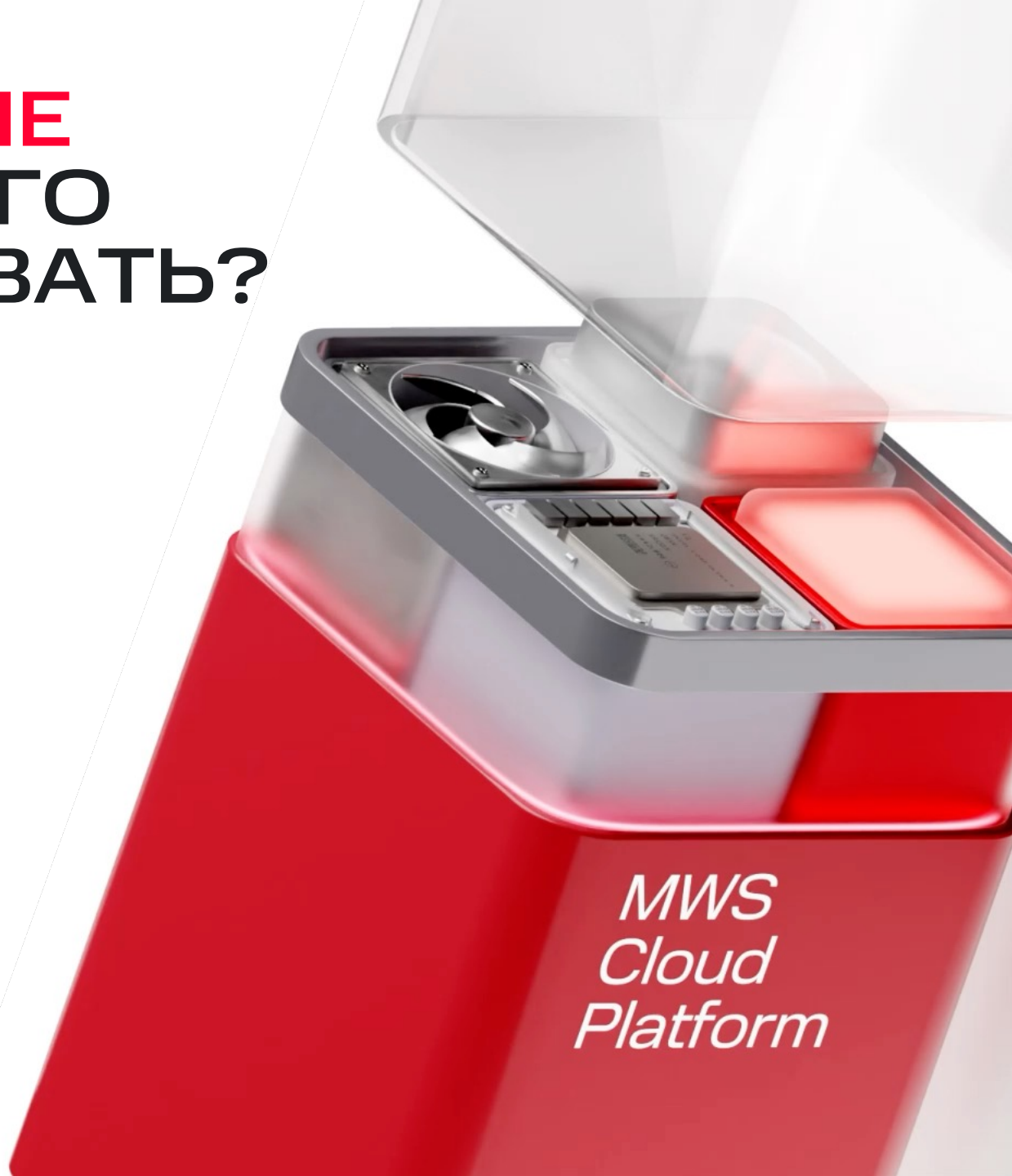
БЫСТРЫЕ ОБЪЕКТНЫЕ ХРАНИЛИЩА: ДЛЯ ЧЕГО И КАК ИХ ИСПОЛЬЗОВАТЬ?

Кирилл Беспалов

Технический руководитель
MWS Cloud Platform



MWS Cloud Platform Object Storage — это:

Erasure Encoding & Replication

Надёжность

?

