Yandex

Яндекс Облако

Контроль доступа к данным в облаке при помощи Key Management Service

Андрей Иванов

Руководитель направления развития сервисов безопасности

Алексей Захаров

Руководитель подразделения YC KMS

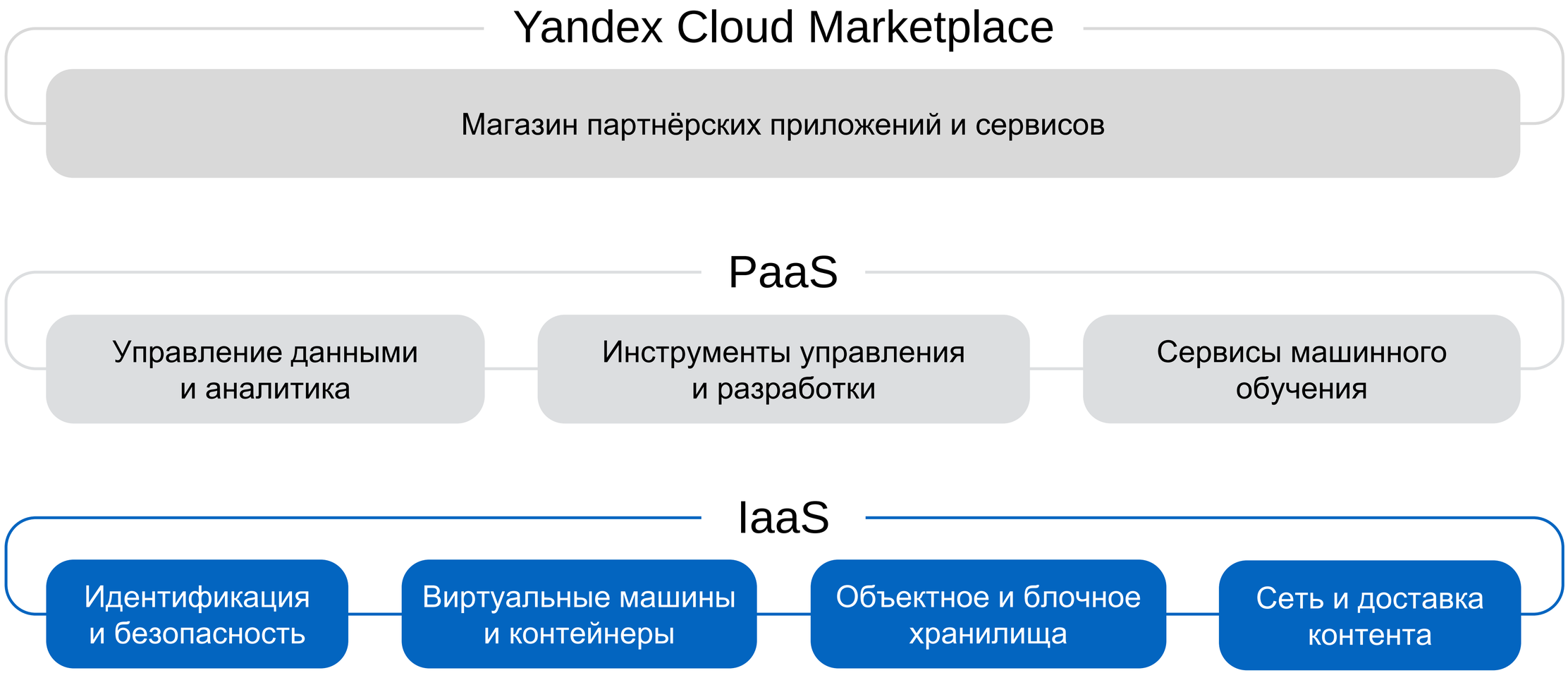
ayza@yandex-team.ru

@ayza11

Программа

- Яндекс.Облако
- Проблемы шифрования в облаке
- Общие принципы KMS
- Архитектура
  - Подходы к реализации потокового шифрования
- Сценарии использования

Яндекс.Облако — это платформа



Разделение ответственности

Клиент Яндекс.Облако

	Собственная инфраструктура	IaaS	PaaS / SaaS
Управление доступом к данным	✓	✓	✓
Безопасность ОС и приложений	✓	✓	●
Сетевая безопасность (Overlay)	✓	✓	●
Резервное копирование	✓	✓	●
Шифрование	✓	✓	●
Логи аудита	✓	●	●
Безопасность хранилища данных и оборудования	✓	●	●
Сетевая безопасность (Underlay)	✓	●	●
Физическая безопасность и катастрофоустойчивость (DR)	✓	●	●

Проблемы шифрования в Облаке

Шифрование в Облаке с точки зрения пользователя

- Где хранятся ключи?
- Какой у ключей жизненный цикл?
- Непонятно, на каком этапе шифруются данные
- Всё тормозит
- Мои данные всё равно утекут

