

The background features a cityscape at night, likely Hong Kong, with illuminated skyscrapers and a harbor. Overlaid on this are several large, semi-transparent, 3D-style arrow shapes pointing to the right. These arrows are composed of multiple parallel lines, some of which are glowing with bright blue and cyan light, while others are more muted. The overall aesthetic is futuristic and high-tech.

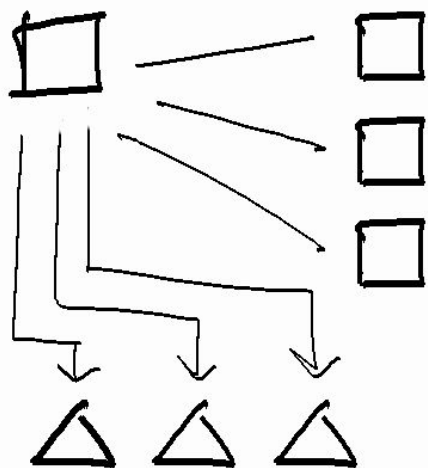
Luxoft

think.
create.
accelerate.

Кэширование в Java

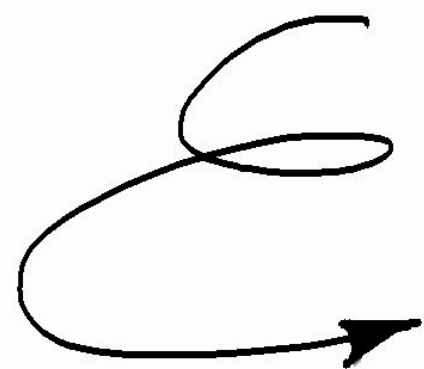
Метадоклад с метасоветами

Почему?



#

Зачем?

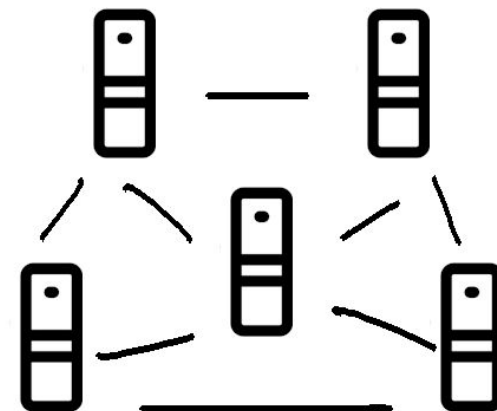
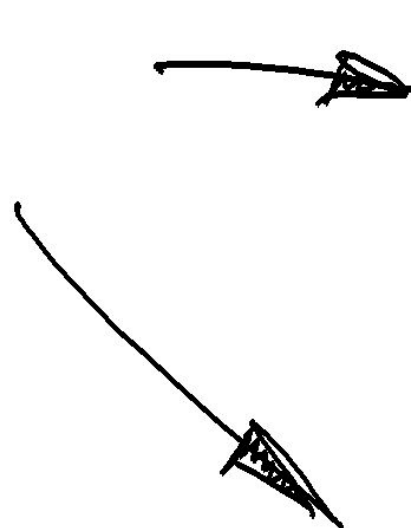
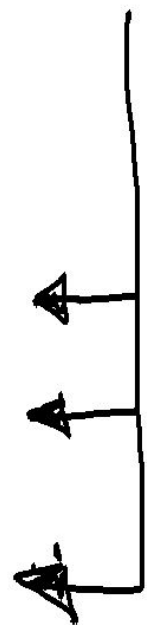
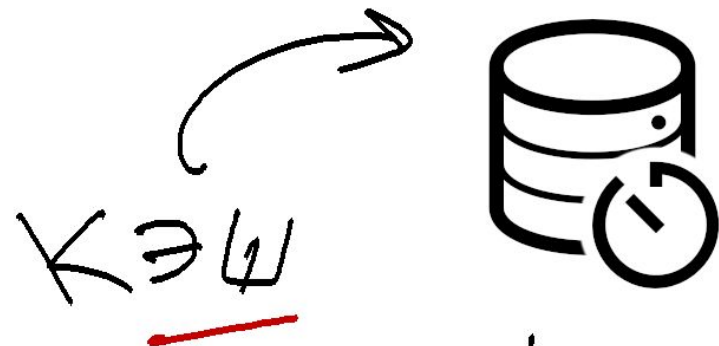


LEVEL
UP ↑

ЧТО ТАКОЕ
КЭШ?

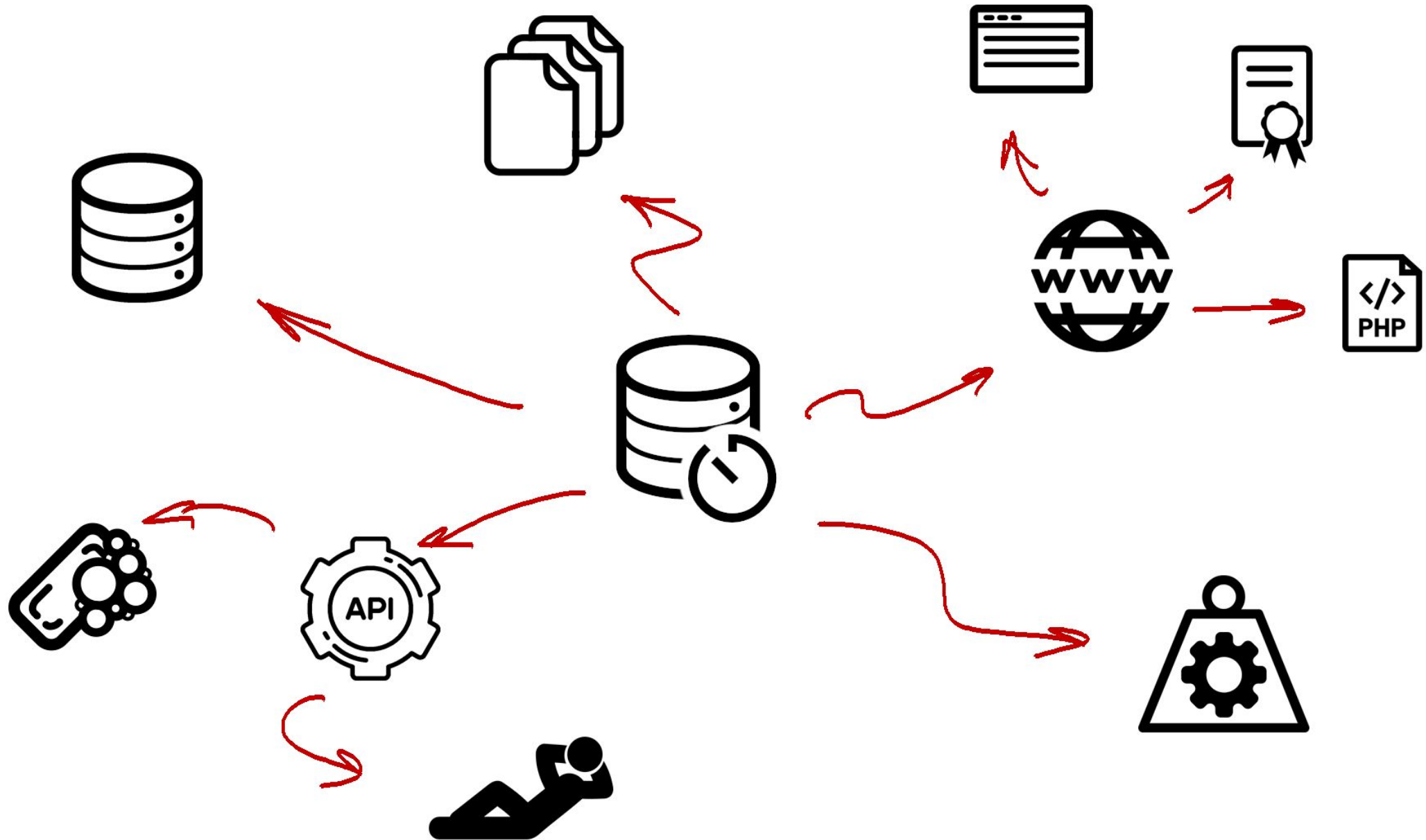


- быстрый
- доступный
- временный



Что?