Производительность

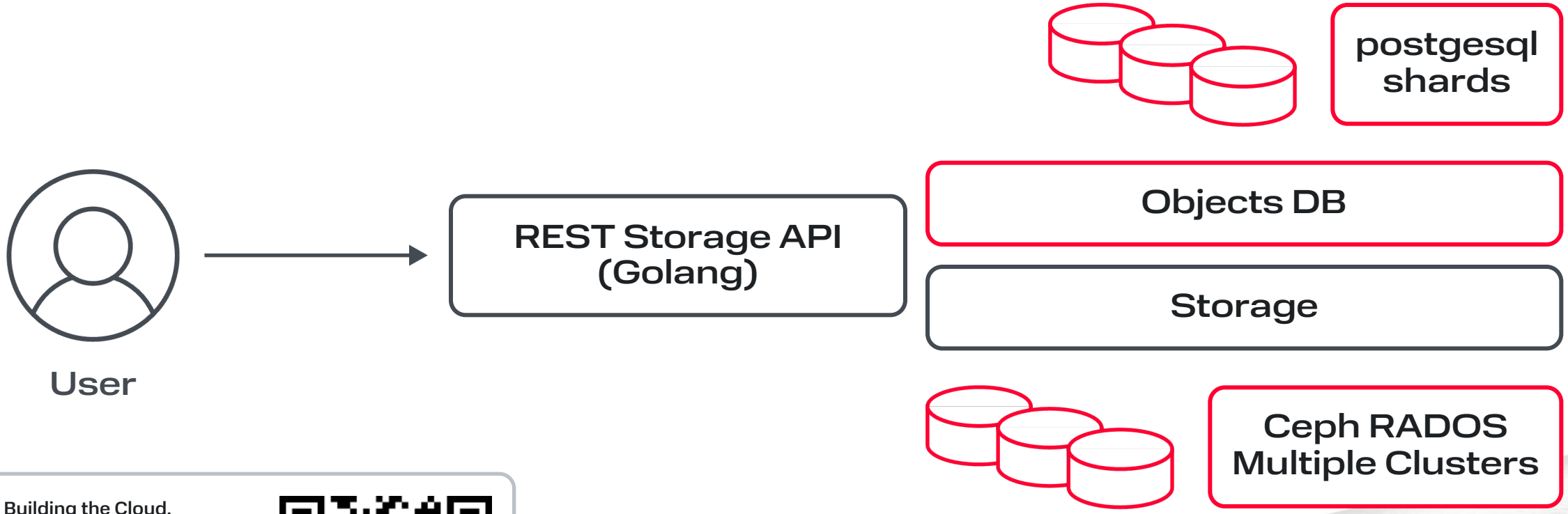
KMS (шифрование)