Шифрование данных с точки зрения разработчика Я.Облака

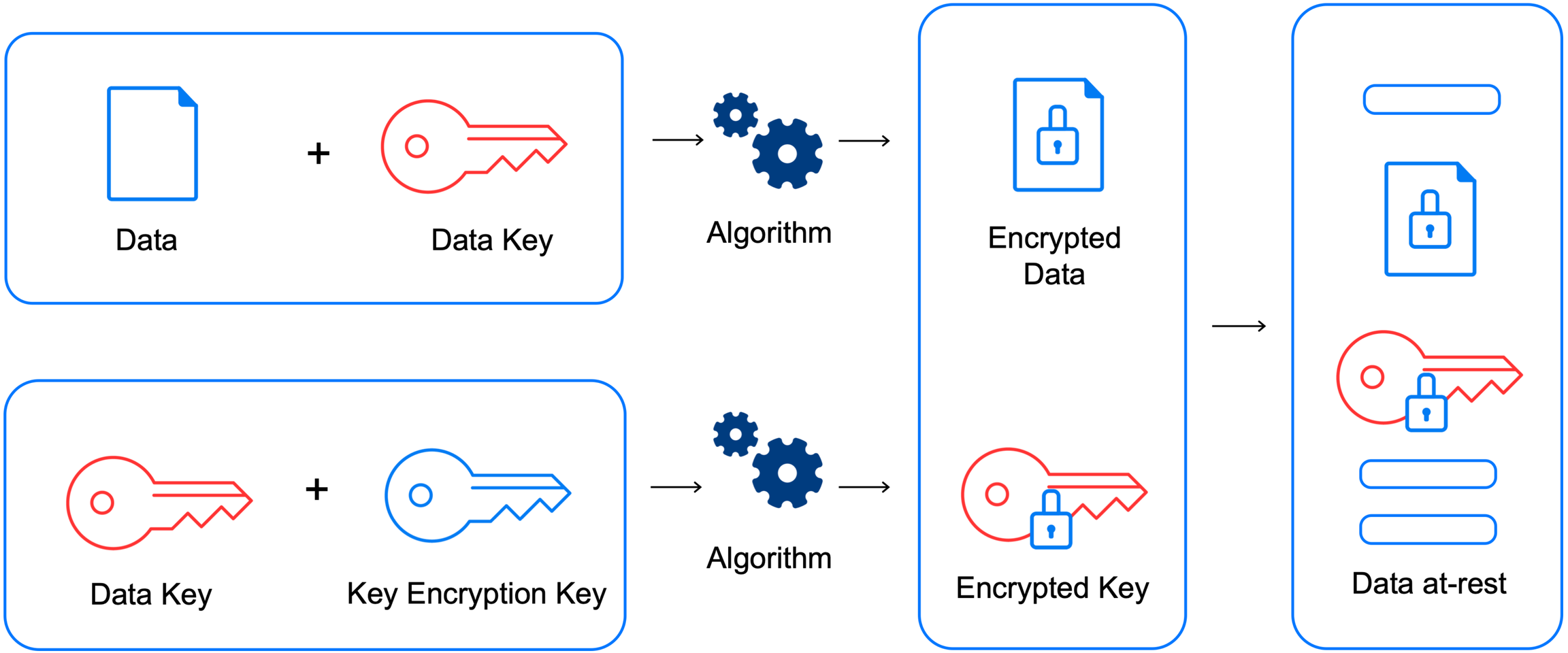
- Много сервисов: необходимость единого центра для управления ключами
- Огромный объём данных
- Хранение ключей в зашифрованном виде

Общие принципы KMS

Принципы KMS

- Защита данных → защита ключей
- Отдельный сервис по управлению ключами
- Повышенные требования к безопасности
- Распределённая схема шифрования данных