Анаго ли?



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

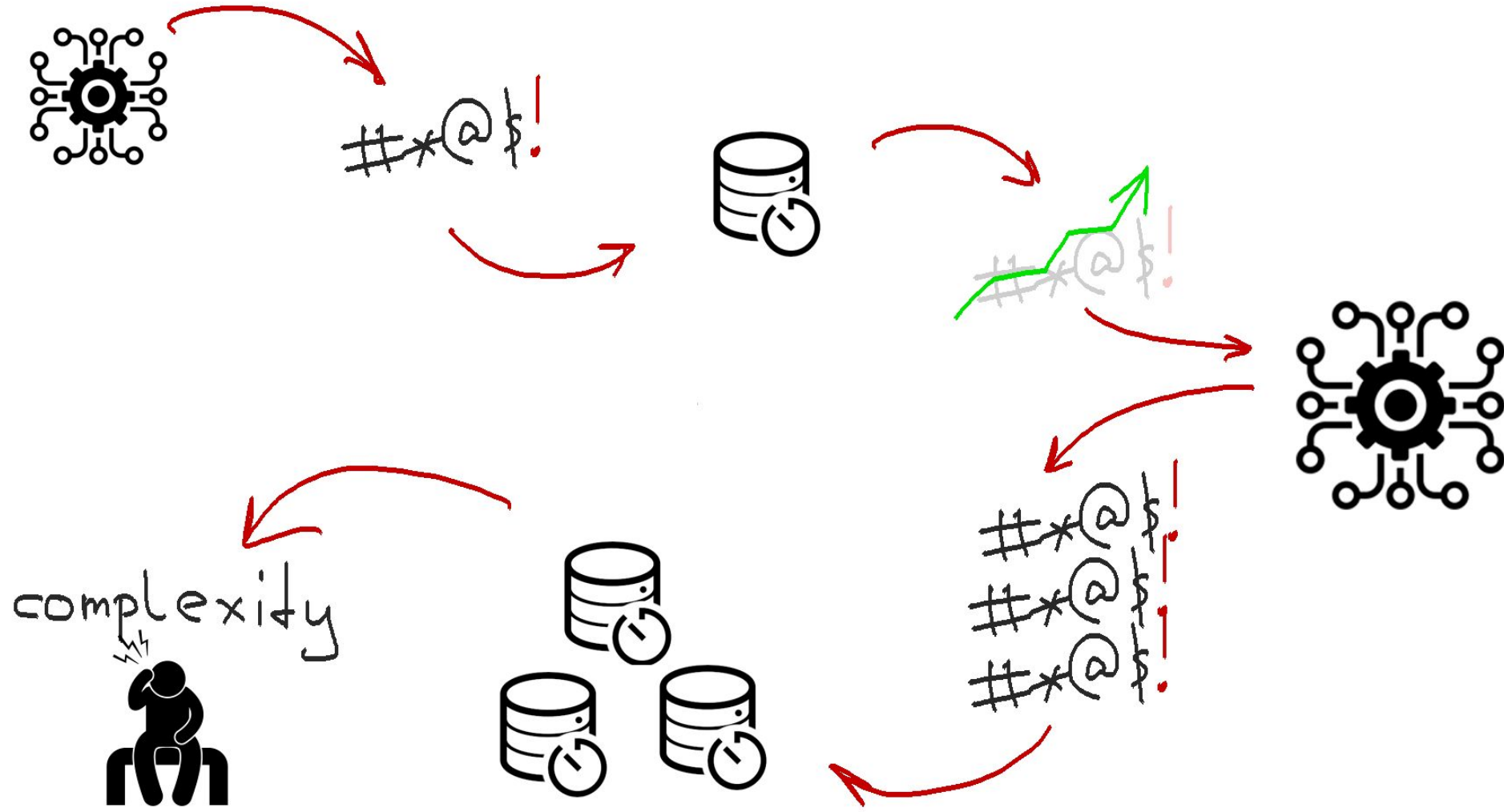
D. Knuth

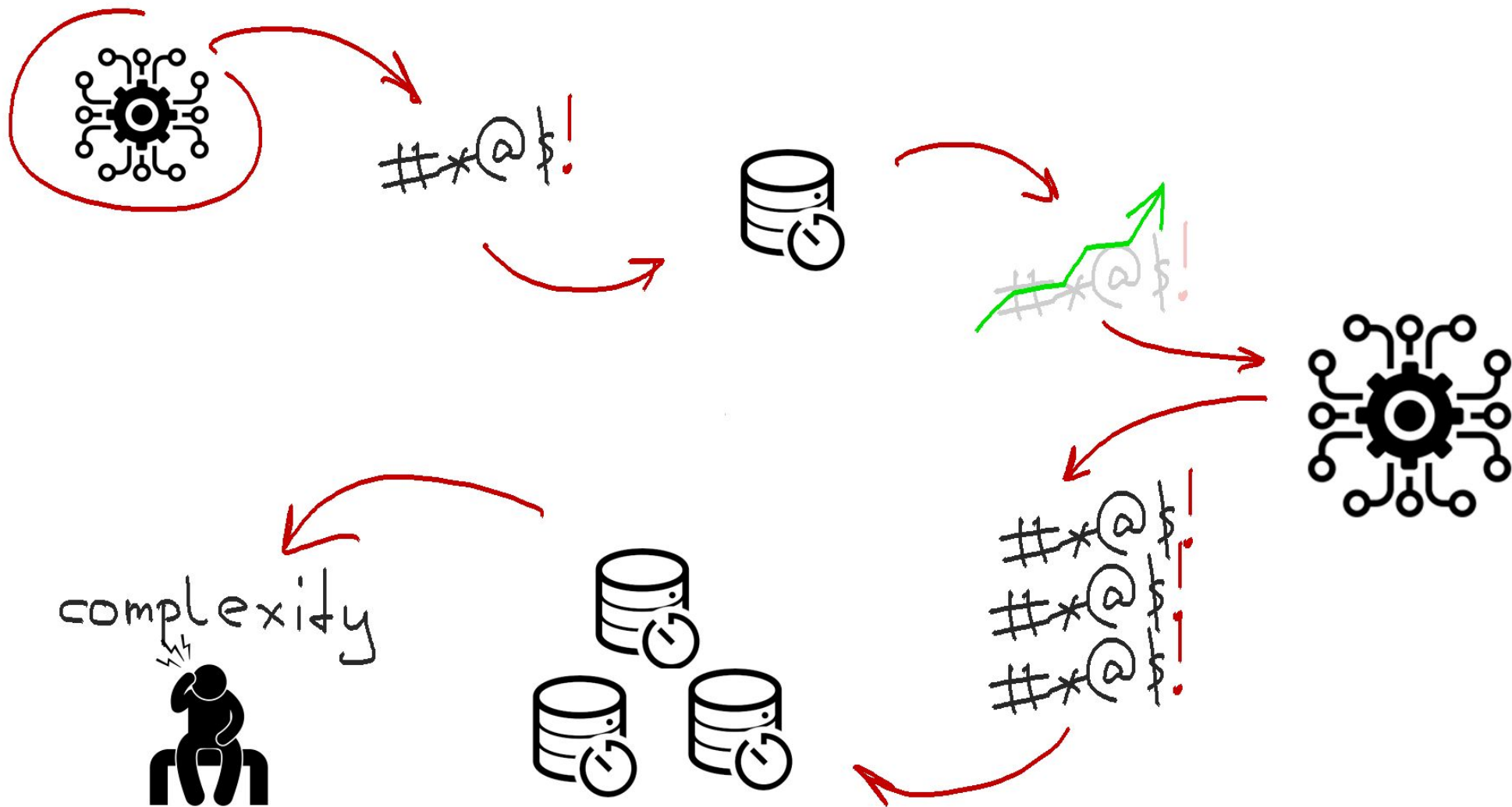
$$A_u + A_c + A_n < \text{value}$$