Безопасность

SLA 99,95%

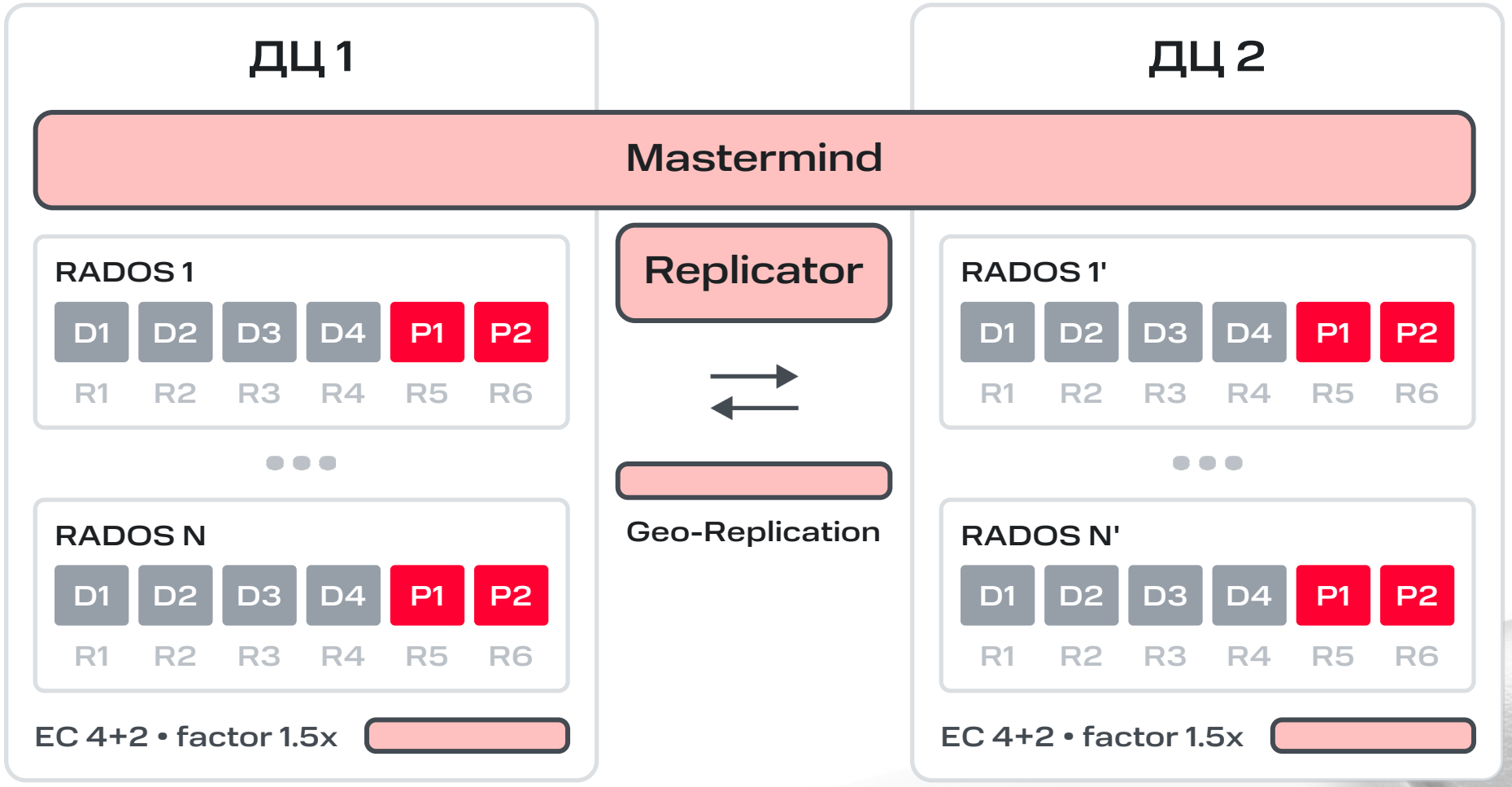
Доступность

Общая архитектура

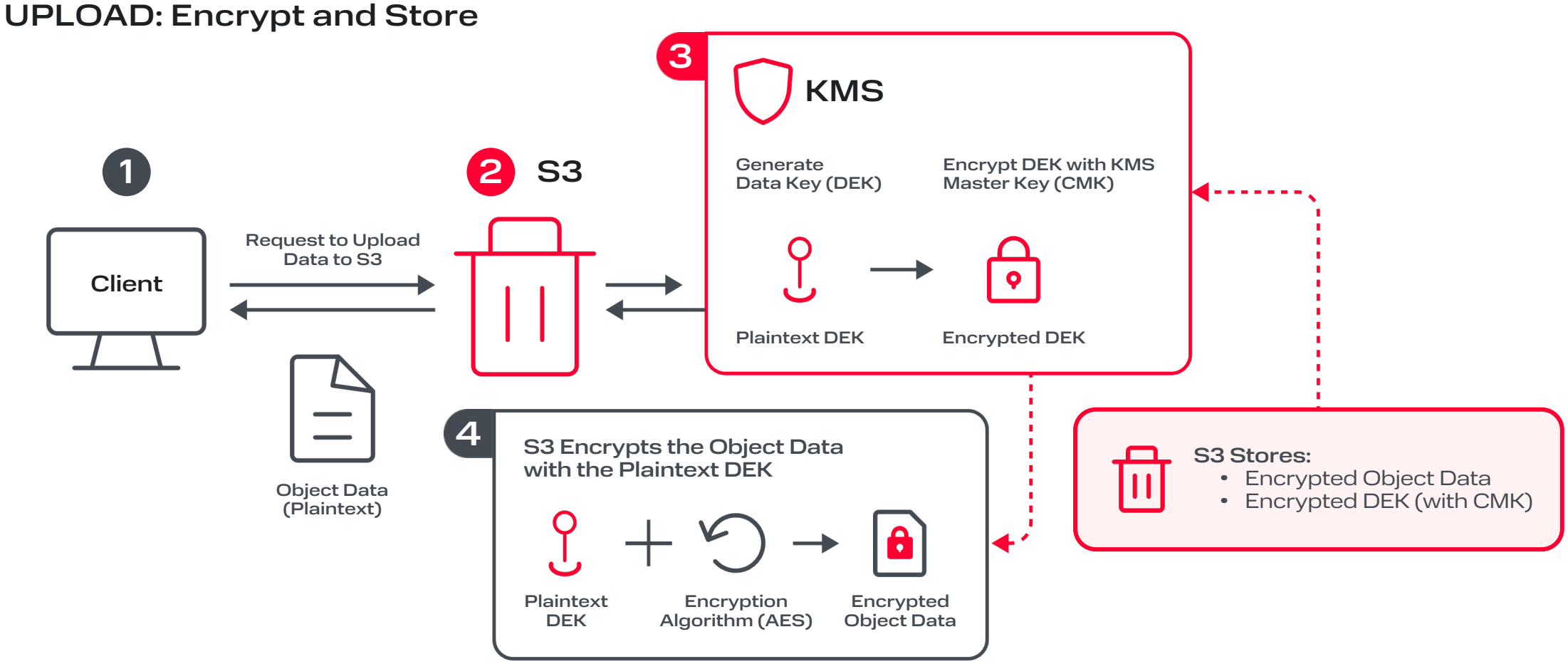


Building the Cloud.
Episode 3: Object Storage

Отказоустойчивость



Безопасность



Производительность

Standard — основной класс хранения

Отлично подходит для:

- Бэкапов
- Резервных копий
- Артефактов CI/CD
- Статического контента и так далее

