Chproxy - ClickHouse Proxy

CONTENTSQUARE

Who are we?

Paweł Gontarz

Data Engineer

François Milhem

Data Engineer

ClickHouse @ Contentsquare

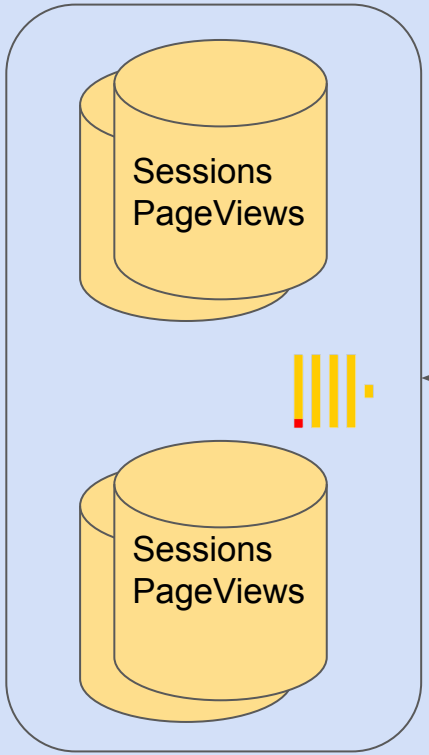
Data @ Contentsquare

Sessions

session_number	session_time	device_id	session_duration_msec
1	2022-01-25 23:06:36	2	1154741

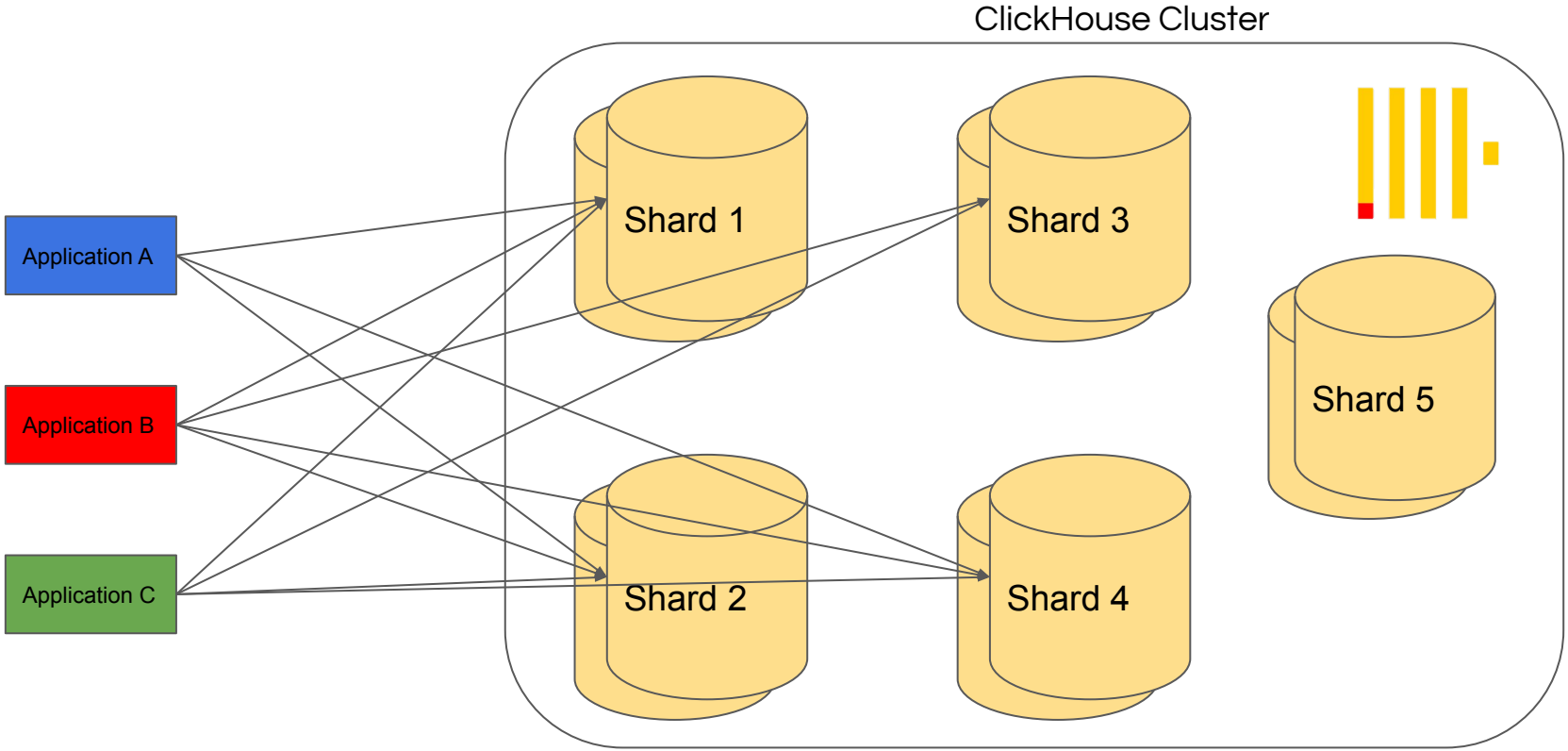
Views

session_number	session_time	device_id	view_duration_msec
1	2022-01-24 20:27:14	4	0