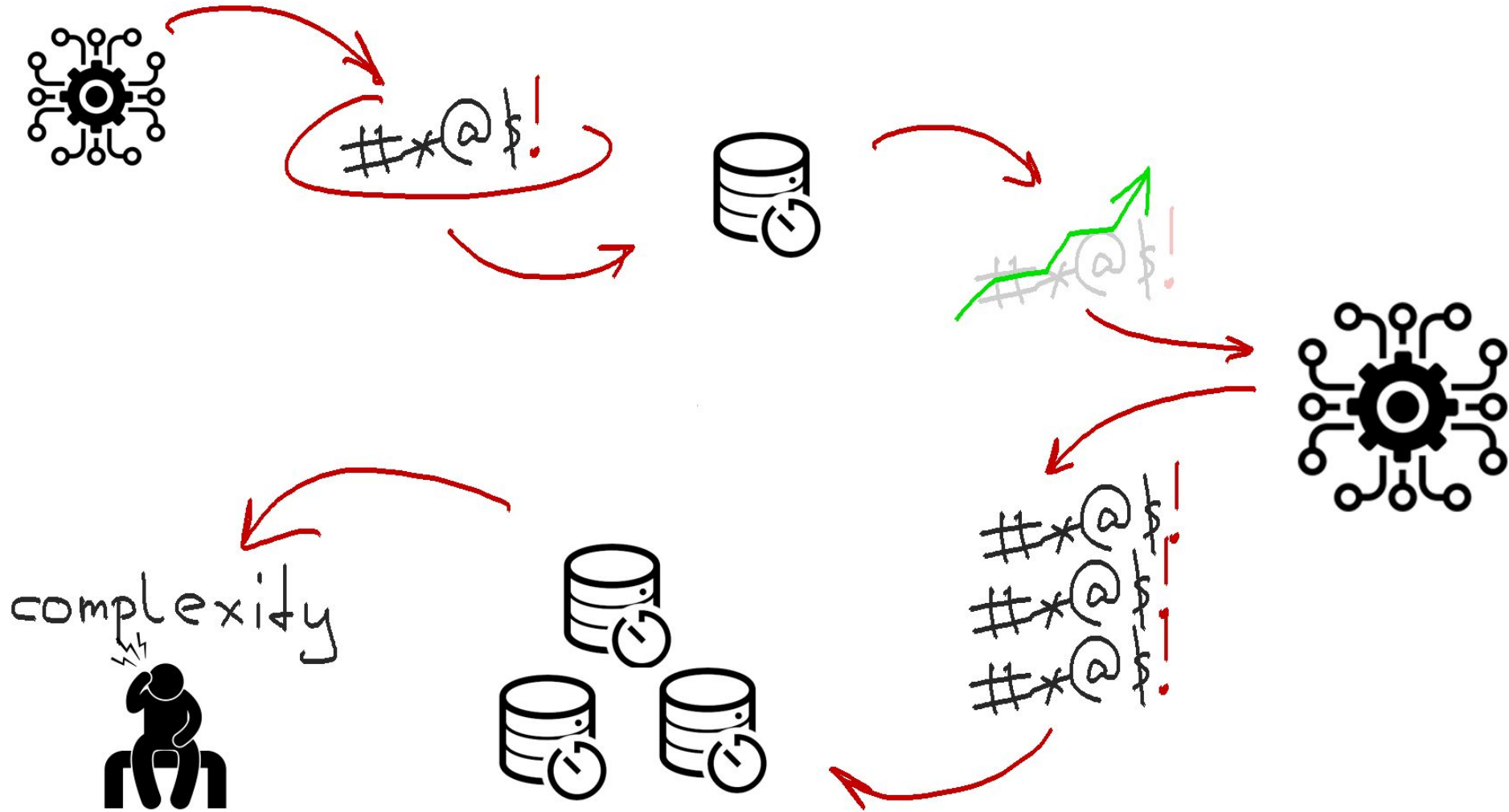
изучение

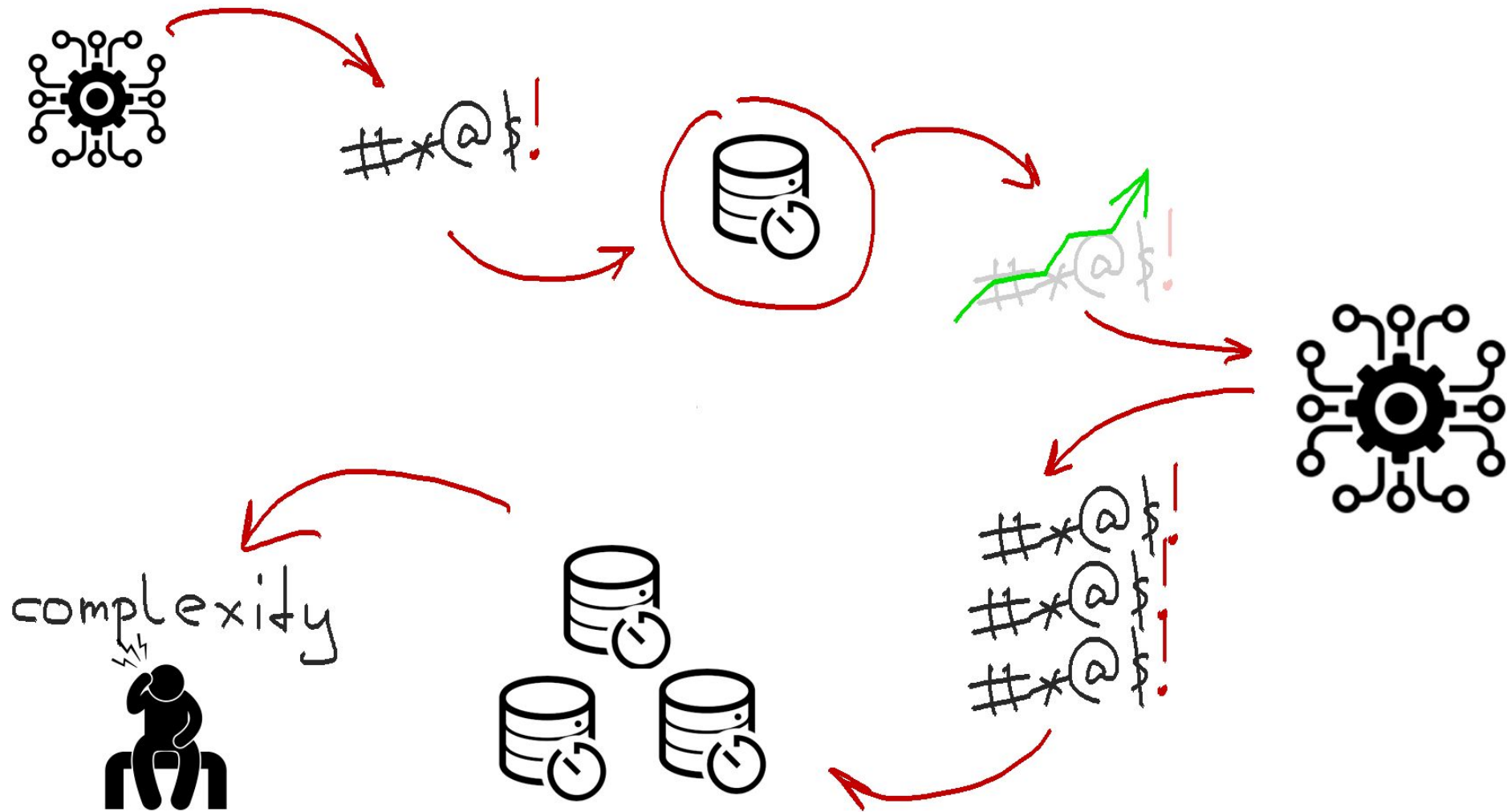
сопровождение

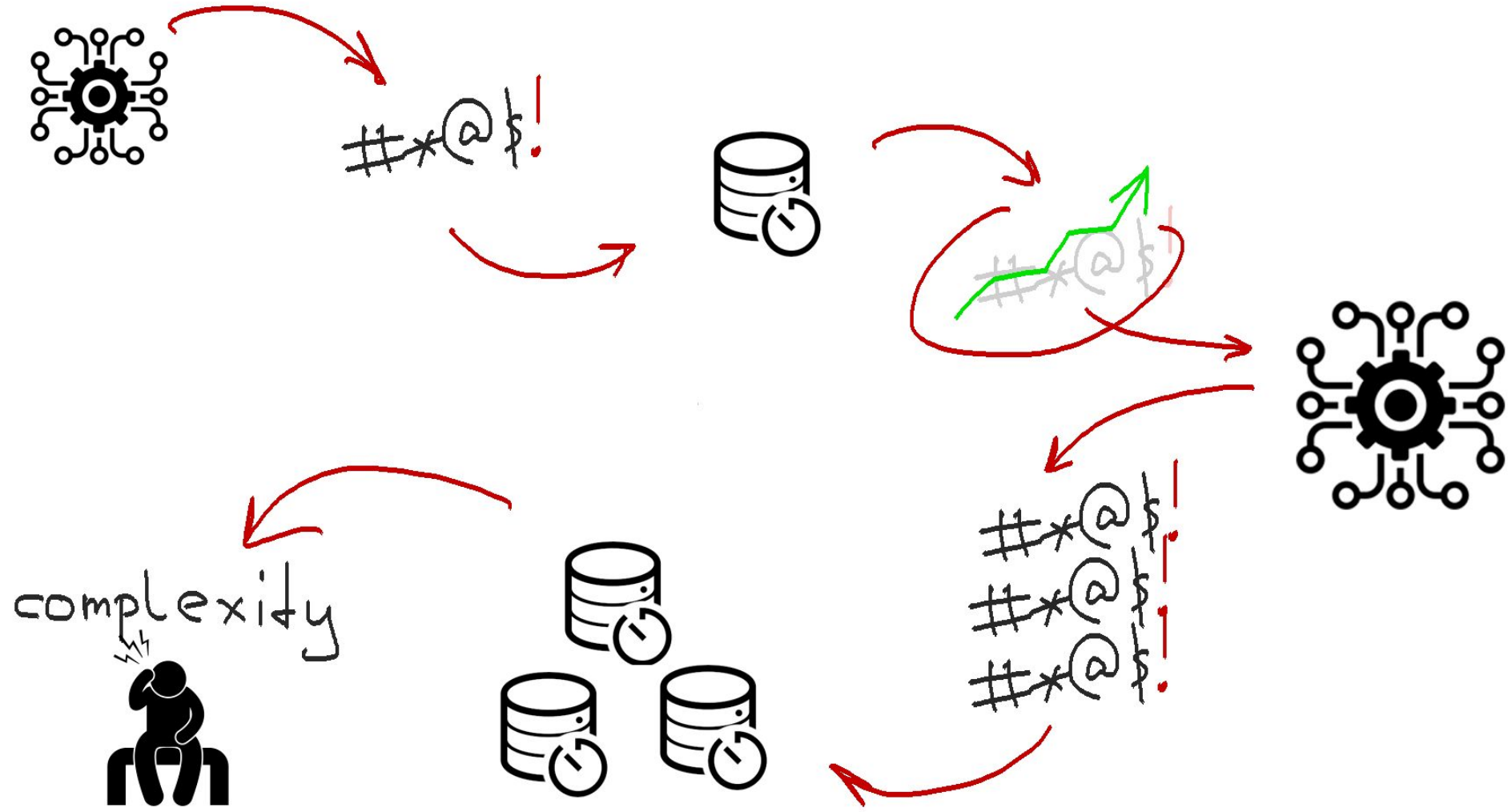
поддержка

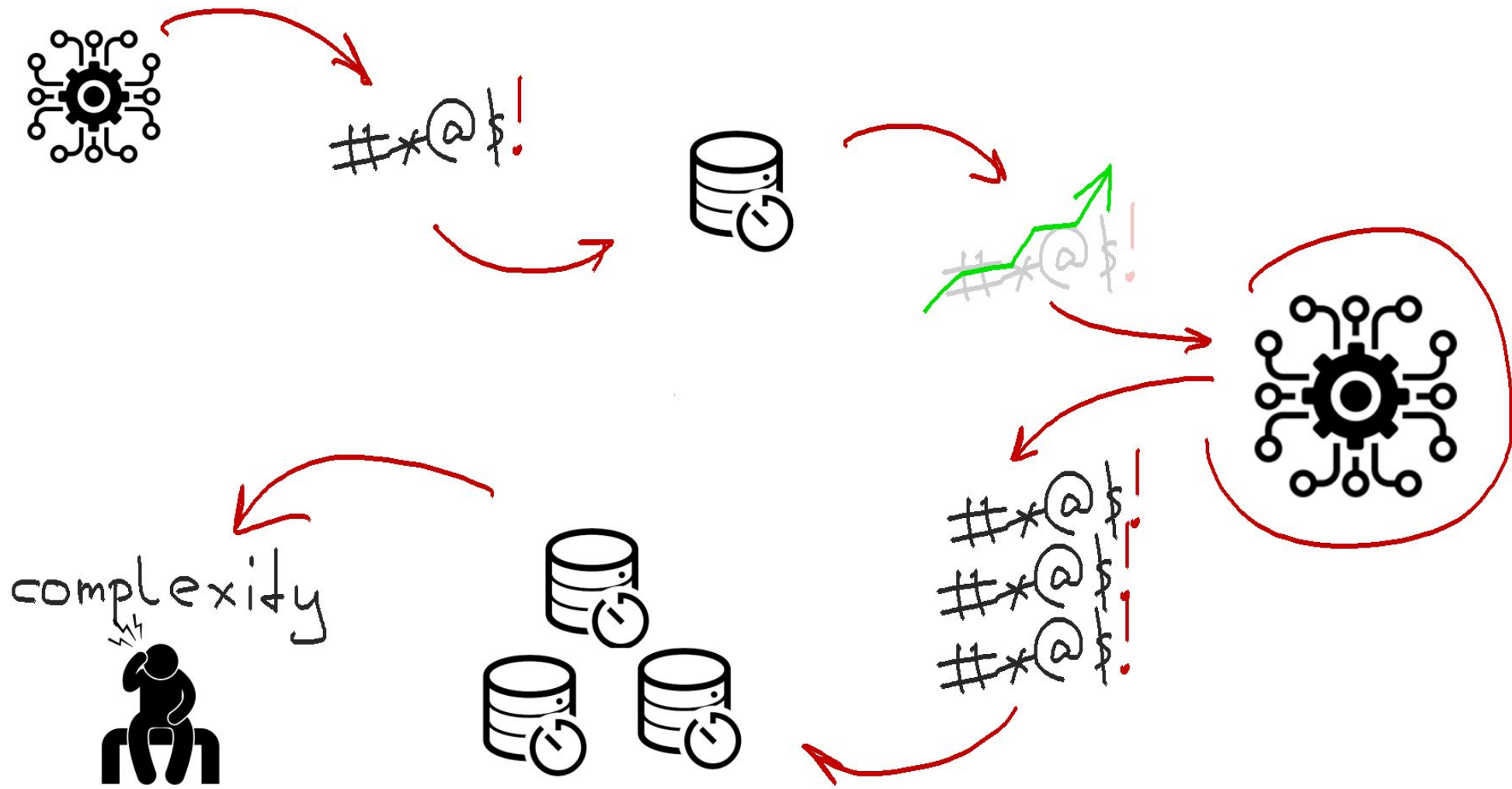