Standard Class

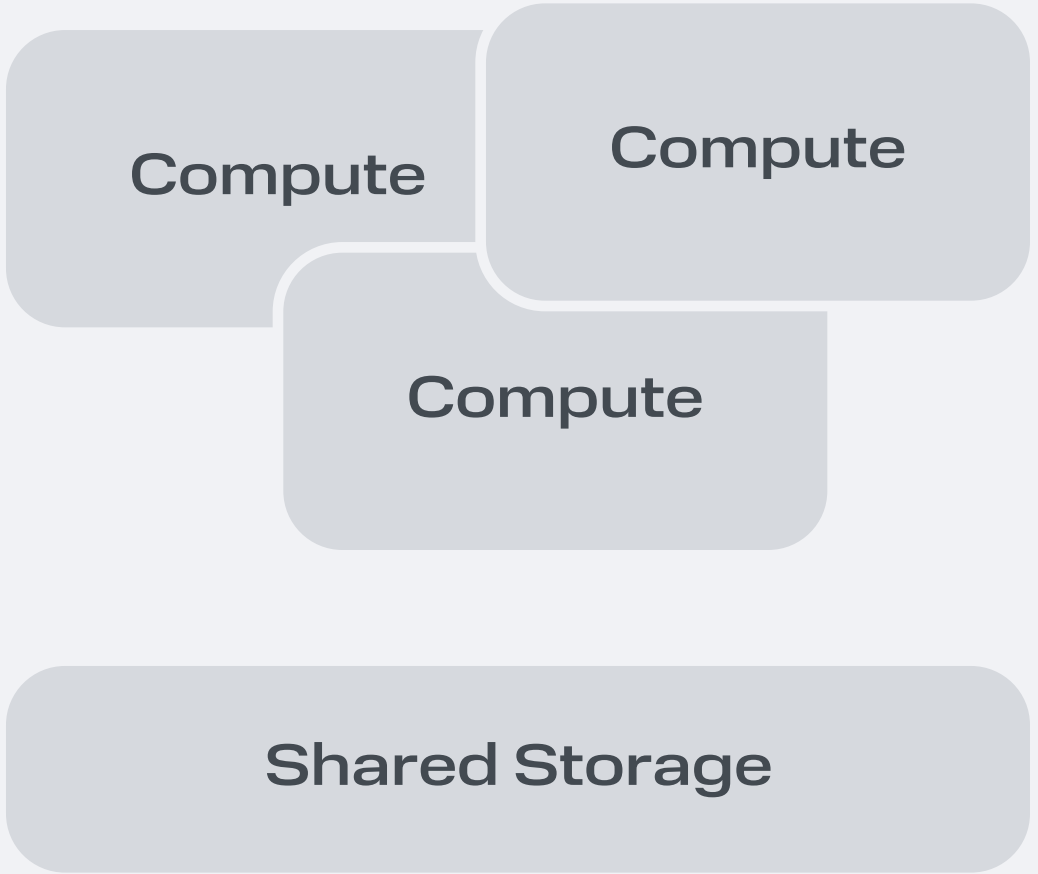
Cold

Ice

Требования к latency и throughput растут

Новые типы нагрузок

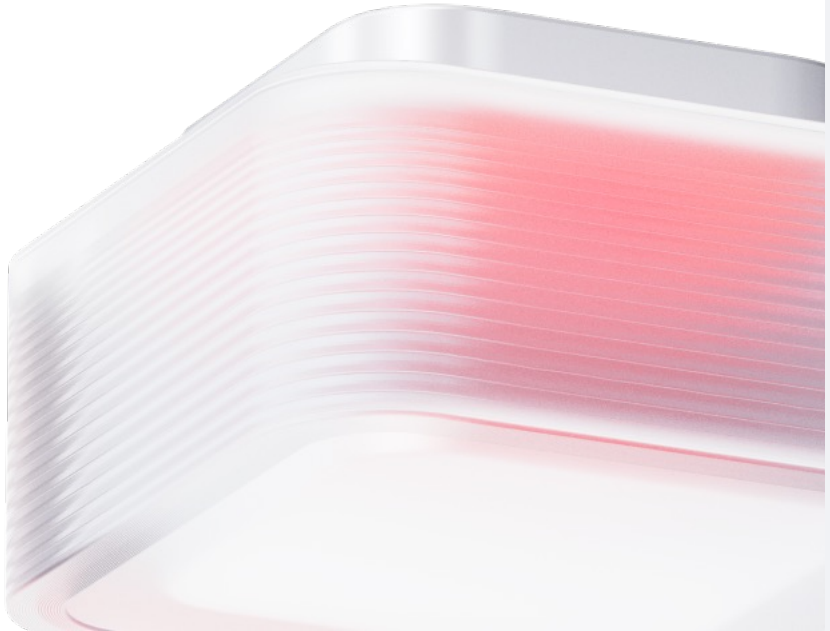
- Lakehouse & Serverless BI
- ML & Model Training
- Observability



