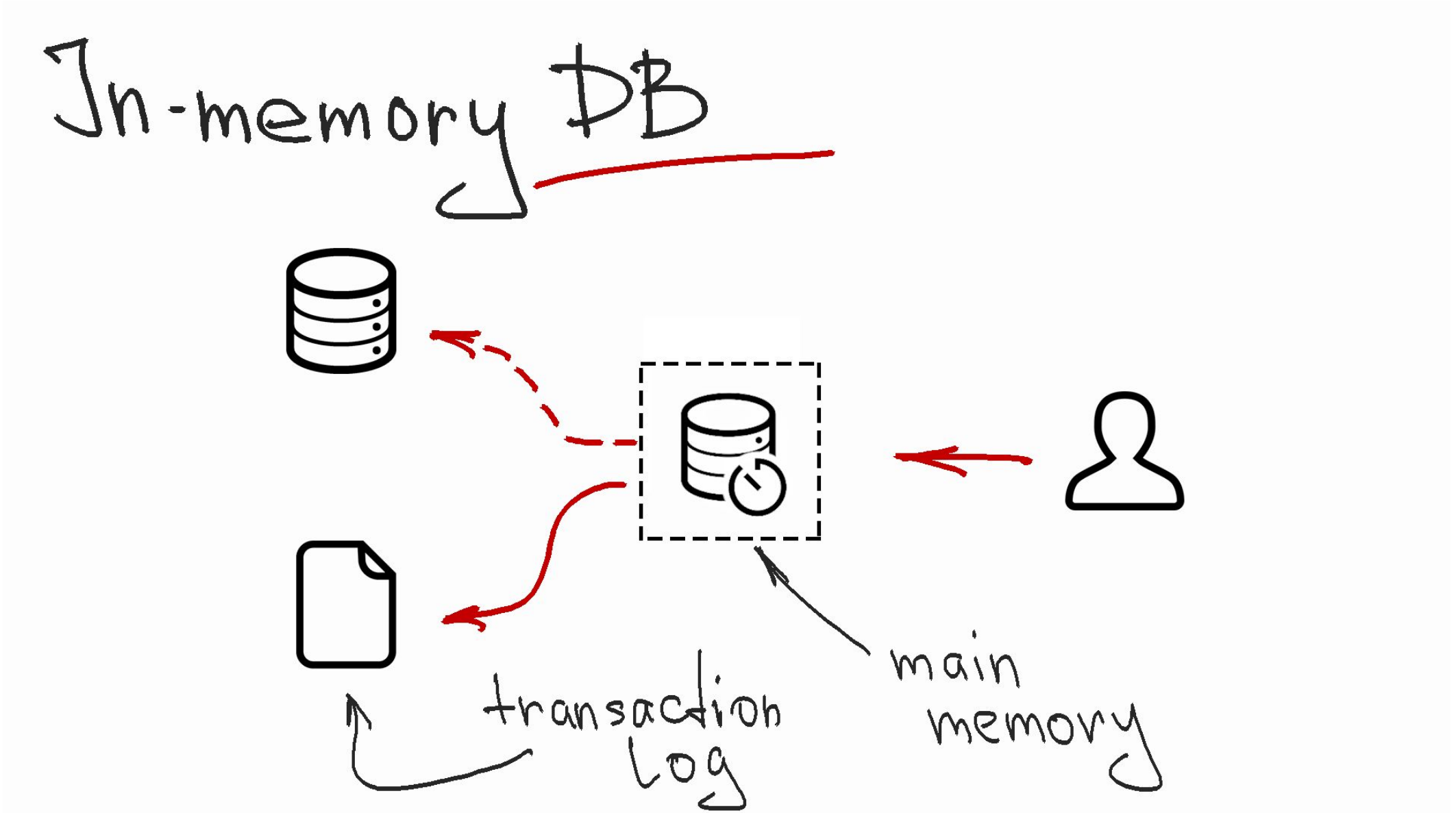
Виды кэшей

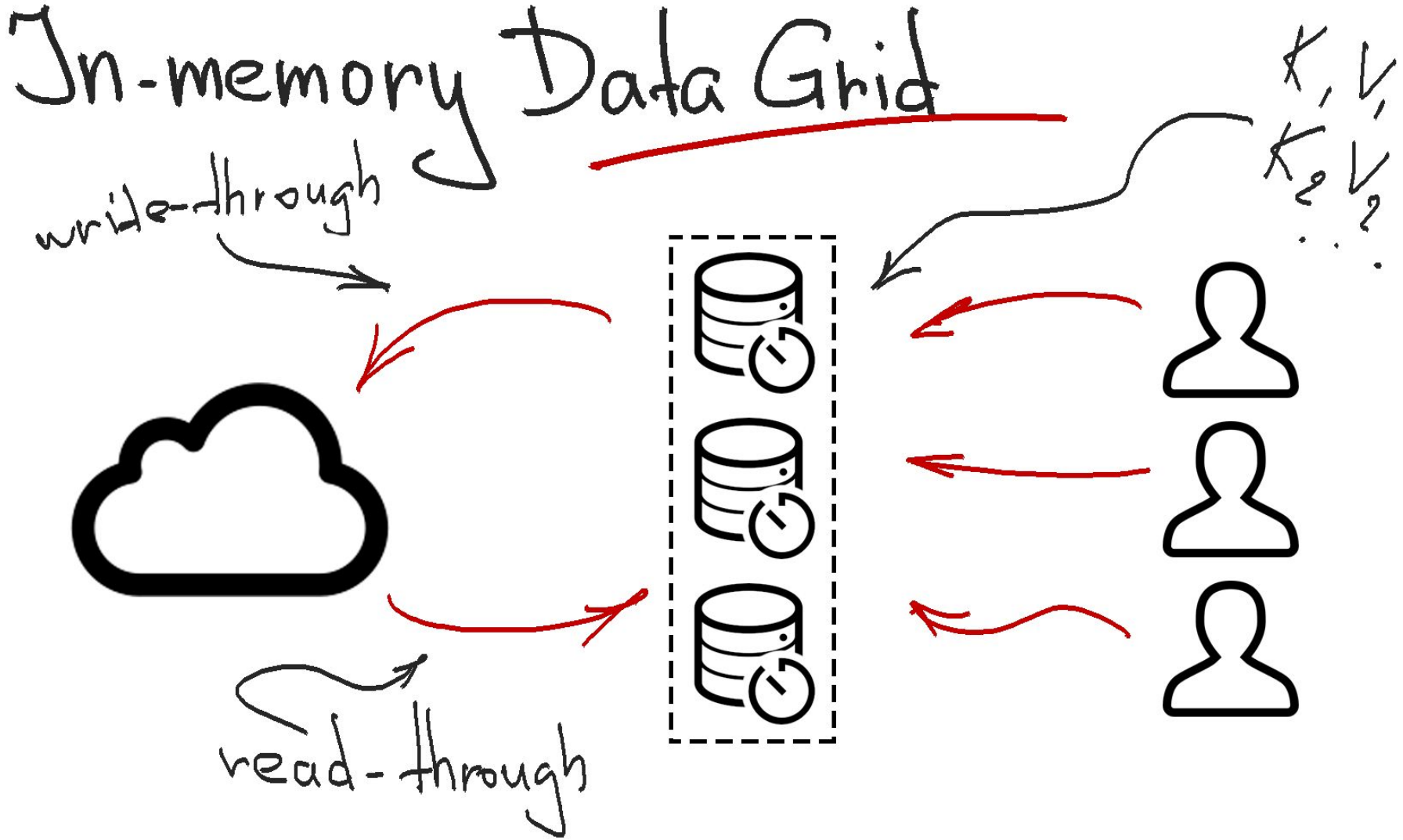
In-process



Distributed