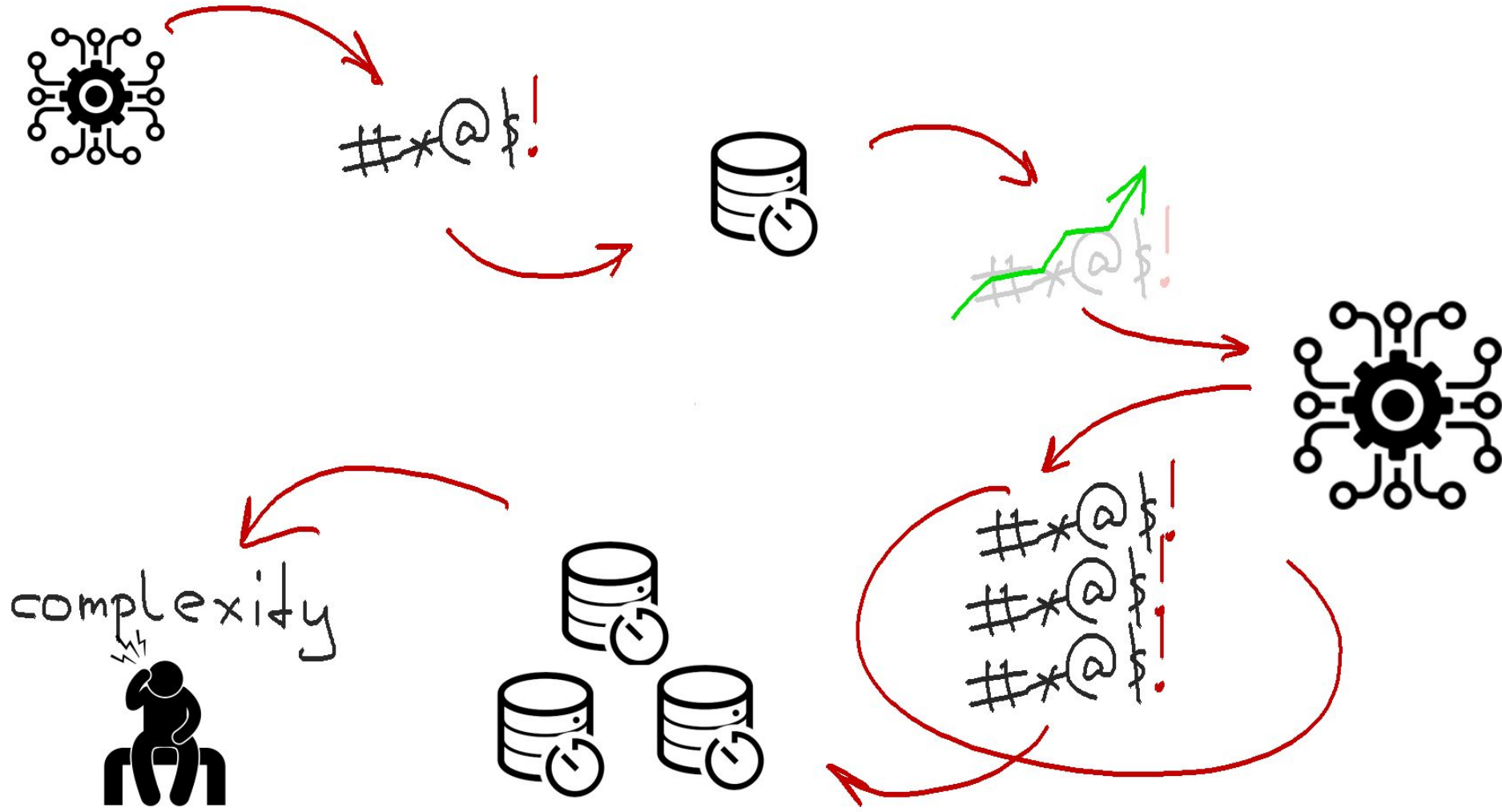


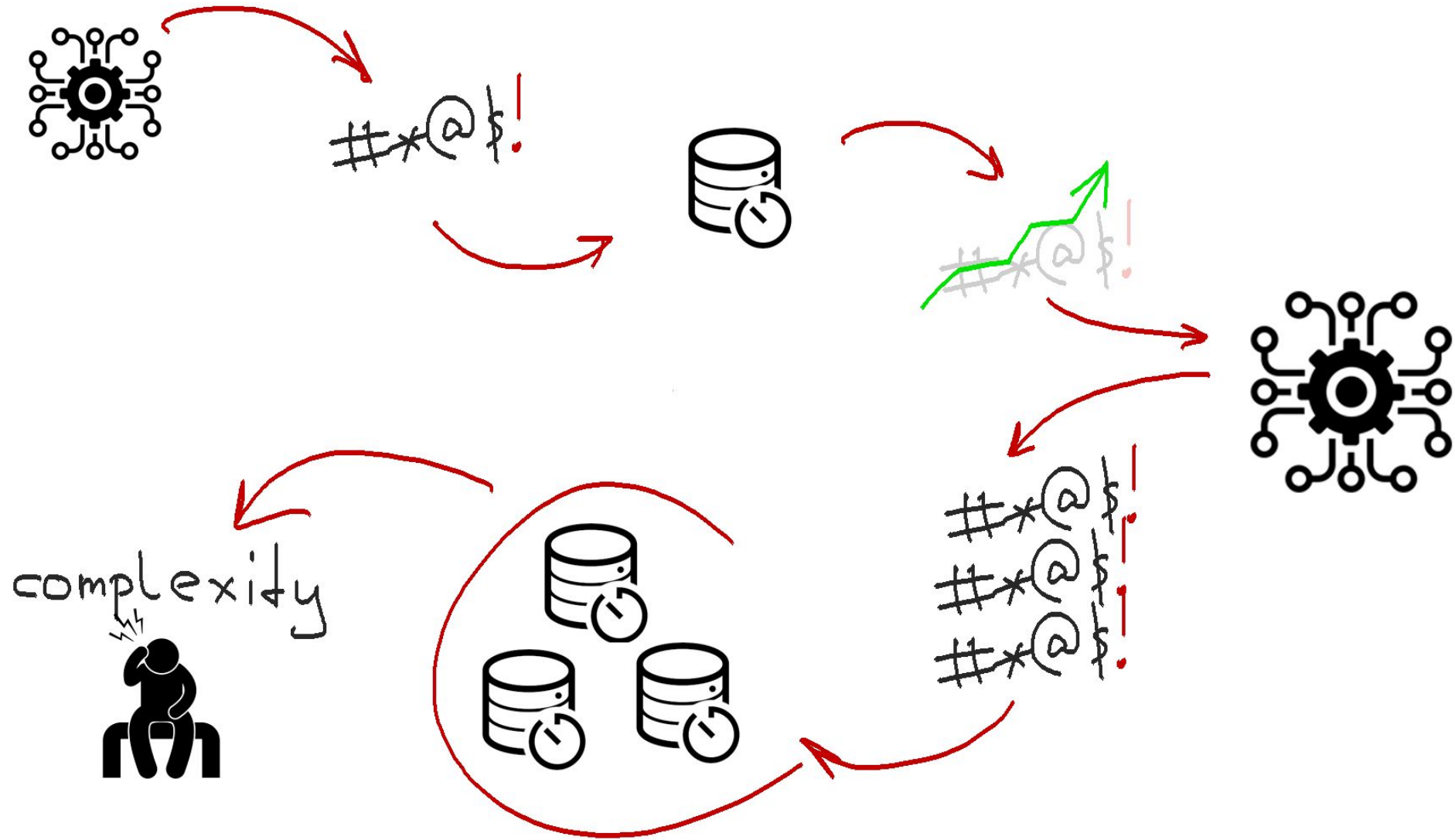


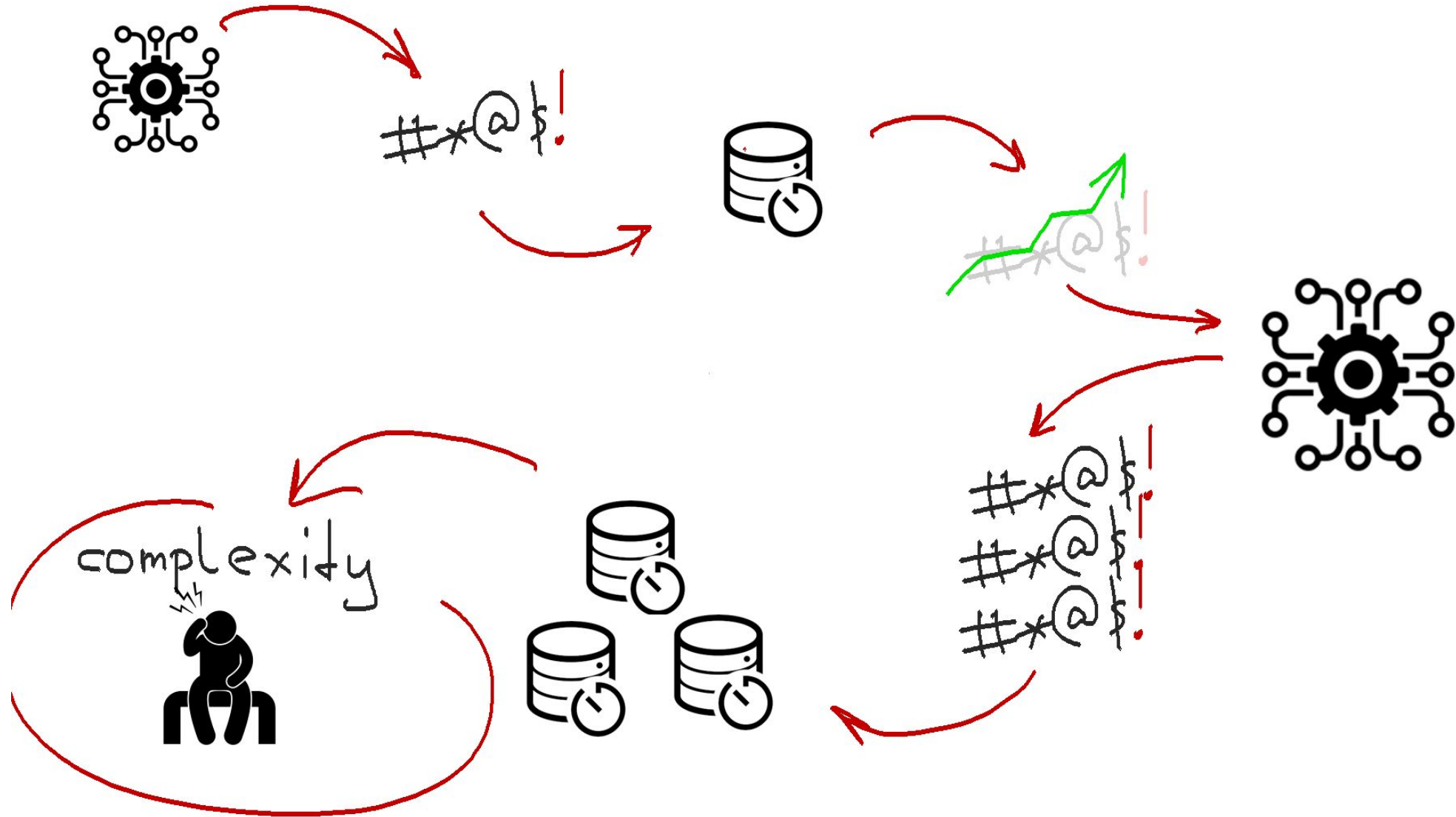













There are only two hard things
in Computer Science:
cache invalidation and