MWS Cloud Platform — классы хранения

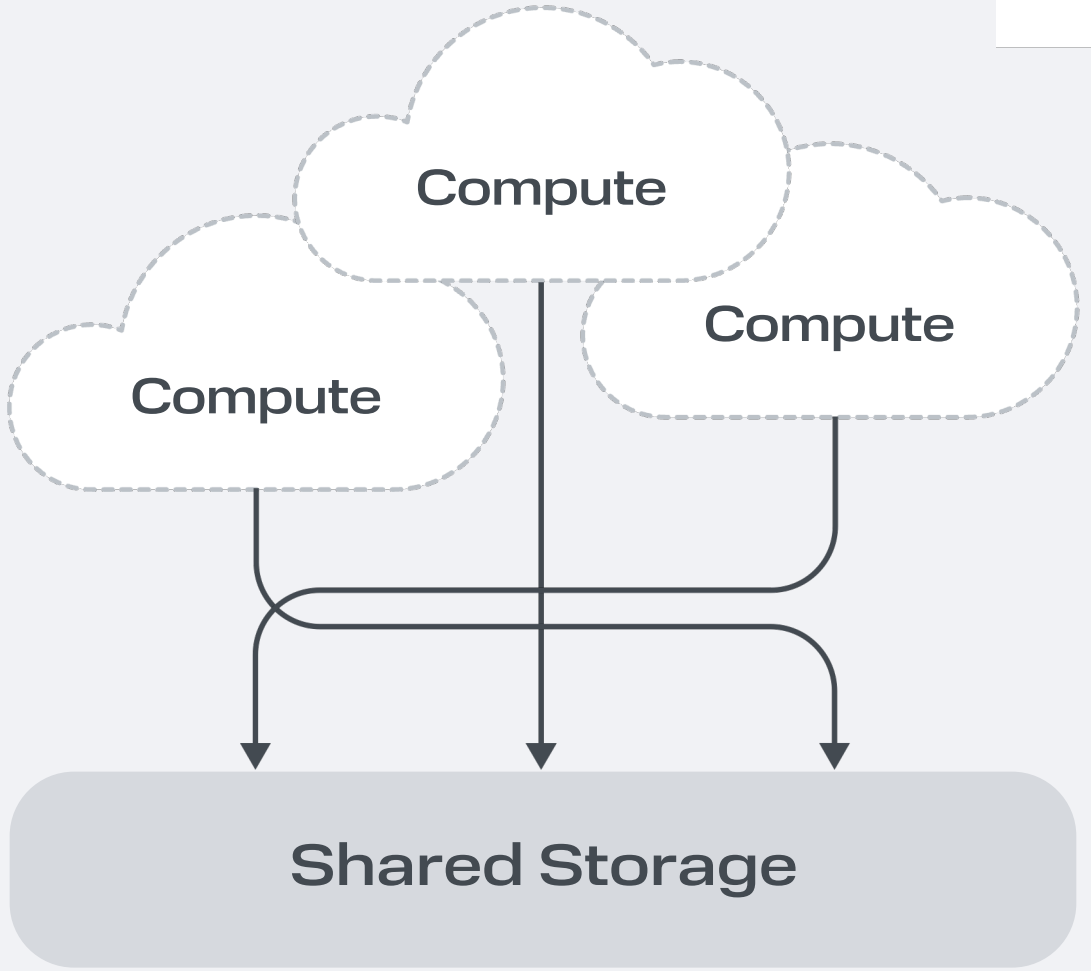
Новые типы нагрузок

- Lakehouse & Serverless BI
- ML & Model Training
- Observability



Что их объединяет?

Концепция
Compute & Storage
Separation



Базовый сценарий. Что измеряем?