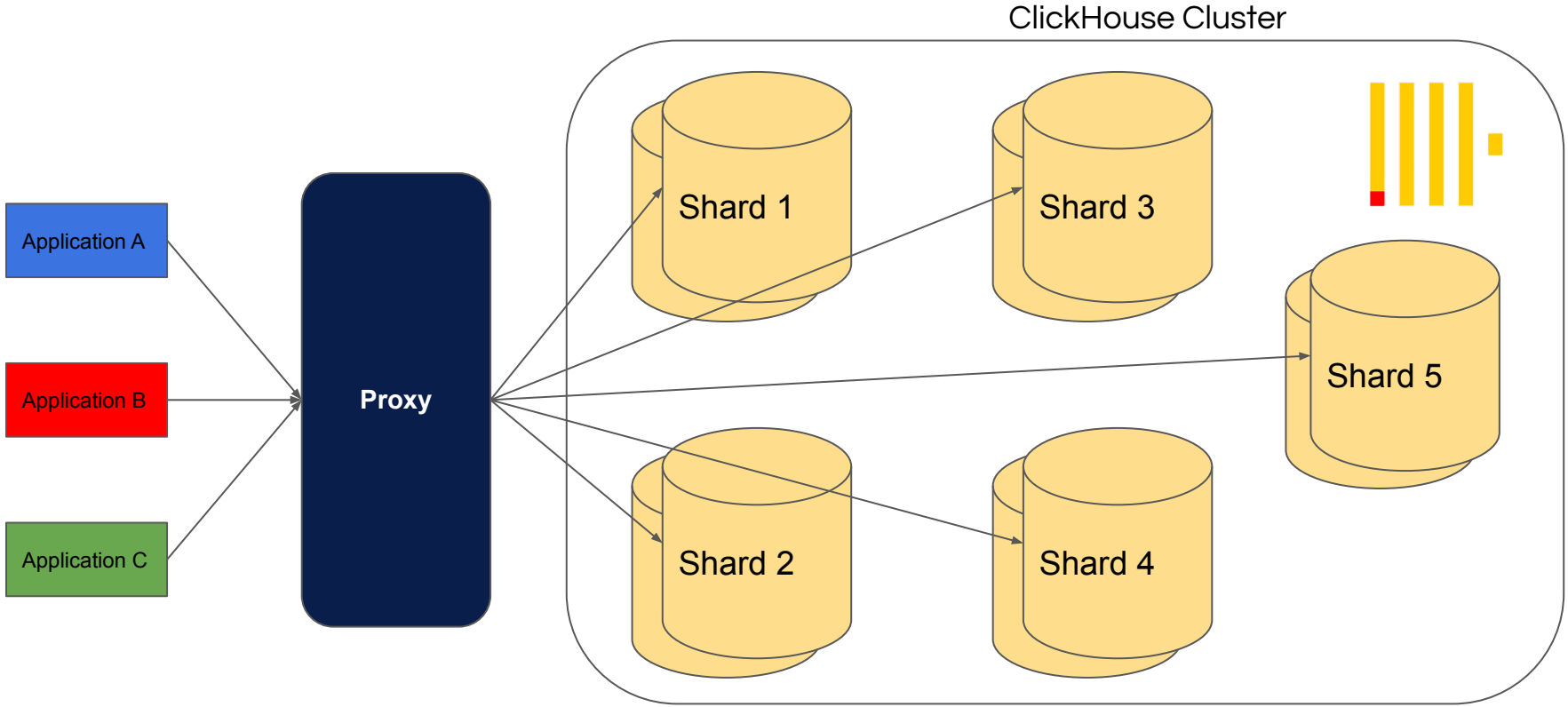



```
SELECT
avg(session_duration)
FROM
views
WHERE session_date
>= …
```

ClickHouse cluster topology



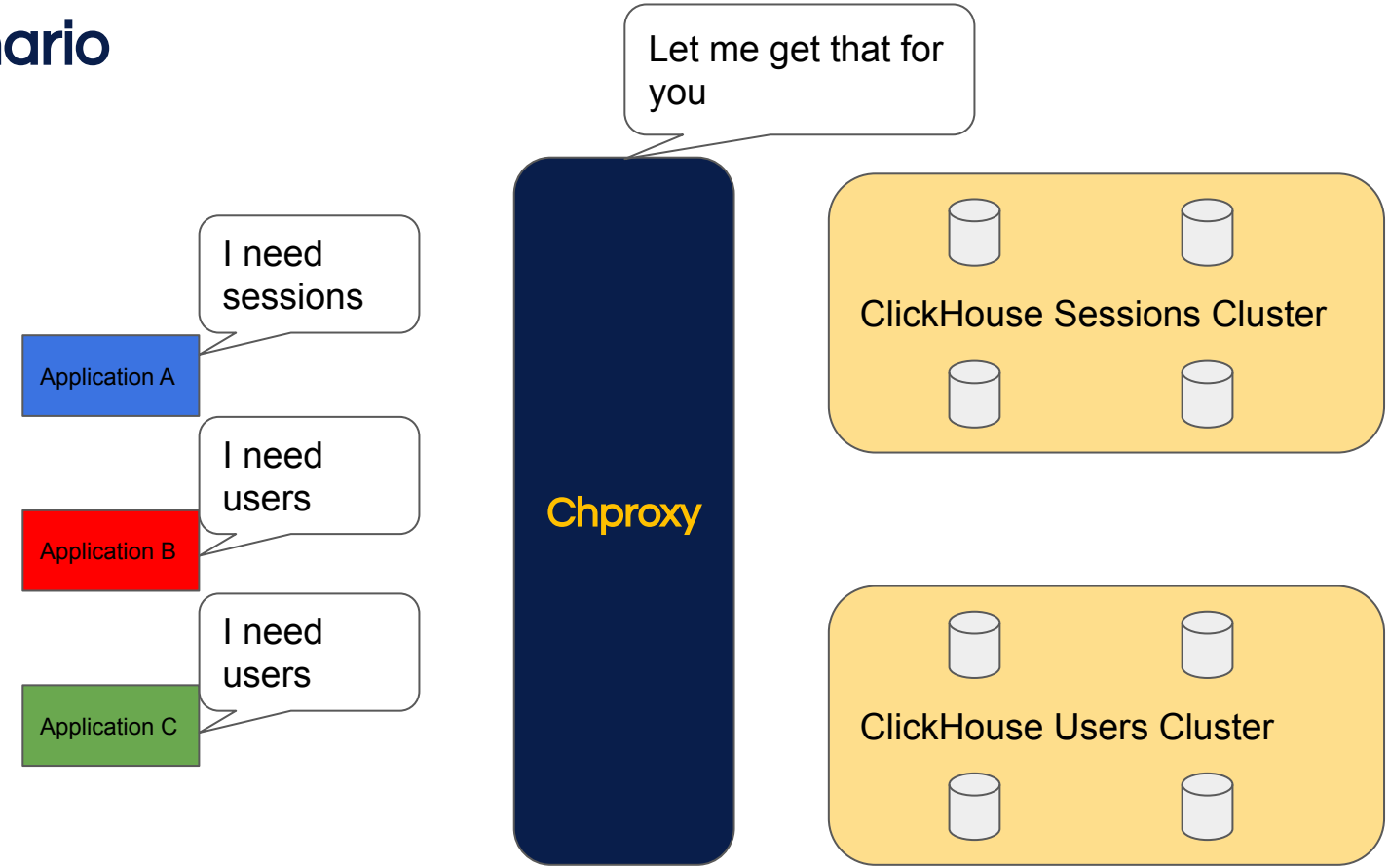
We need a proxy!