Envelope encryption

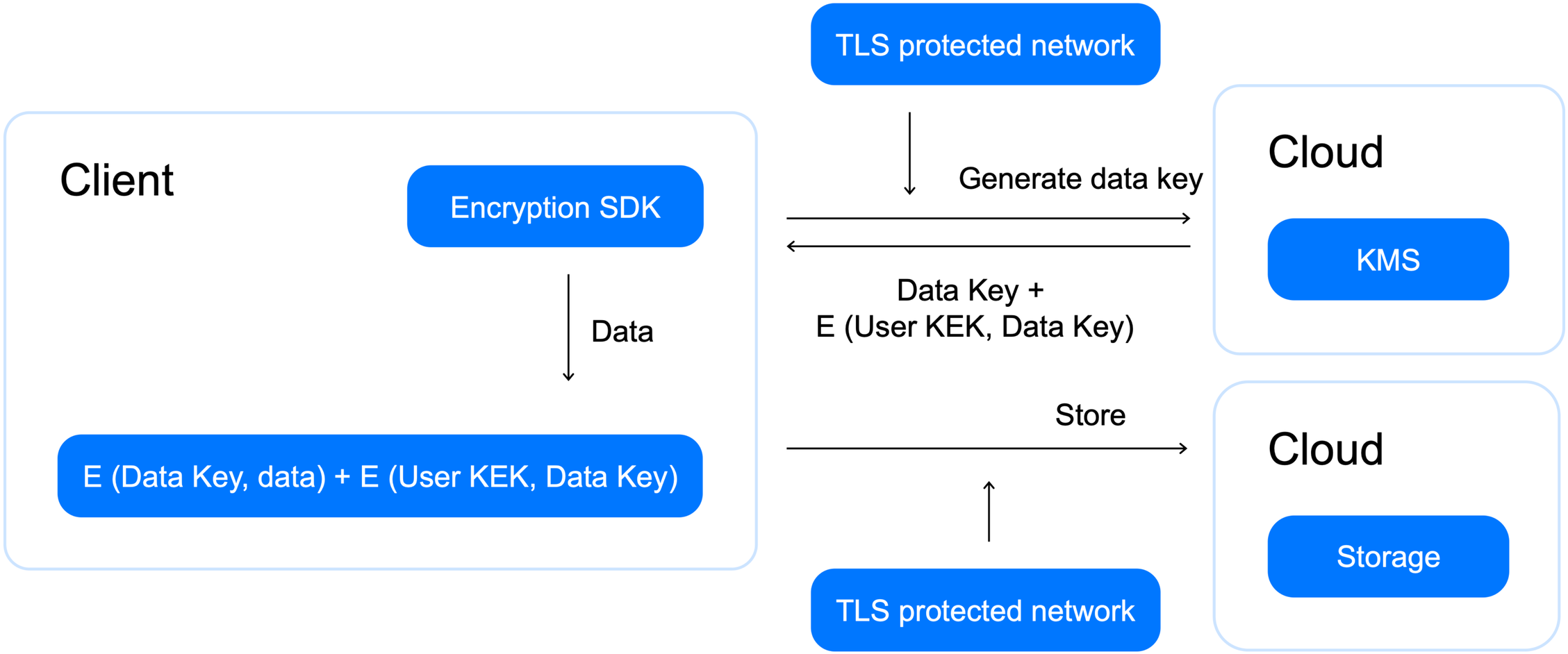


Security Promise

KEK не покидают KMS в открытом виде

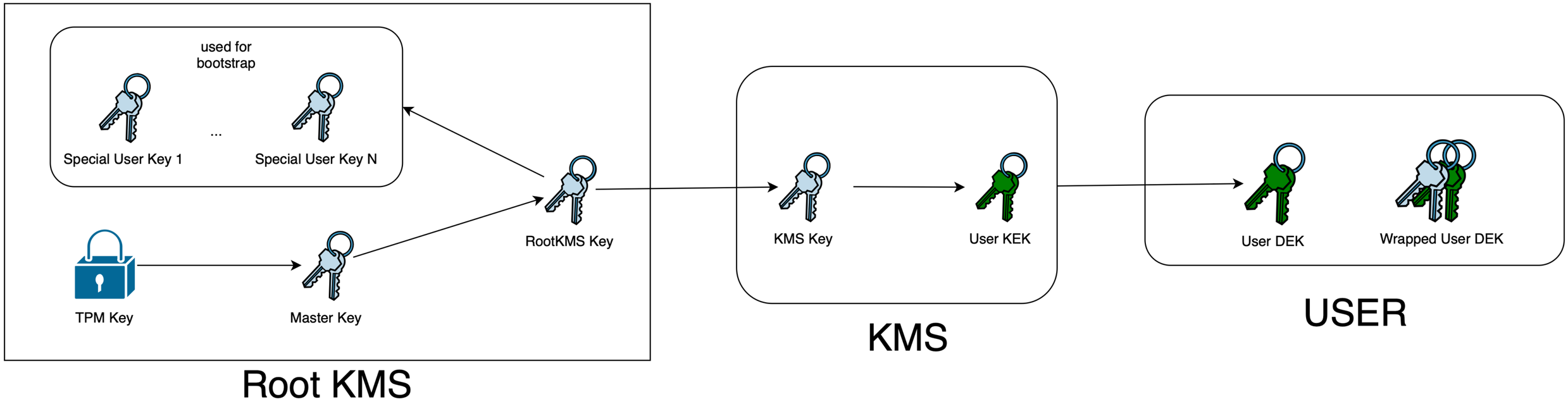
- KEK не покидают периметр Яндекс.Облака ни в каком виде
- KEK - только в памяти KMS
- DEK для Envelope Encryption - только по TLS, уничтожаются после передачи пользователю