naming things.

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

```
graph TD; A[улучшение производительности] --> B[масштабируемости]
```

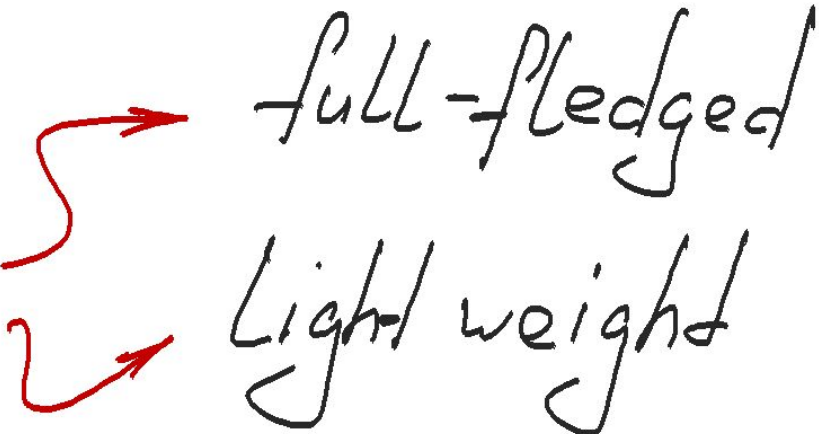