Chproxy - Overview

Features we use @Contentsquare

Scenario


Let me get that for you
I need sessions
Application A
I need users
Application B
I need users
Application C
Chproxy
ClickHouse Sessions Cluster
ClickHouse Users Cluster

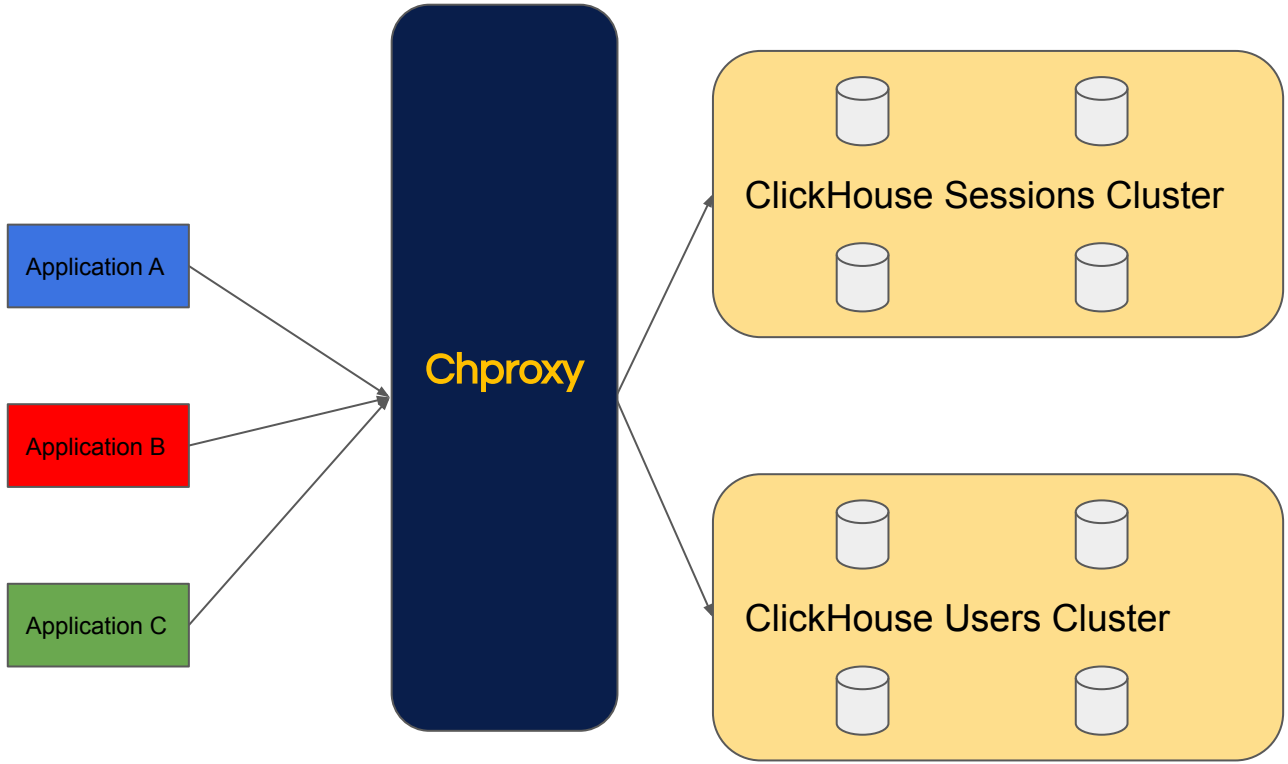
Cluster & User Definition

```yaml
clusters:
  - name: "clickhouse-sessions"
    scheme: "http"
    replicas:
      - name: "replica_1"
        nodes:
          - "clickhouse-sessions-shard-1-replica-1.io:8123"
          - "clickhouse-sessions-shard-2-replica-1.io:8123"
      - name: "replica_2"
        nodes:
          - "clickhouse-sessions-shard-1-replica-2.io:8123"
          - "clickhouse-sessions-shard-2-replica-2.io:8123"
    users:
      - name: "read_only_user"
      - password : "clickhouse_pa$$word"
  - name: "clickhouse-users"
    scheme: "http"
    replicas:
      - name: "replica_1"
        nodes:
          - "clickhouse-users-shard-1-replica-1.io:8123"
          - "clickhouse-users-shard-2-replica-1.io:8123"
      - name: "replica_2"
        nodes:
          - "clickhouse-users-shard-1-replica-2.io:8123"
          - "clickhouse-users-shard-2-replica-2.io:8123"
```