KMS: схема работы



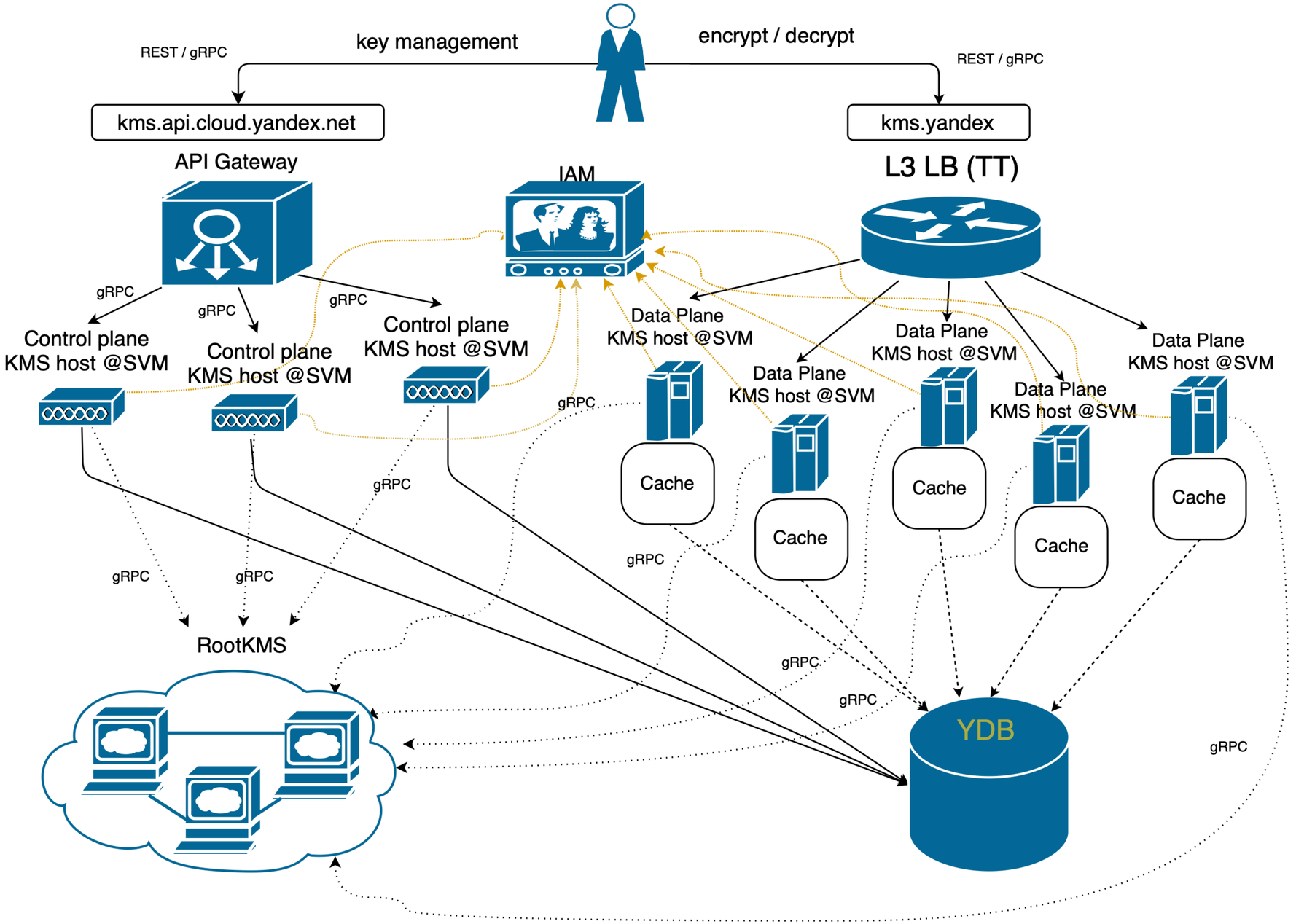
Архитектура

Chain of trust



Root KMS

- Отдельные физические машины
- Статические конфиги ключей
- Не зависит от базы данных
- Только encrypt / decrypt API для пользователей
- MasterKey шифруется ключом TPM



Basic Architecture

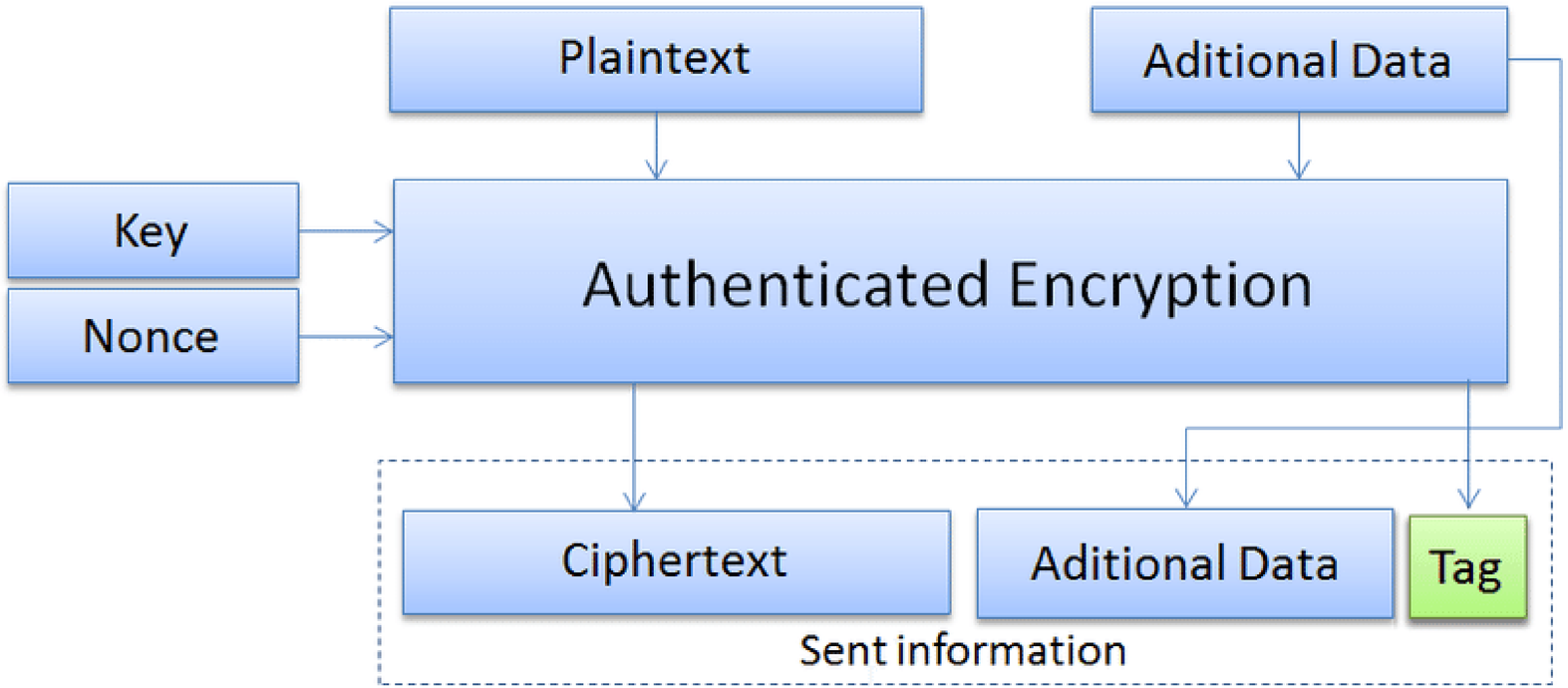
Подходы к реализации потокового шифрования