② Как неактуальные данные повлияют на работу системы / пользователя?

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?

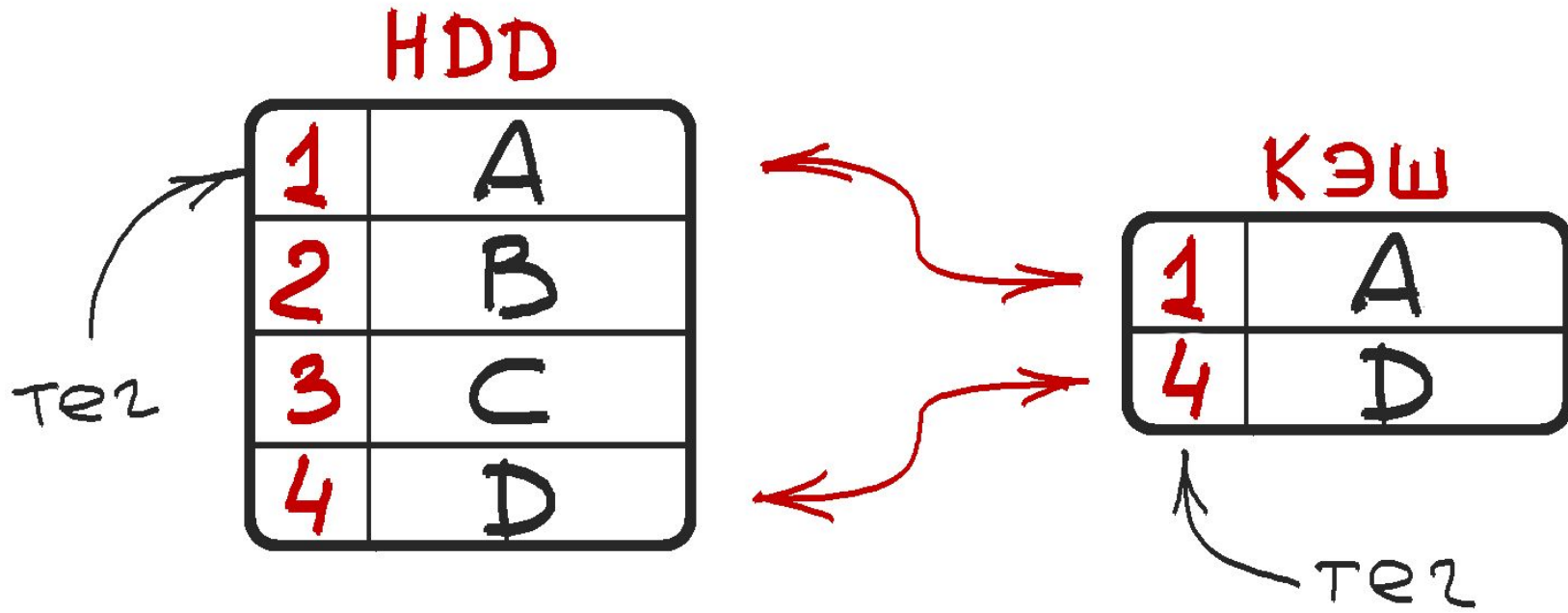
③ Тип кэша? In-process, distributed?