cmd: warp run scenario.yaml

PUT OBJECT 64 MB — MEDIAN THROUGHPUT

Warm storage class

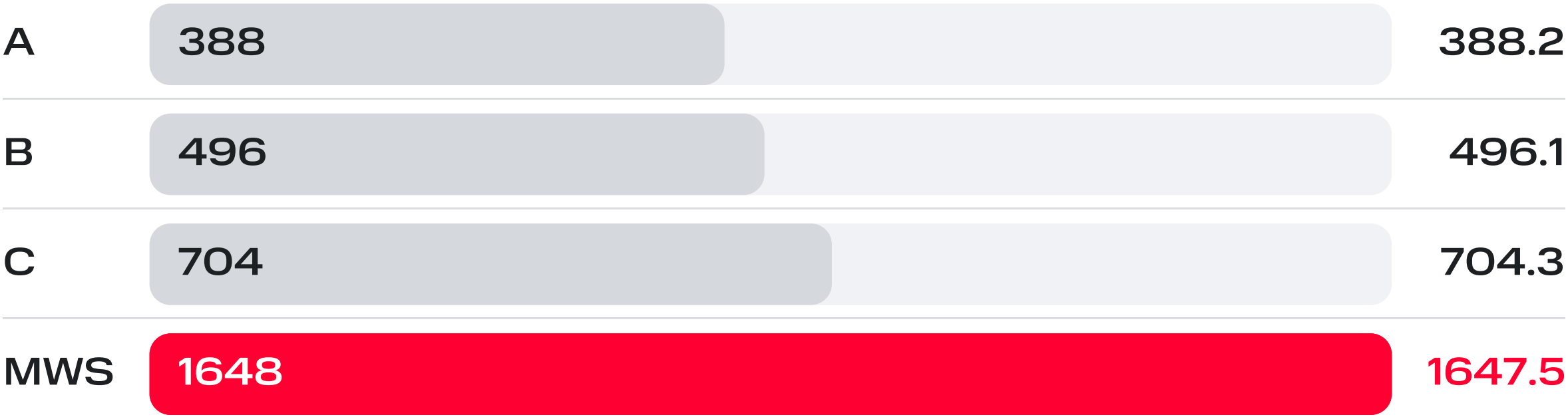
MiB/s

A	286	285.8
B	732	732.3
C	795	794.6
MWS	1799	1799.3

GET OBJECT 64 MB — MEDIAN THROUGHPUT

Warm storage class

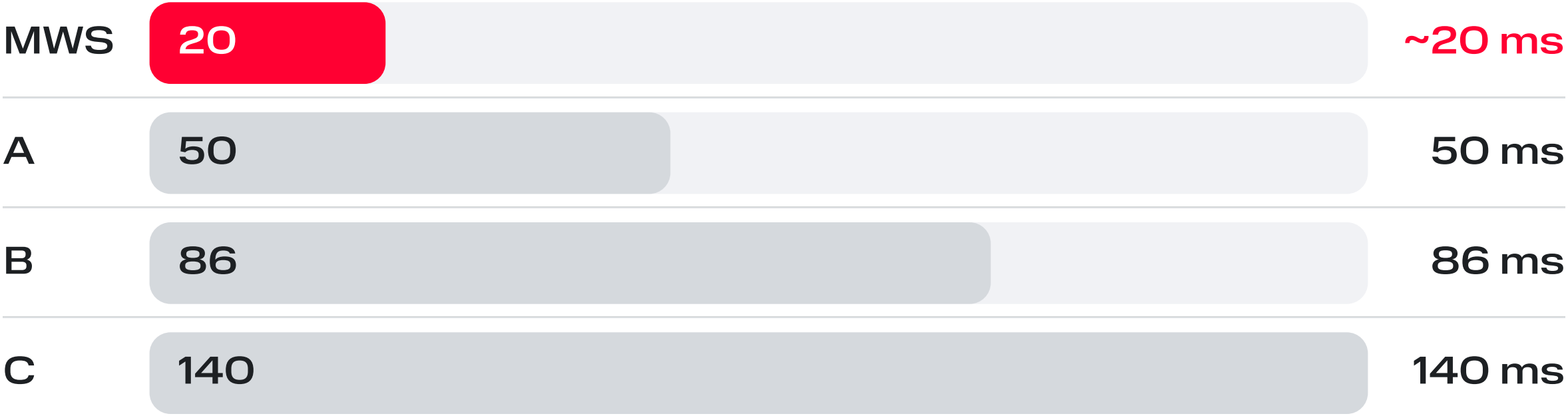
MiB/s



TIME TO FIRST BYTE — P90

Warm storage class

ms (меньше — лучше)



Почему так?

Высокопроизводительный SDN

Up to 25 Gbps per link

DPDK

VPP

Производительность meta и storage слой Object Storage

storage

storage

nvme

pg

Классический Lakehouse

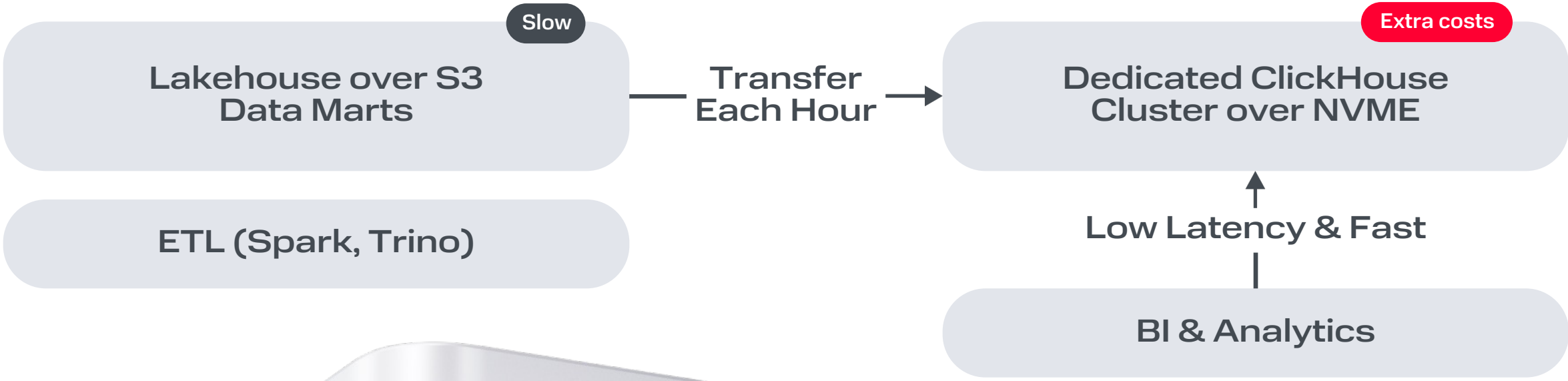
Raw (Bronze) → ODS / DDS (Silver) → Data Marts (Gold)