Потоковое шифрование

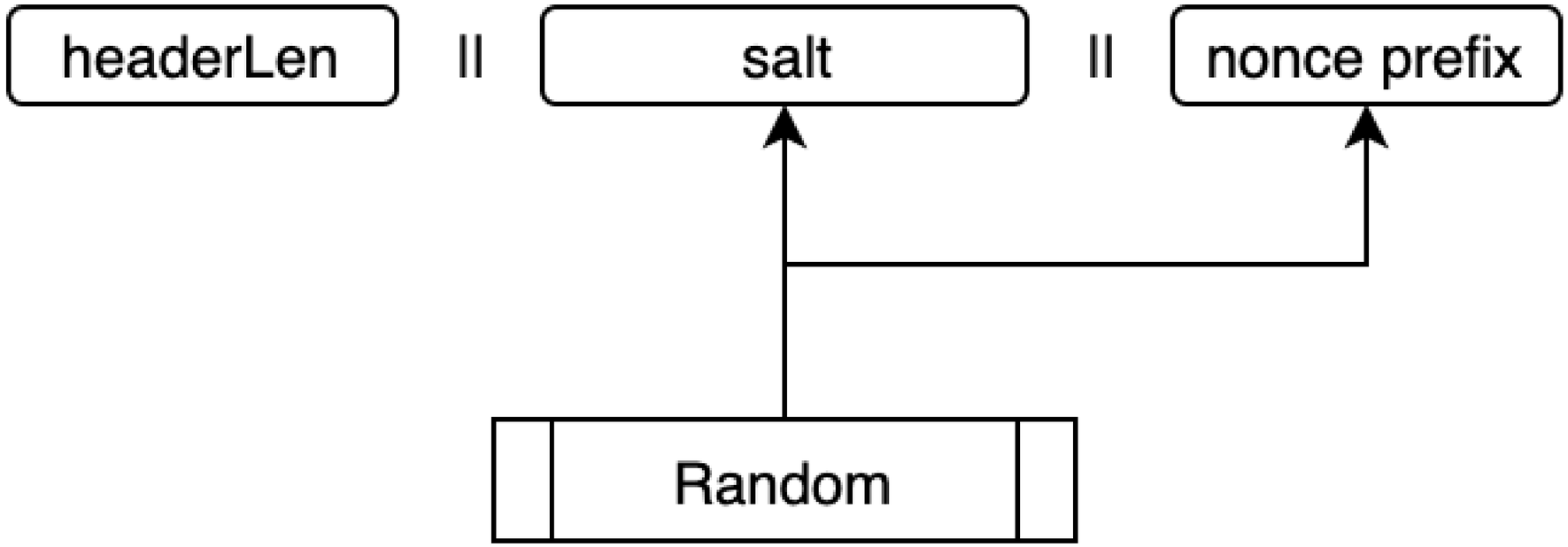
- Шифрование больших файлов
- Возможность range-запросов на чтение
- Разбиение исходного потока данных на сегменты
- AES-GCM-HKDF-STREAMING
- Реализовано в связке KMS + Object storage

Шифрование

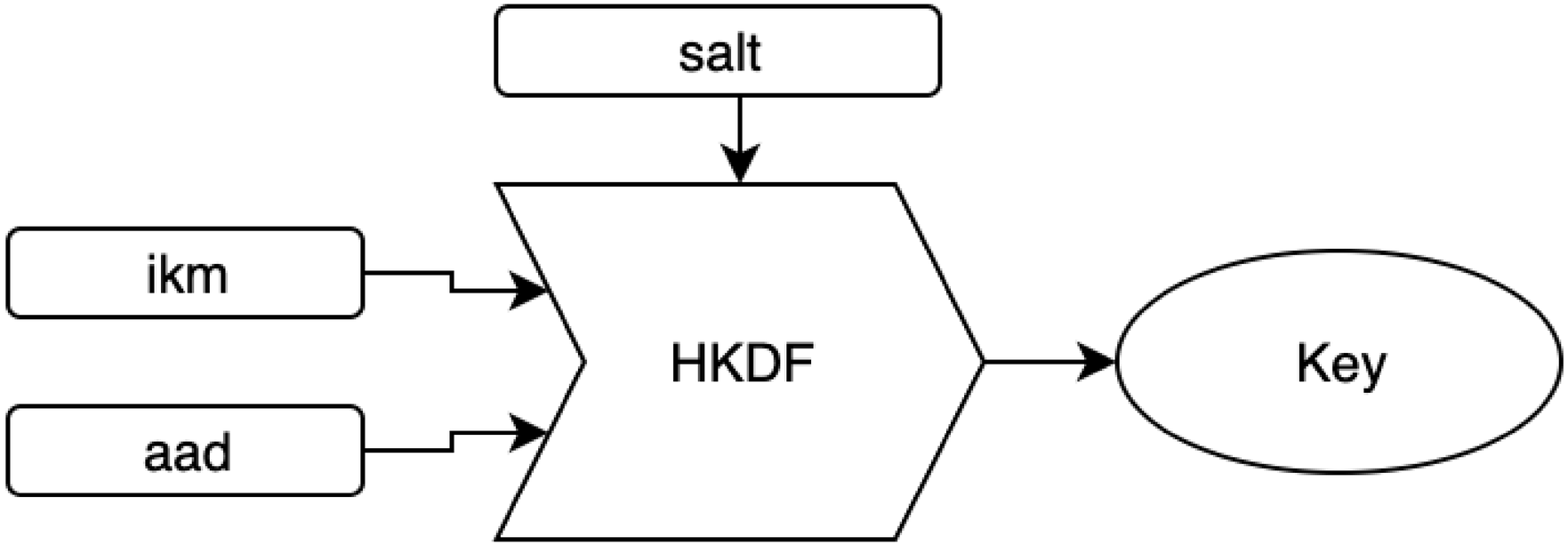
- Authenticated Encryption with Additional Data: **AES-GCM**
- Совмещает в себе и шифрование и аутентификацию данных