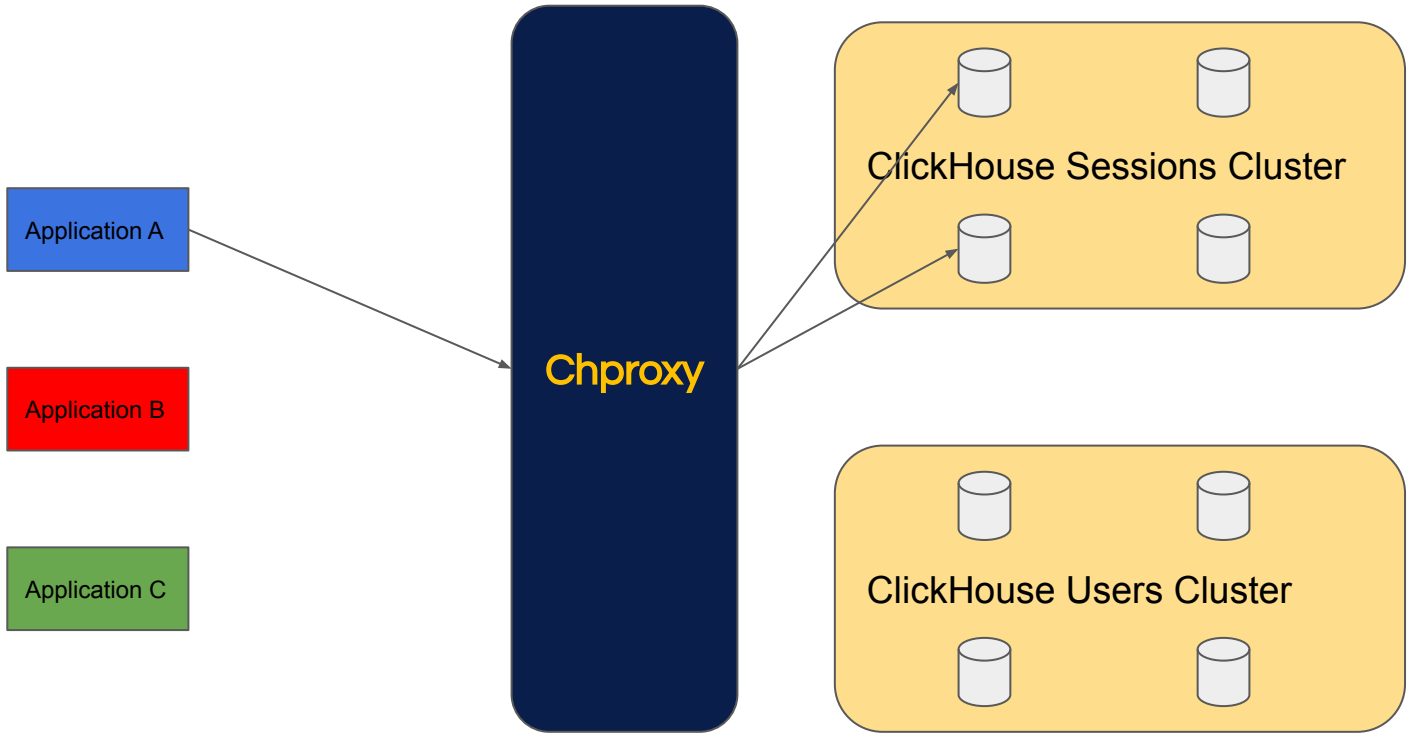
```yaml
users:
    - name: "application_a"
      password: "chproxy_pa$$word"
      to_cluster: "clickhouse-sessions"
      to_user: "read_only_user"
    - name: "application_b"
      password: "chproxy_pa$$word_b"
      to_cluster: "clickhouse-users"
      to_user: "read_only_user"
    - name: "application_c"
      password: "chproxy_pa$$word_c"
      to_cluster: "clickhouse-users"
      to_user: "read_only_user"
```

Routing

```
users:
    - name: "application_a"
      to_cluster: "clickhouse-sessions"
      to_user: "read_only_user"
      password: "chproxy_pa$$word"
      cache: "application_a"
    - name: "application_b"
        to_cluster: "clickhouse-users"
        to_user: "read_only_user"
        password: "chproxy_pa$$word"
        cache: "application_b_and_c"
    - name: "application_c"
        to_cluster: "clickhouse-users"
        to_user: "read_only_user"
        password: "chproxy_pa$$word"
        cache: "application_b_and_c"
```