④ Производительность? Надежность?

и т.д.

Терминология

Tez



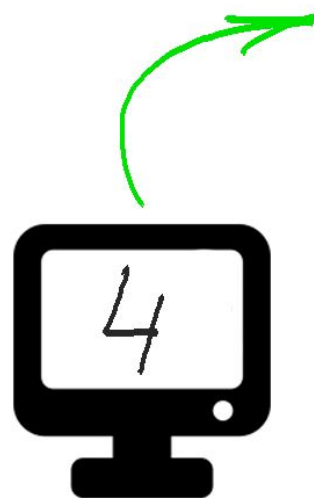
Попадание (hit)

HDD

1	A
2	B
3	C
4	D

КЭШ

1	A
4	D



Промис (miss)

HDD

1	A
2	B
3	C
4	D

КЭШ

1	A
4	D



Метрики

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

u T.g.

Eviction policies

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

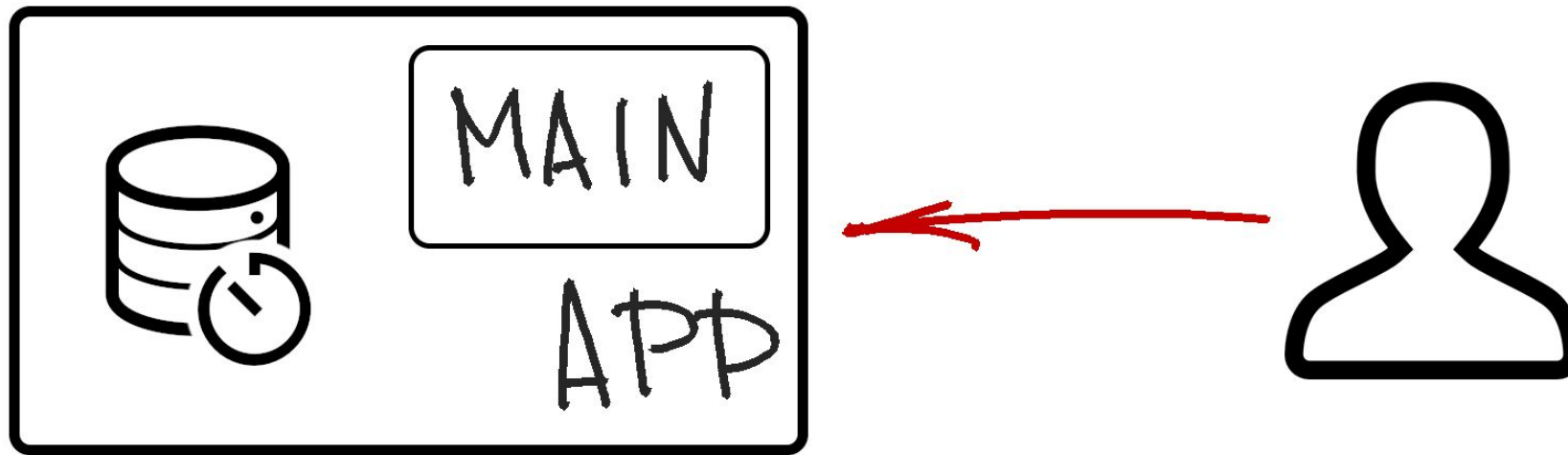
u.t.g.

Writing policies

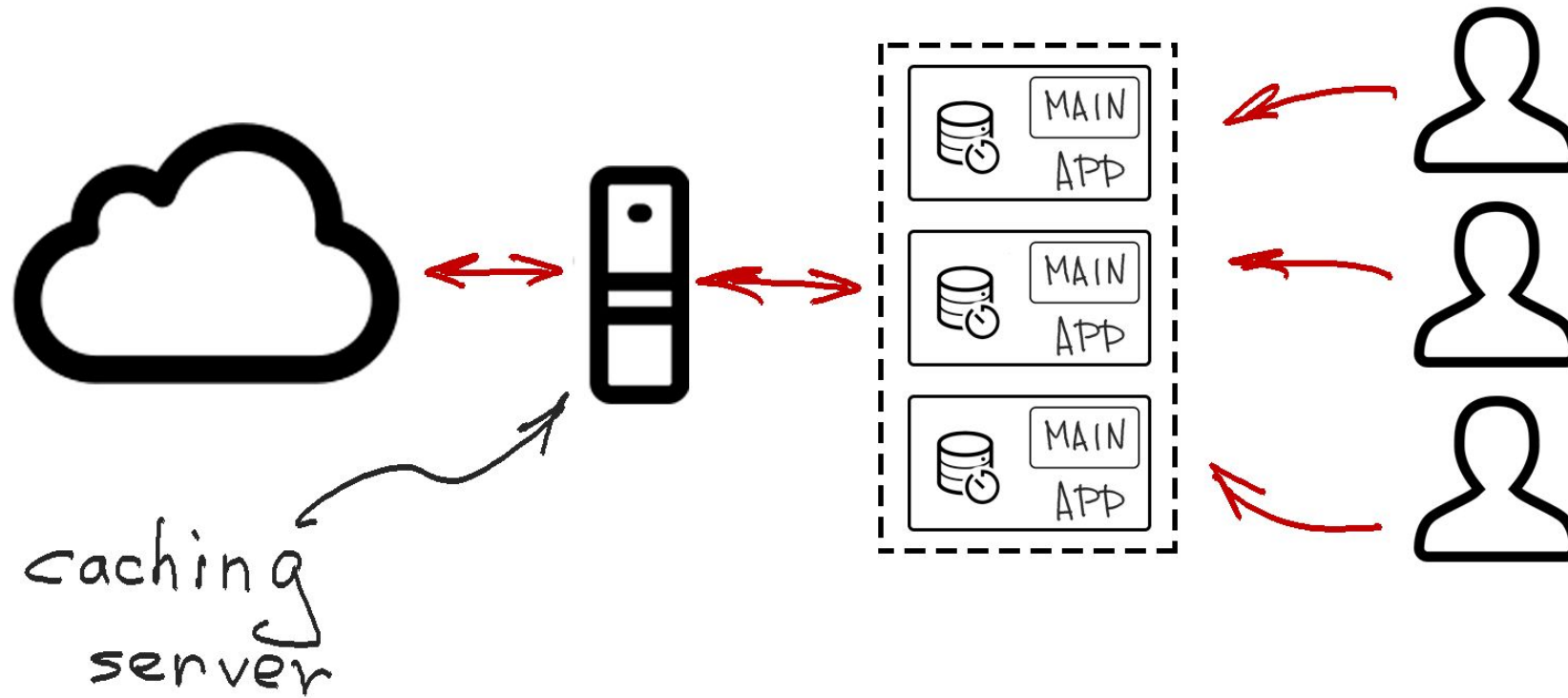
- сквозная запись (write-through)
- отложенная запись (write-back)