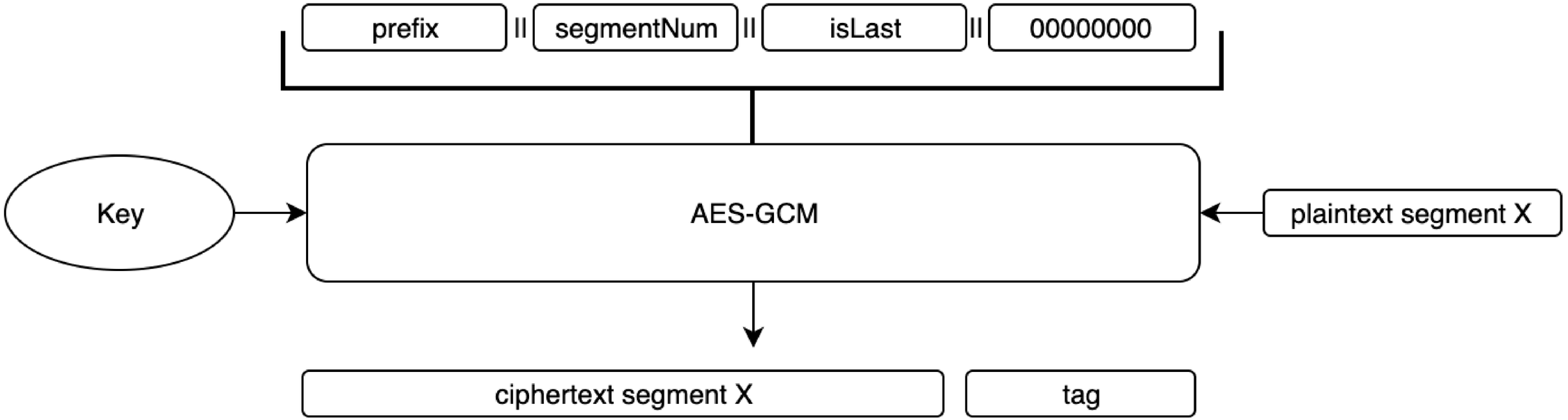
Потоковое шифрование: заголовок



Потоковое шифрование: получение ключа

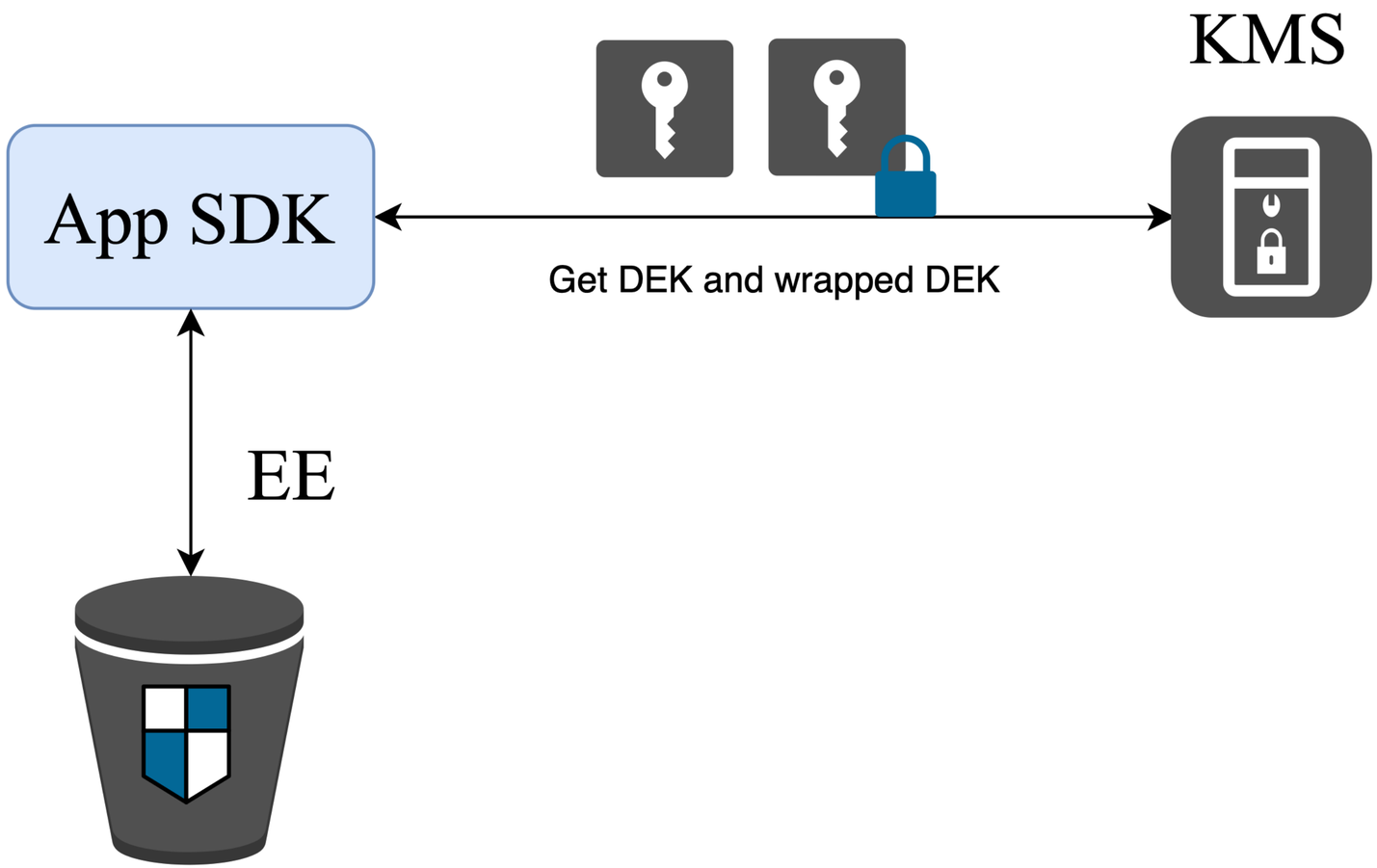


Потоковое шифрование: общая схема

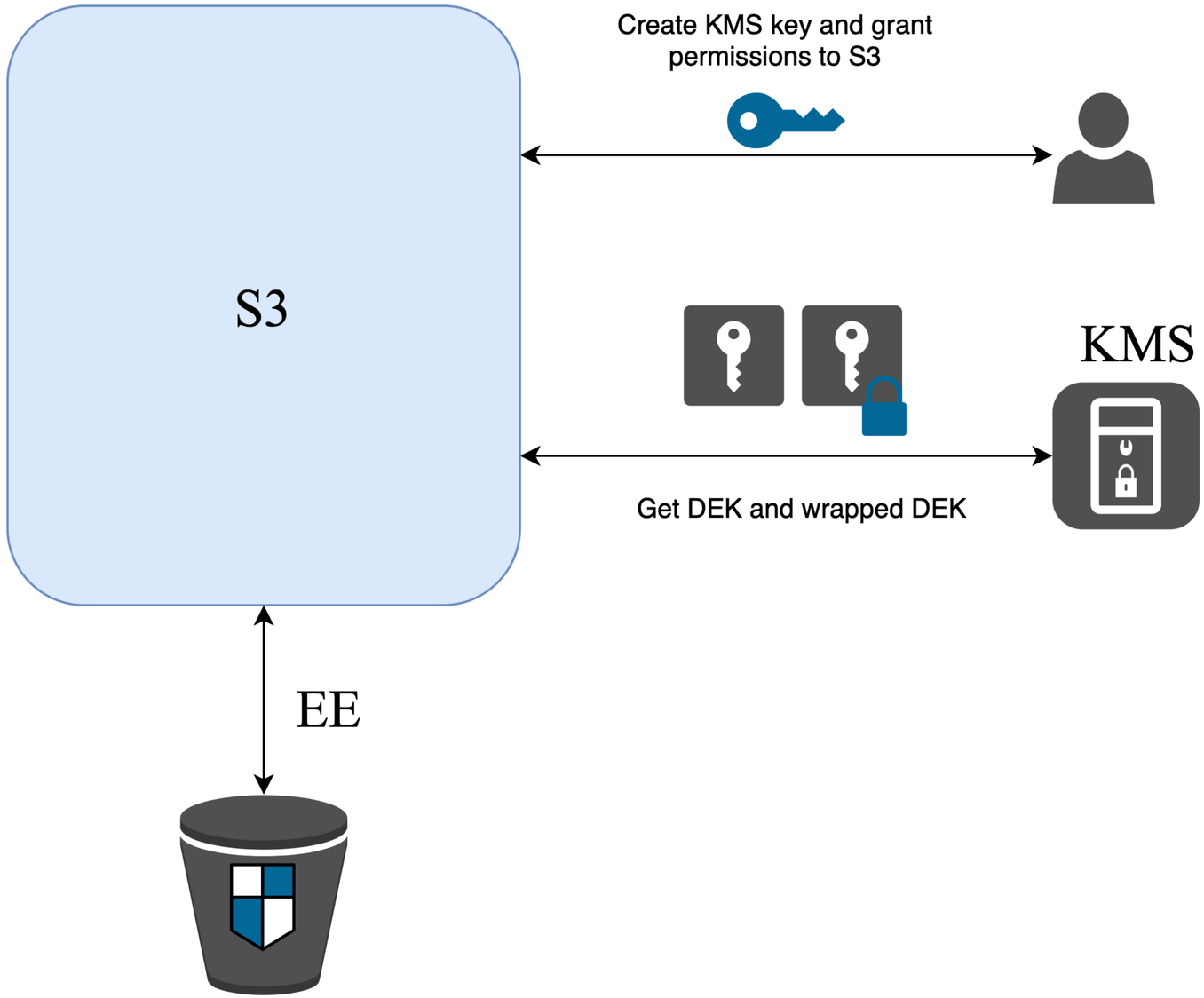


Use cases

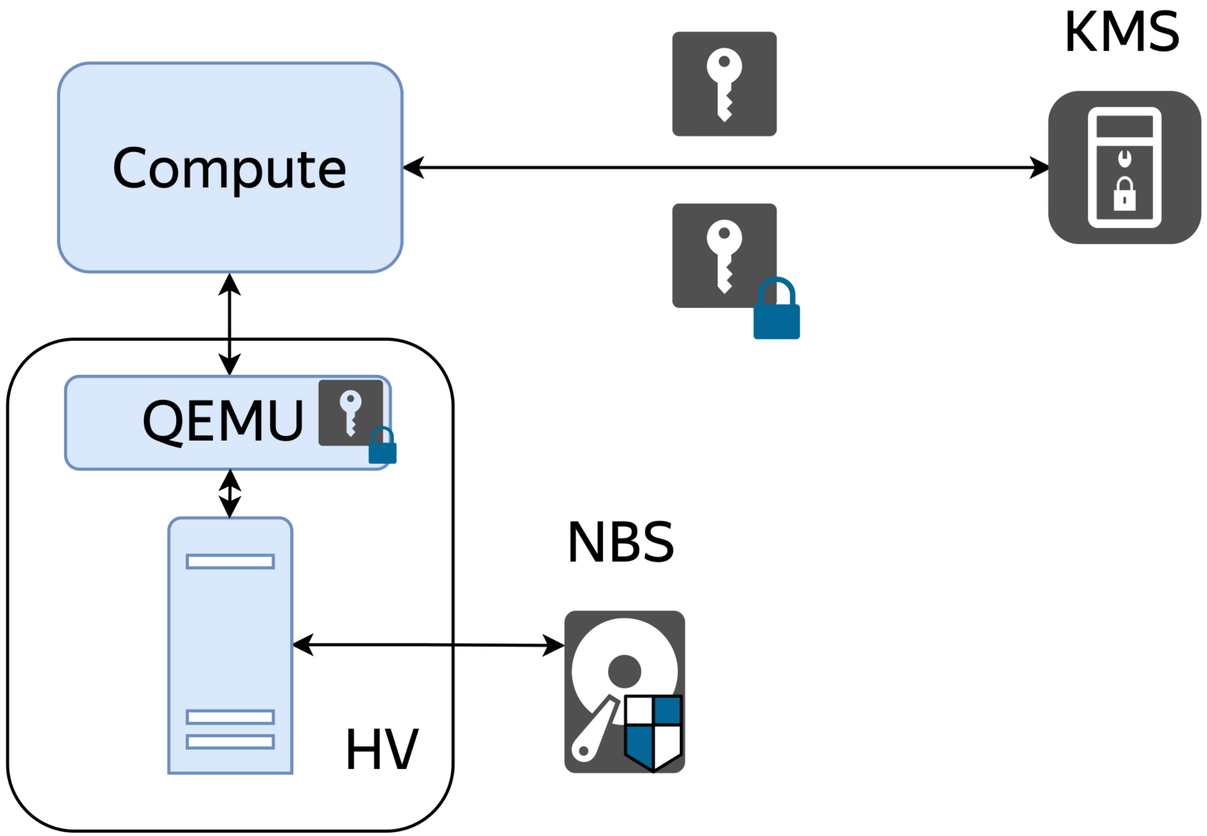