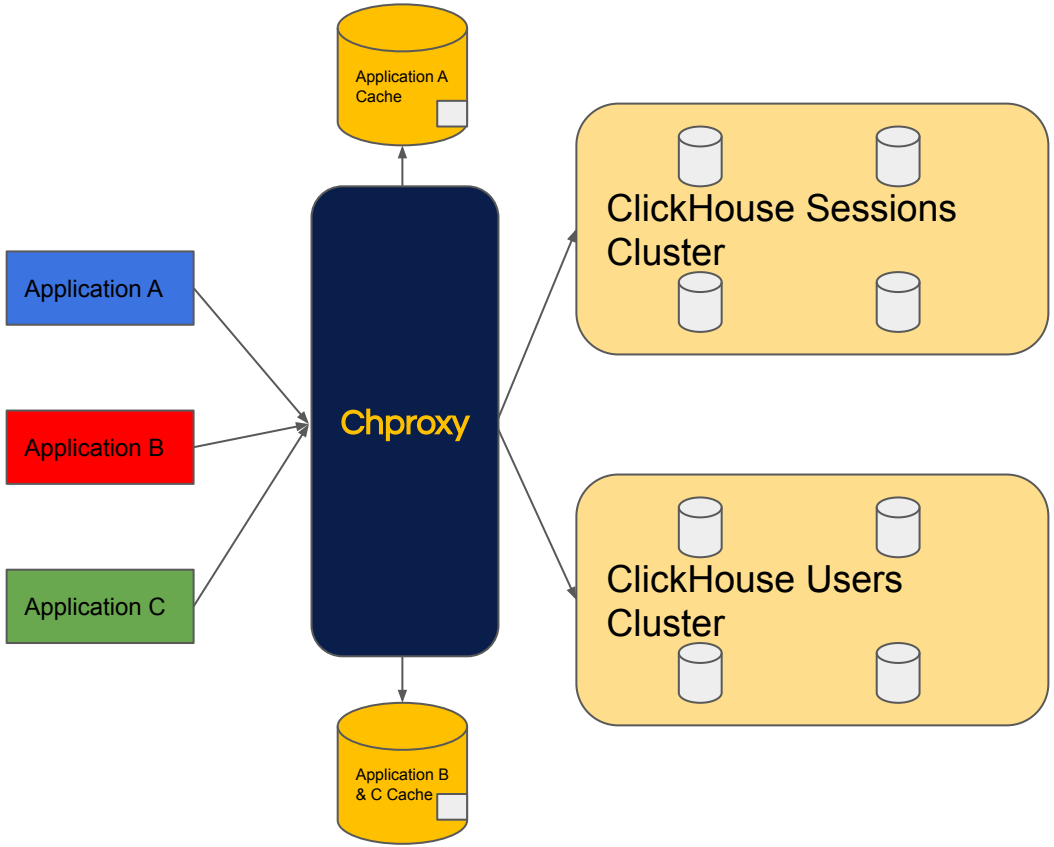
Load Balancing



Caching

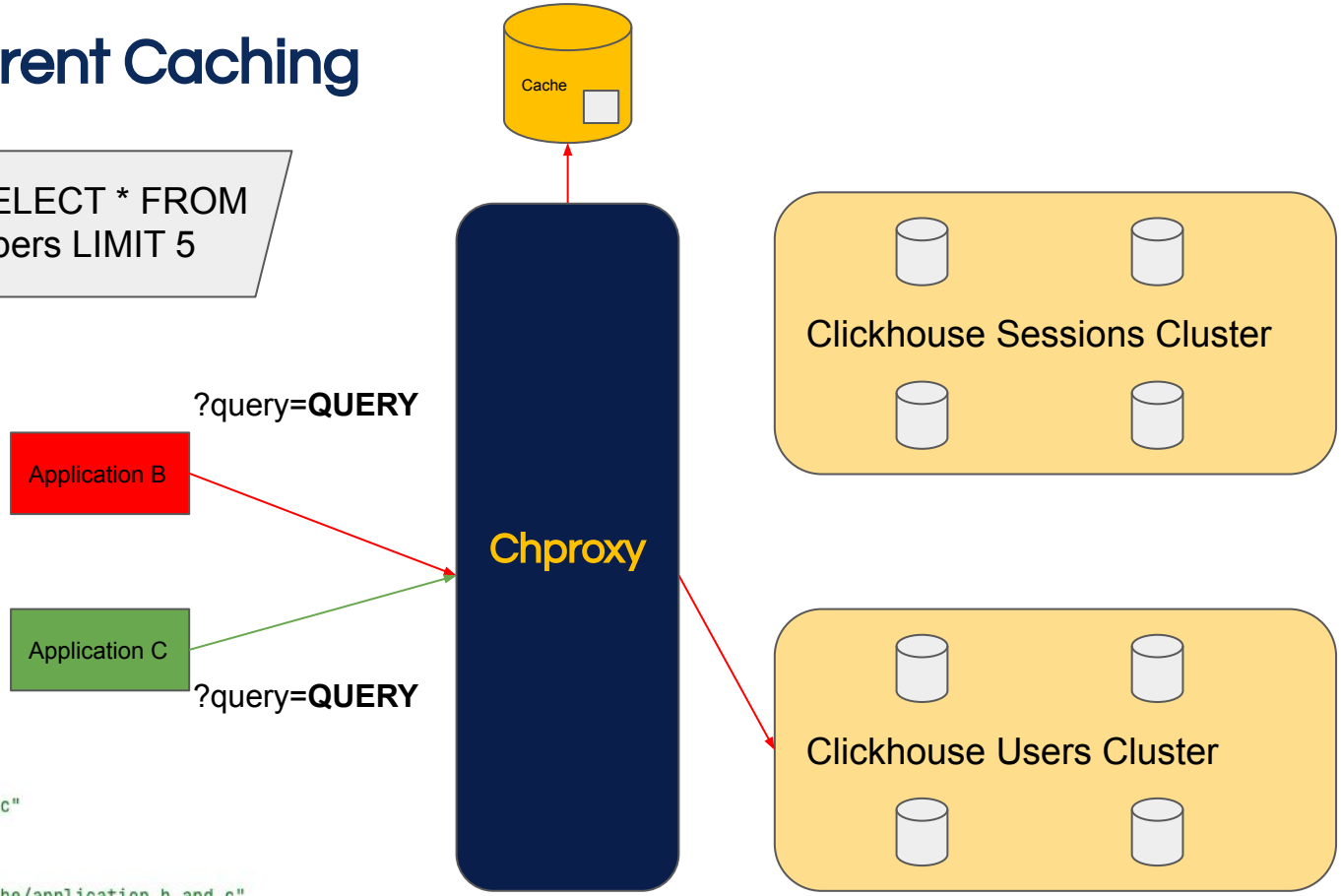
```yaml
users:
    - name: "application_a"
      to_cluster: "clickhouse-sessions"
      to_user: "read_only_user"
      password: "chproxy_pa$$word"
      cache: "application_a"
```

```yaml
caches:
- name: "application_a"
  expire: "1h"
  file_system:
    dir: "/data/disk_00/cache/application_a"
    max_size: "10G"
  grace_time: "30s"
  mode: "file_system"
- name: "application_b_and_c"
  expire: "1h"
  file_system:
    dir: "/data/disk_00/cache/application_b_and_c"
    max_size: "100"
  grace_time: "30s"
  mode: "file_system"
```



Concurrent Caching

QUERY = SELECT * FROM system.numbers LIMIT 5




```yaml
caches:
- name: "application_b_and_c"
  expire: "1h"
  file_system:
    dir: "/data/disk_00/cache/application_b_and_c"
    max_size: "100"
  grace_time: "30s"
  mode: "file_system"
```

Miscellaneous Features

- Query stripping : https://host?query=...&max_execution_time=1000 → https://host?query=...
- Allowed Networks : whitelist ip ranges (10.0.0.0/8)
- Concurrent queries : user setting **max_concurrent_queries**
- Max execution time per user : user setting **max_execution_time**

