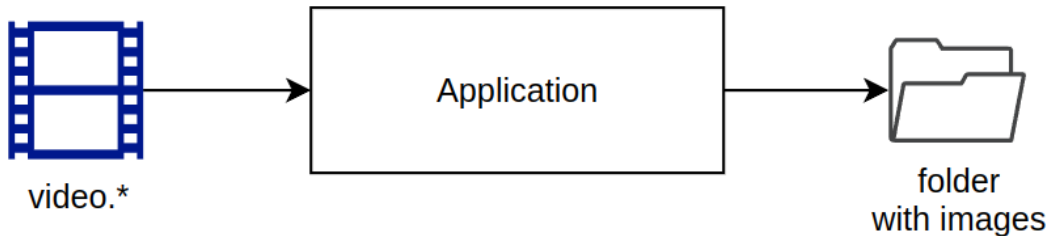


## Задача 1

Задача: написать модуль фрагментирования видео

Цель: написать консольное приложение, вызывающее метод, который видеофайл преобразует в множество отдельных кадров на языке Golang



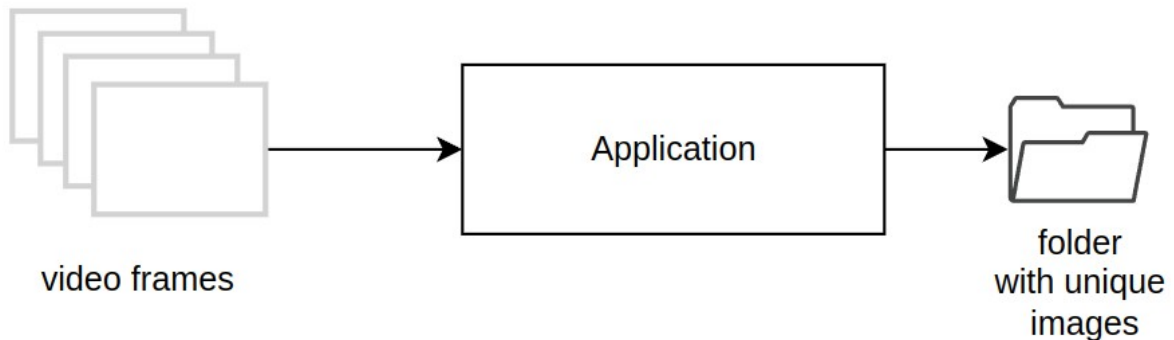
Дополнительным плюсом будет:

1. Использовать популярные библиотеки
2. Передавать видеофайл не единым файлом, а потоком байт
3. Использовать C bindings для вызова процедур ffmpeg

## Задача 2

Задача: написать модуль объединения схожих кадров видео

Цель: написать консольное приложение, вызывающее метод, который прочитывает папку с кадрами видео, выявляет среди них схожие кадры (например сцены) и записывает их в другую папку.



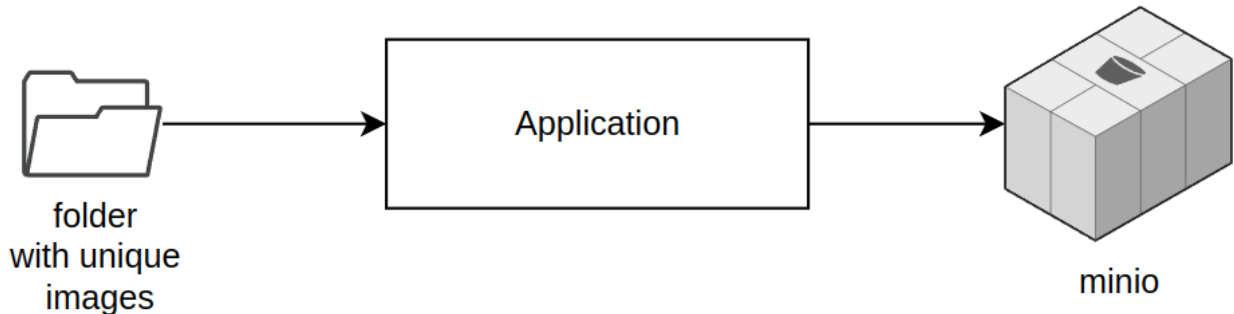
Дополнительным плюсом будет:

1. Использовать популярные библиотеки
2. Работа с потоком изображений (считать что изображения отсортированы)

### Задача 3

Задача: сделать выгрузку изображений в хранилище minio

Цель: написать консольное приложение, вызывающее метод, который прочитывает папку с изображениями и записывает их в папку в хранилище minio



В простейшем случае Minio можно запустить посредством Docker контейнера.

Код для docker-compose:

```
version: '3'

services:
  minio:
    image: minio/minio
    ports:
      - "9000:9000"
      - "9001:9001"
    volumes:
      - minio_storage:/data
    environment:
      MINIO_ROOT_USER: admin
      MINIO_ROOT_PASSWORD: admin
      command: server --console-address ":9001" /data

volumes:
  minio_storage: {}
```

Для справки: <https://hub.docker.com/r/minio/minio>

Дополнительным плюсом будет:

1. Использовать популярные библиотеки
2. Многопоточная запись в хранилище

## Задача 4\*

Задача: реализовать микросервис

Цель: объединить вышеизложенные задачи в один единый модуль

Модуль должен постоянно слушать очередь RabbitMQ и выполнять приходящие задачи. Сообщения для RabbitMQ отправлять вручную через интерфейс. Пример сообщения:

```
{
  "minio_bucket" : "bucket",
  "folder_name"  : "folder"
}
```

Алгоритм работы модуля:

- 1) Открыть видео файл в minio
- 2) Разбивая на кадры, сравнивать их с предыдущим на схожесть. Схожие записывать в папку в minio
- 3) По завершении отправить в очередь с названием «Done» сообщение с путем к итоговой папке формата:

```
{
  "minio_bucket" : "bucket",
  "folder_name"  : "folder"
}
```

Потребуется экземпляр Minio и RabbitMQ. Проще всего их запустить в Docker контейнере. Код для docker-compose:

```
version: '3'

services:
  minio:
    image: minio/minio
    ports:
      - "9000:9000"
      - "9001:9001"
    volumes:
      - minio_storage:/data
    environment:
      MINIO_ROOT_USER: admin
      MINIO_ROOT_PASSWORD: admin
    command: server --console-address ":9001" /data

  rabbitmq:
    image: rabbitmq:3.10.7-management
    hostname: rabbitmq
    restart: always
    environment:
      - RABBITMQ_DEFAULT_USER=rmuser
      - RABBITMQ_DEFAULT_PASS=rmpassword
    ports:
      - 15672:15672
      - 5672:5672

volumes:
  minio_storage: {}
```