Shared Object Storage

ETL over Airflow + Spark / Trino

Что это значит на практике?

Классический Lakehouse для интерактивной аналитики требует выделенного кластера, например с ClickHouse для работы аналитиков и BI



Что это значит на практике?

Пример на практике:
выполнение набора аналитических запросов

Пример на практике:

ClickHouse over Network NVME Disk

(Native Merge Tree)

nvme disk,
нативный формат хранения

Diskless ClickHouse over Object Storage (S3 table + raw parquete)

табличная функция s3 table
и «сырой» parquete

Пример на практике:

выполнение набора аналитических запросов

Сопоставимо по времени выполнения для аналитических запросов

В худших случаях overhead порядка ~1,5x

#	ЗАПРОС	NVME, С	OBJSTOR, С	РАЗНИЦА
1	GROUP BY UserID × SearchPhrase	1.818	1.656	×0.9
2	High-cardinality GROUP BY UserID	2.001	1.918	×1.0
3	Search + mobile / desktop split	1.009	1.246	×1.2
4	Top search phrases	0.577	0.733	×1.3
5	count(DISTINCT SearchPhrase)	0.567	0.779	×1.4
6	GROUP BY + uniqExact	0.583	0.842	×1.4
7	count(DISTINCT UserID)	0.404	0.636	×1.6
8	Subquery + session buckets	0.459	0.764	×1.7
9	LIKE '%google%' + domain()	0.938	2.117	×2.3
10	Hourly analytics за период	0.412	0.936	×2.3
11	Funnel + HAVING	0.318	0.868	×2.7
12	Daily stats по top-10 счётчикам	0.288	0.945	×3.3
13	Mobile phone models	0.185	0.765	×4.1

Другой пример: ML

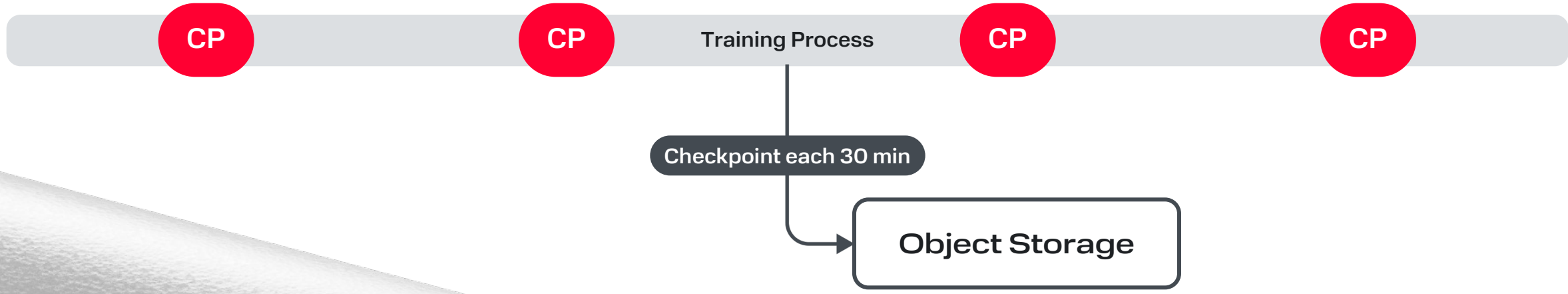
Проблема: GPU простаивают во время checkpoint

GPU

GPU

GPU

GPU



Под капотом: инфраструктура