Glimpse of history

Timeline

2017

Vertamedia creates chproxy

Step 1

2019

Contentsquare starts using Clickhouse & chproxy

Step 2

2020

Last commit on the repository, support from Vertamedia dropped

Step 3

2022

Contentsquare takes ownership of the component

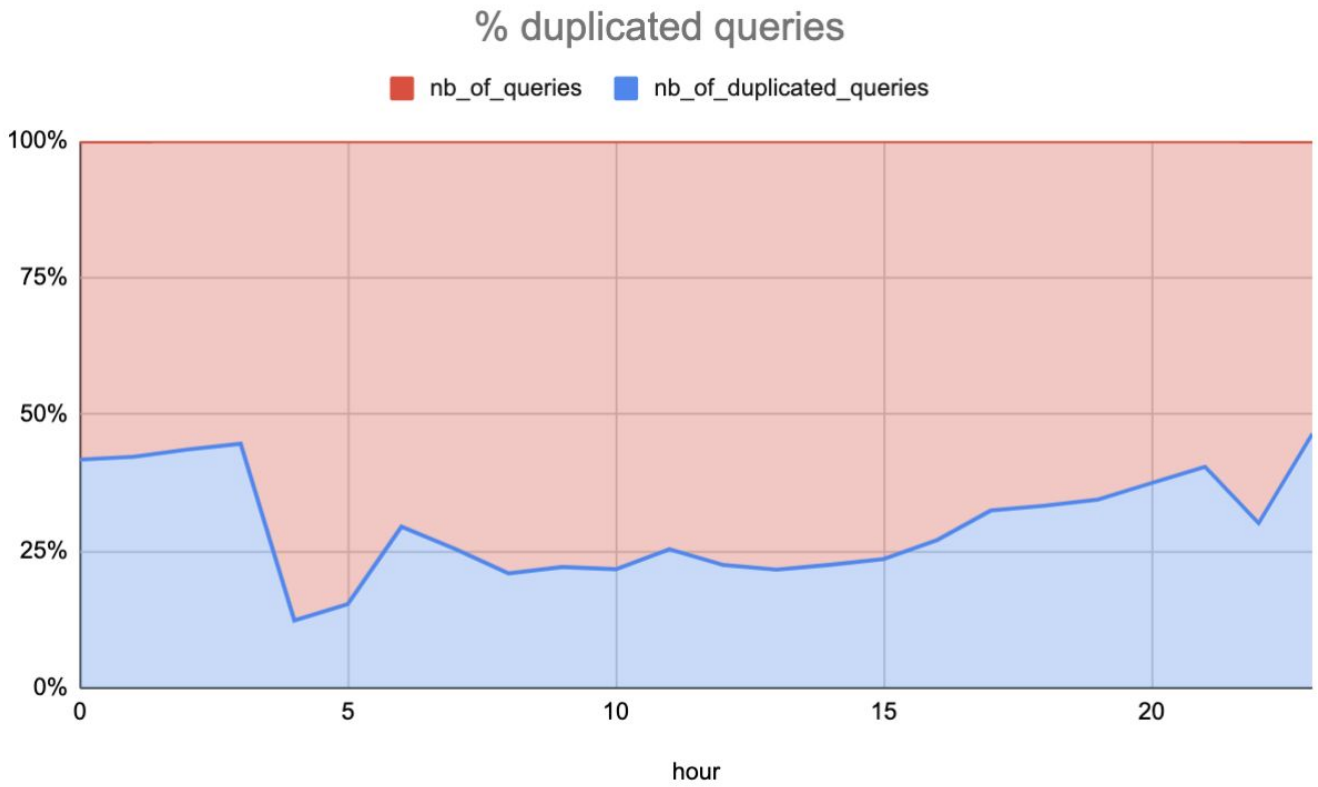
Step 4

2024

Chproxy makes an IPO

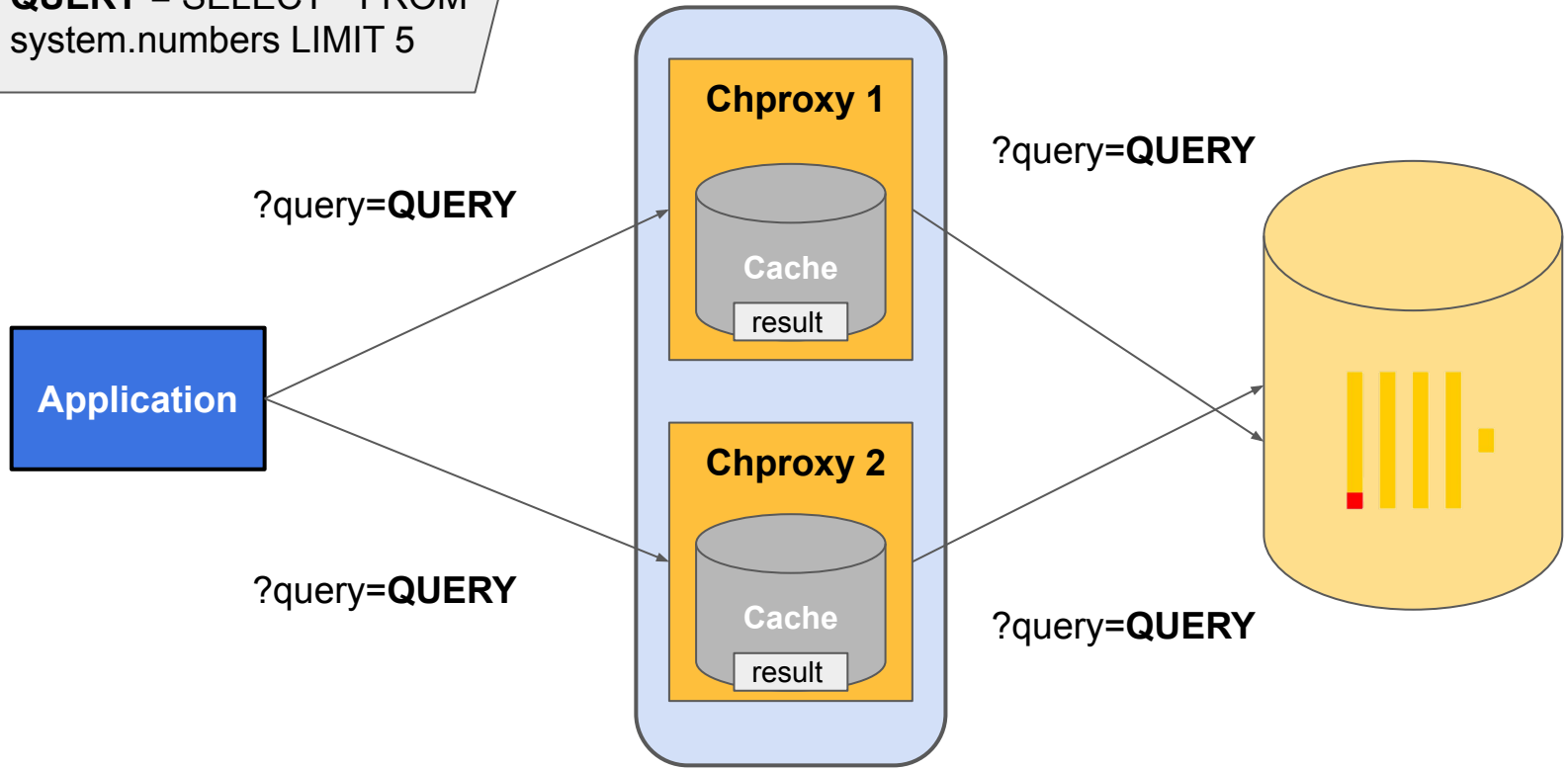
Step 5

In 2021 we realised something...