Client-side encryption в S3



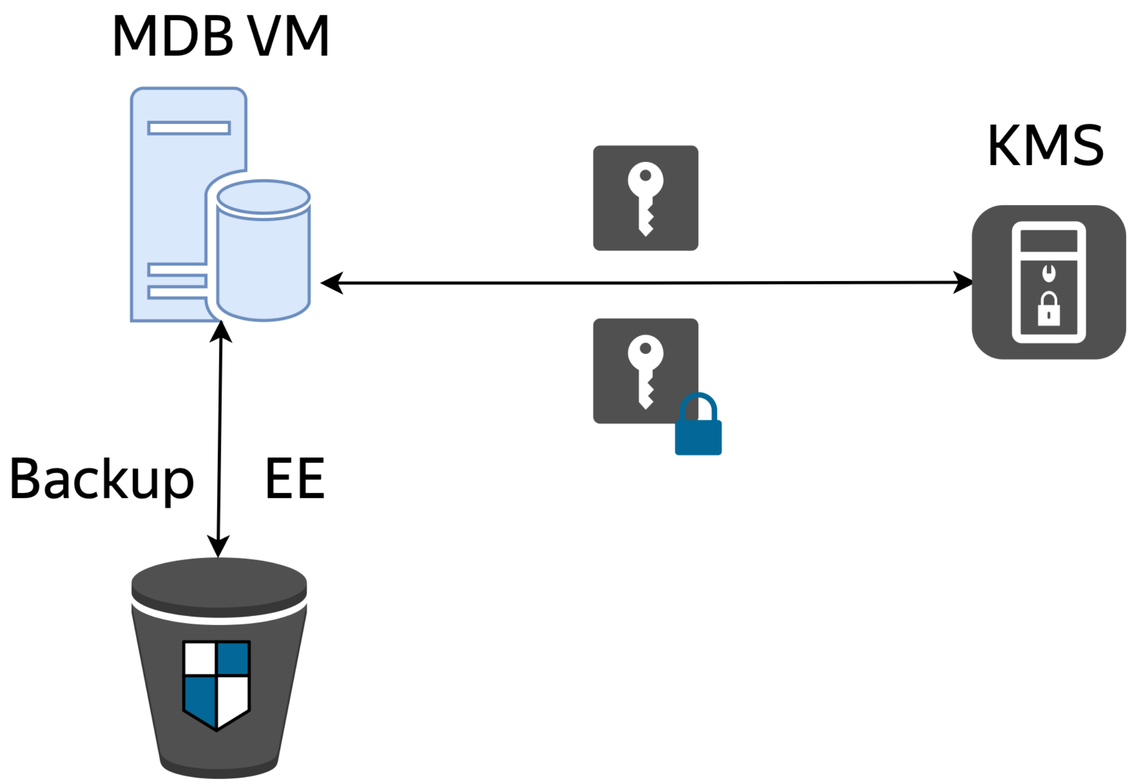
Server-side encryption в S3



Шифрование дисков VM



Шифрование бекапов в MDB



В итоге: что есть KMS?

- Полный жизненный цикл ключей KEK
- Encrypt / Decrypt API
- Шифрование большого объёма данных с помощью envelope encryption
- Аудит использования ключей
- Гранулярный контроль доступа к ключам
- Separation of duties на уровне сервисов - нужно сломать 2 сервиса, чтобы получить доступ к данным
- Гарантированное удаление данных при удалении ключа

Спасибо!