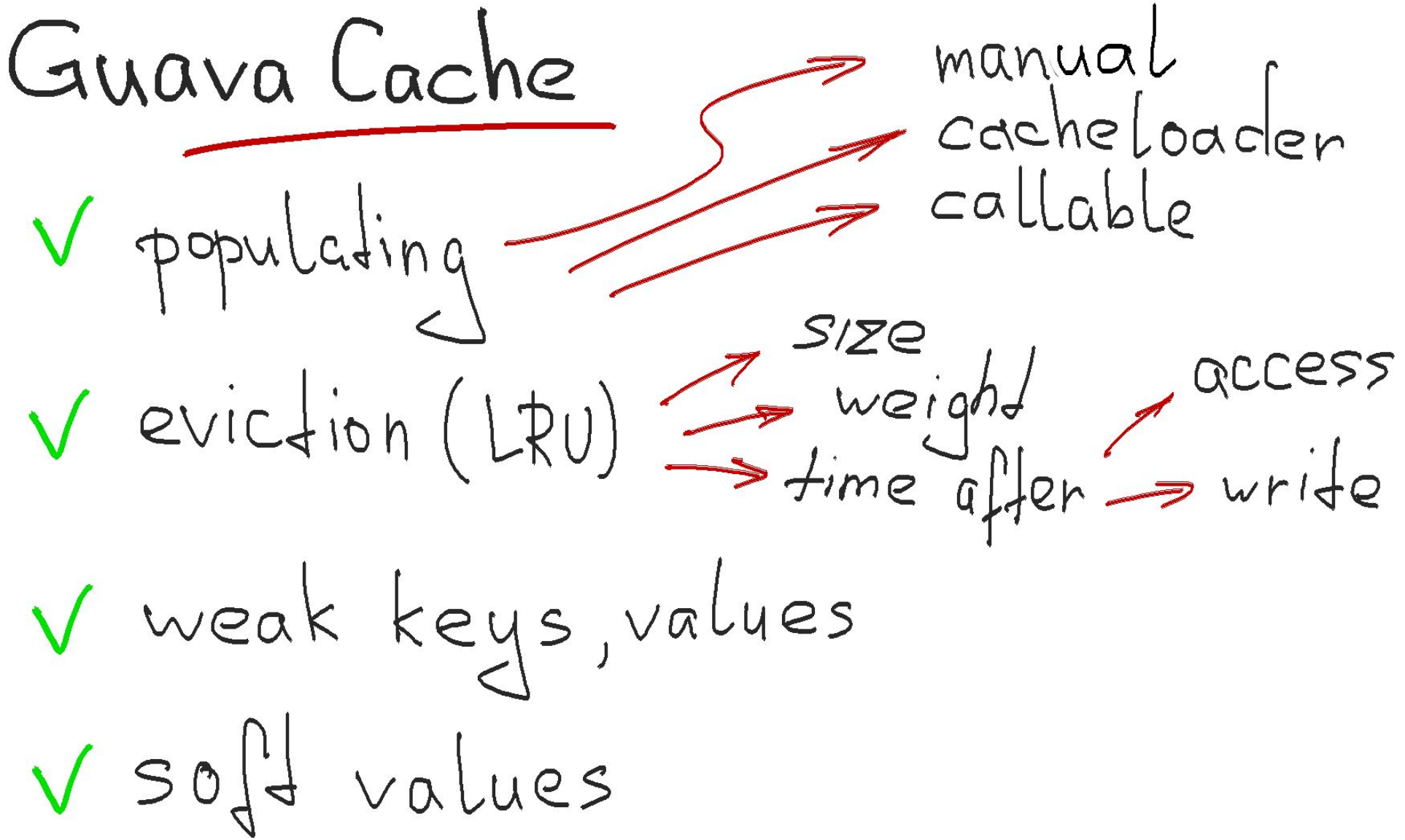

In-memory DB
transaction log
main memory


In-memory Data Grid
write-through
read-through


Библиотеки

Guava Cache

Guava Cache


manual
cacheloader
callable
populating
eviction (LRU)
size
weight
time after
access
write
weak keys, values
soft values