Problem illustration

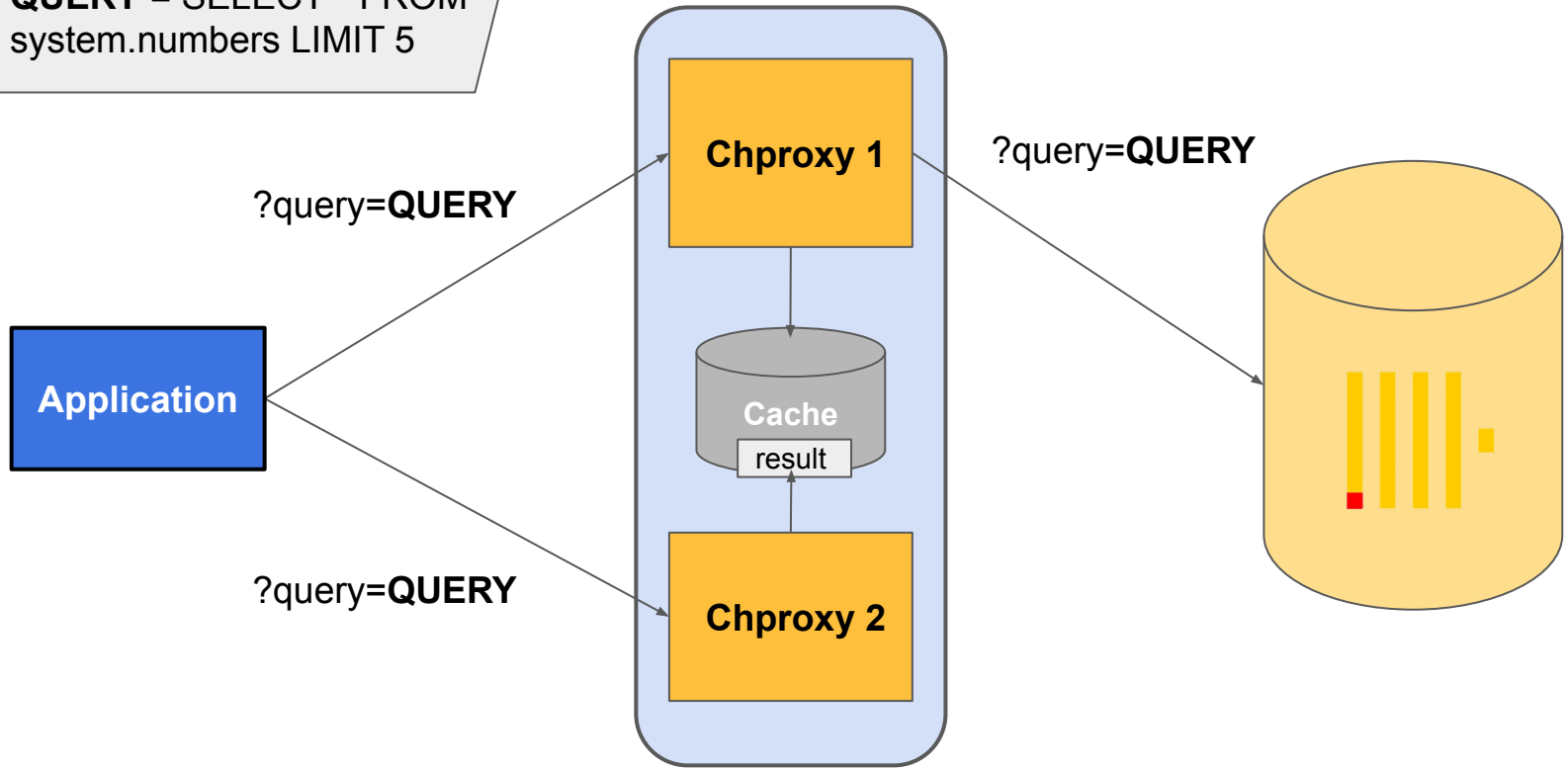
QUERY = SELECT * FROM system.numbers LIMIT 5