Бүгд кэшеи

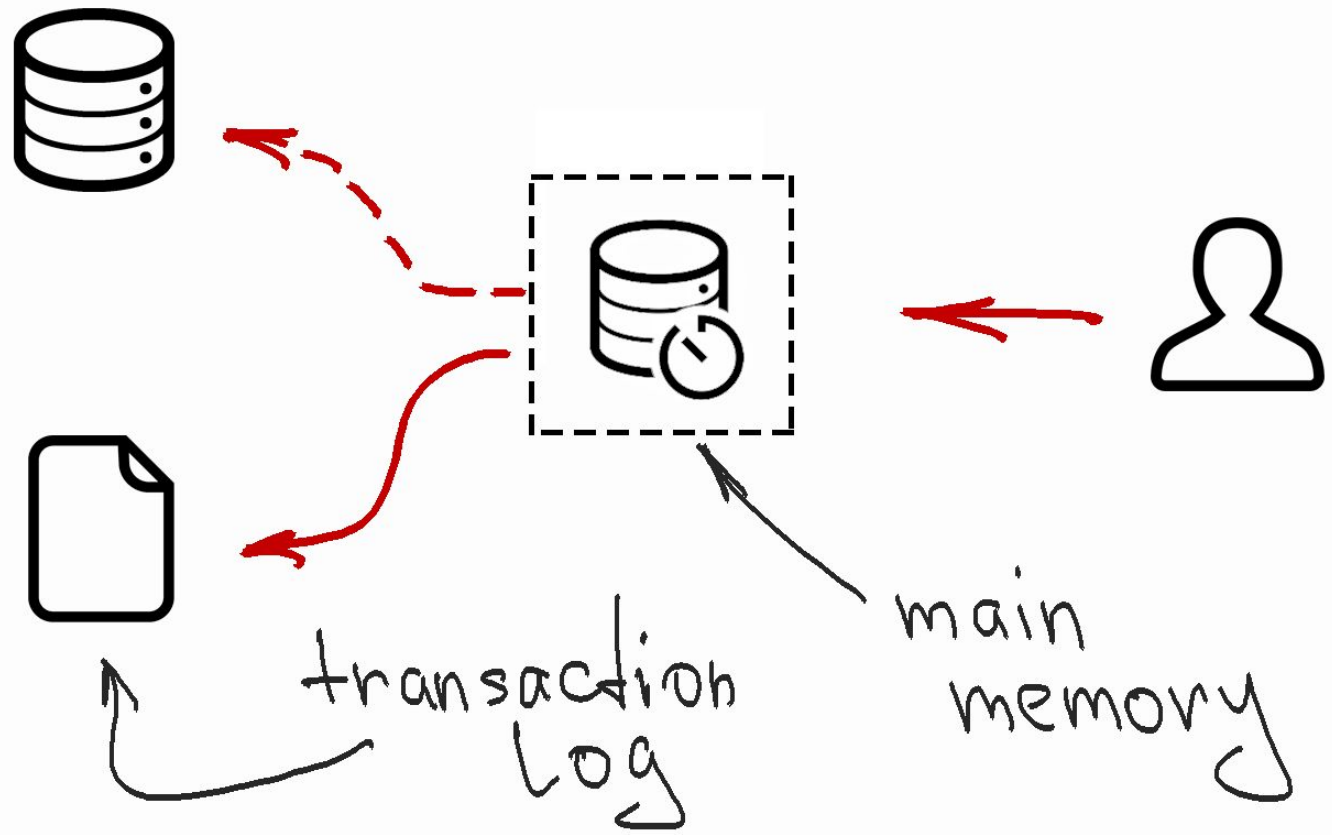
In-process



Distributed



In-memory DB

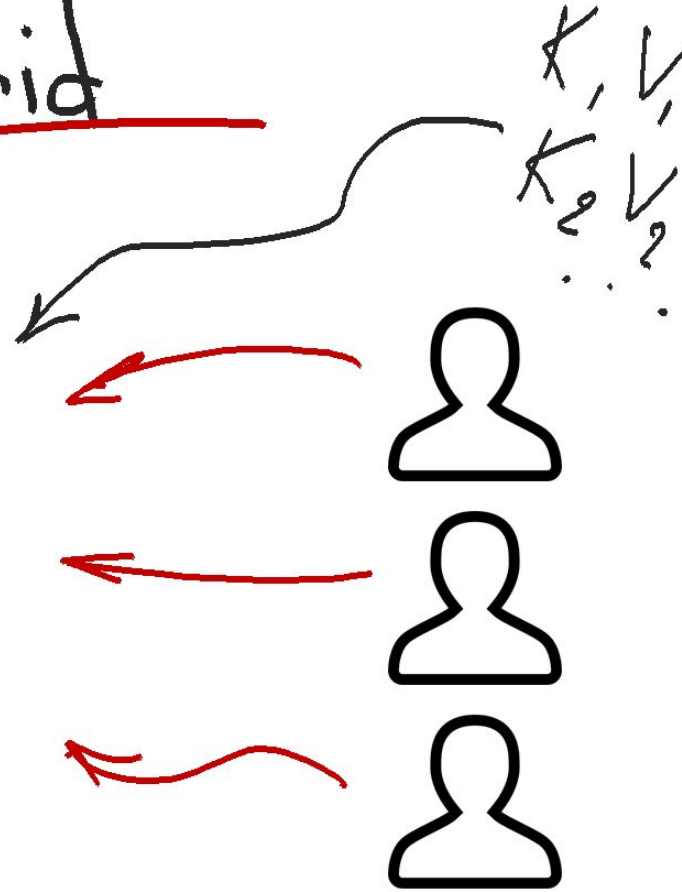
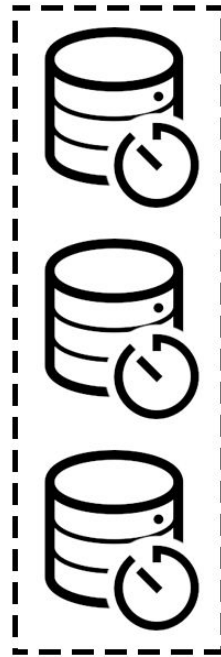


In-memory Data Grid

write-through



read-through



Библиотеки

Guava Cache

Guava Cache

✓ populating

manual
cache loader
callable

✓ eviction (LRU)

size
weight
time after
access
write

✓ weak keys, values

✓ soft values

Guava Cache

- ✓ refresh cache on 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

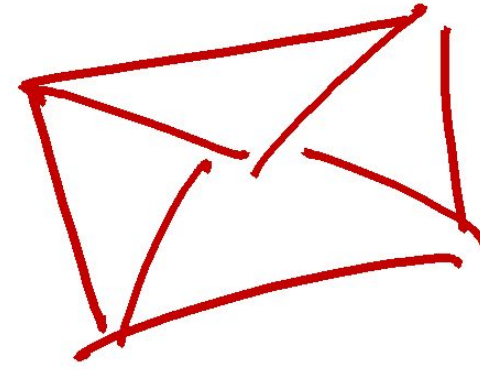
...just for completeness

Cache2k

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

performance 





L. Kivattsev
@ Luxoft.com

Вопросы?