Guava Cache

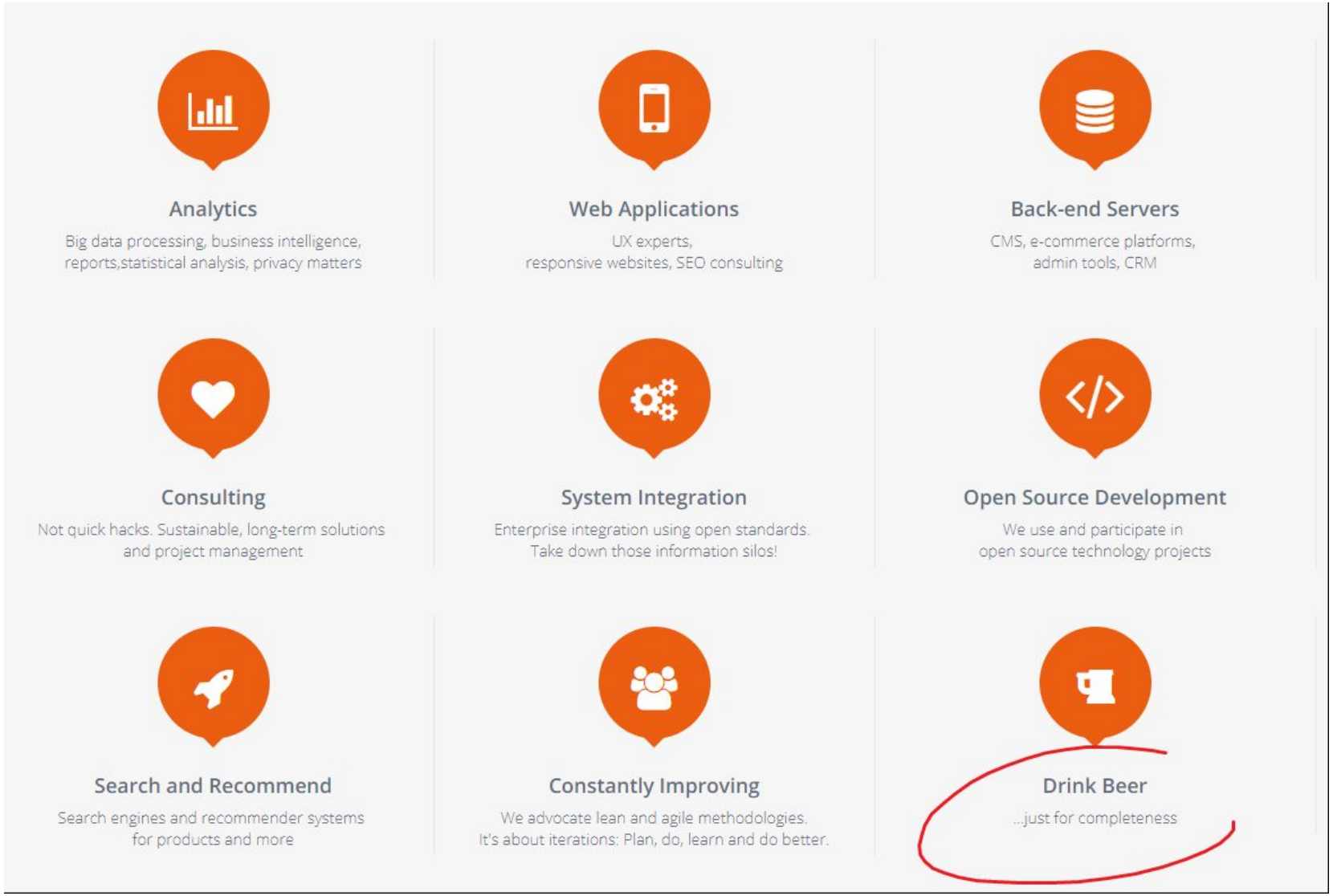
- ✓ refresh cache or entry
- ✓ removal Listener
- ✓ manual invalidation
- ✓ thread-safe

Guava Cache

- ✗ null values
- ✗ auto clean up


Caffeine

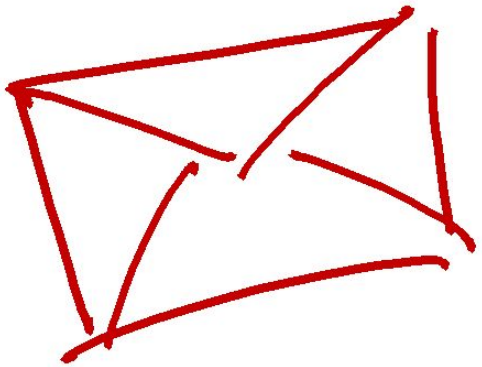

Cache2k


Analytics
Big data processing, business intelligence, reports,statistical analysis, privacy matters
Web Applications
UX experts, responsive websites, SEO consulting
Back-end Servers
CMS, e-commerce platforms, admin tools, CRM
Consulting
Not quick hacks. Sustainable, long-term solutions and project management
System Integration
Enterprise integration using open standards. Take down those information silos!
Open Source Development
We use and participate in open source technology projects
Search and Recommend
Search engines and recommender systems for products and more
Constantly Improving
We advocate lean and agile methodologies. It's about iterations: Plan, do, learn and do better.
Drink Beer
...just for completeness

Cache2k

- ✓ listeners → update, insert, expiry
- ✓ refresh ahead
- ✓ null keys, values


performance



LKivattsev

@ Luxoft.com


Вопросы?