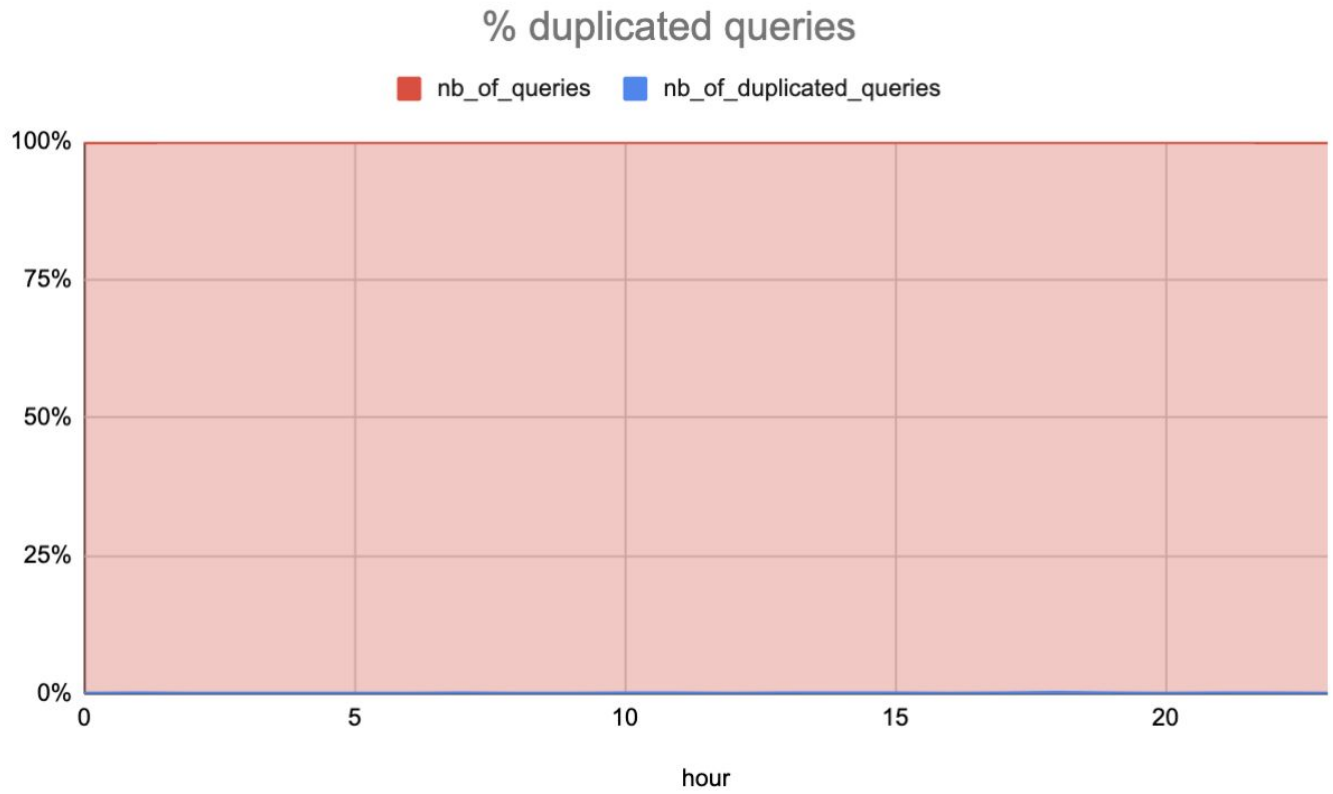
New feature! Distributed cache

Distributed cache

QUERY = SELECT * FROM system.numbers LIMIT 5



Distributed cache impact @Contentsquare



Distributed cache implications

Filesystem cache

```
caches:
  - name: "longterm"
    mode: "file_system"
    file_system:
      dir: "cache_dir"
      max_size: "10B"
    expire: "1m"
```

Distributed cache - redis

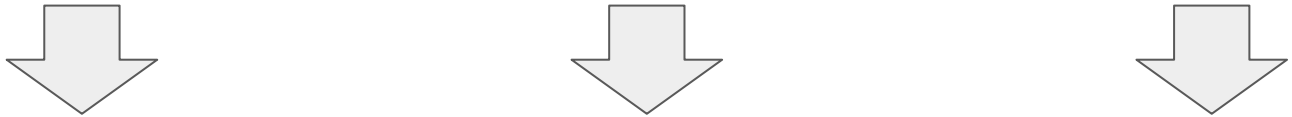
```
caches:
  - name: "redis-cache"
    mode: "redis"
    redis:
      addresses:
        - "localhost:6379"
      max_size: "10M"
    expire: "1m"
```

Distributed cache design

Filesystem Cache

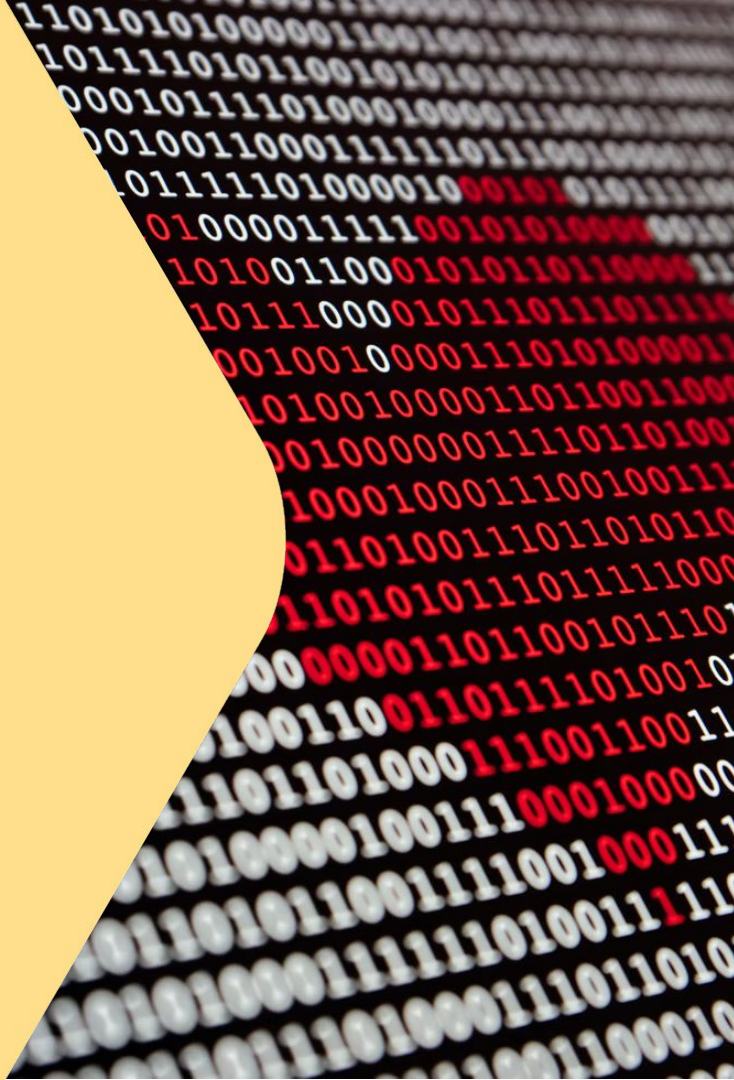
Redis Cache

Your cache ?

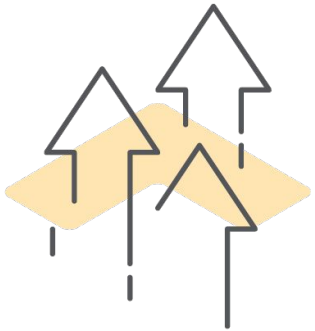


```
// Cache stores results of executed queries identified by Key
type Cache interface {
    io.Closer
    Stats() Stats
    Get(key *Key) (*CachedData, error)
    Put(r io.Reader, ctMetadata ContentMetadata, key *Key) (time.Duration, error)
    Name() string
}
```

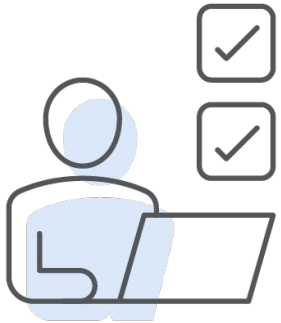
What’s next ?



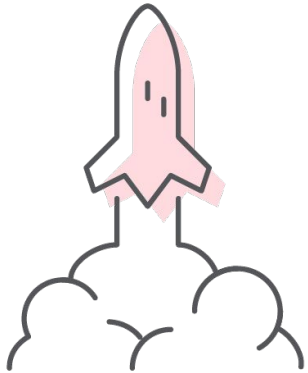
Roadmap for 2022



Reliability



Documentation



Innovation

https://www.chproxy.org/

CONTENTSQUARE