Django

Stripe API — python library 지원.

website 간 adding.

Api, SDKs

Payment.

price data and quantity

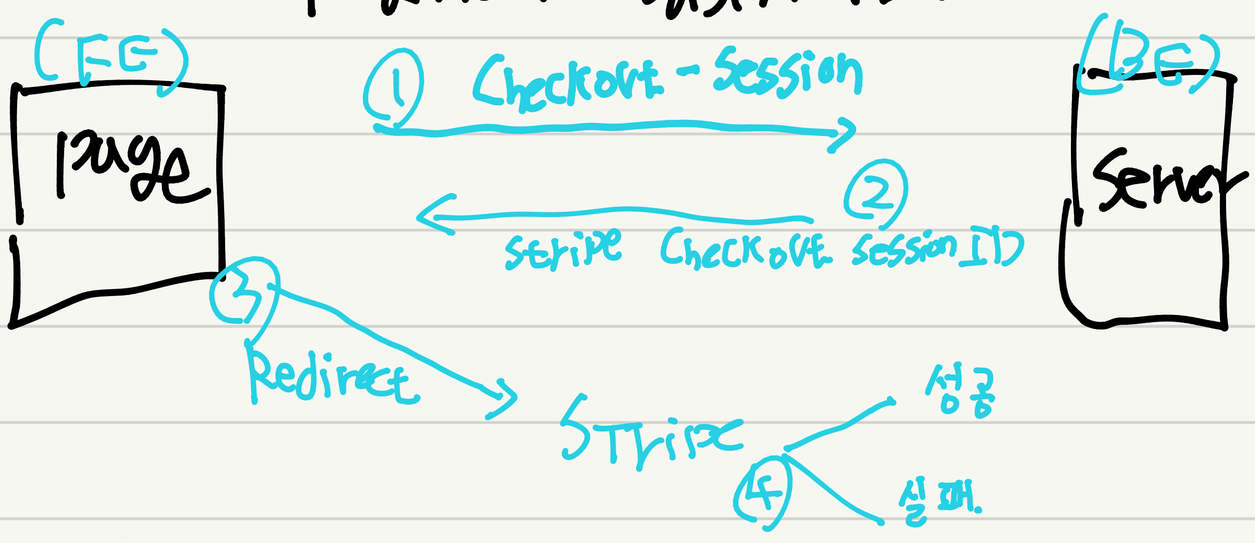
Client-side integration

Server-side integration

Handing success-and-failure redirect.

Handing web hook - to recevie real-time updates about events related to your account and transcation.

Payment System FLow



Celery

- 사용자 request를 사용할 때 꽤 오래걸리는 테스크가 중복 그럴때 사용하는게 Celery

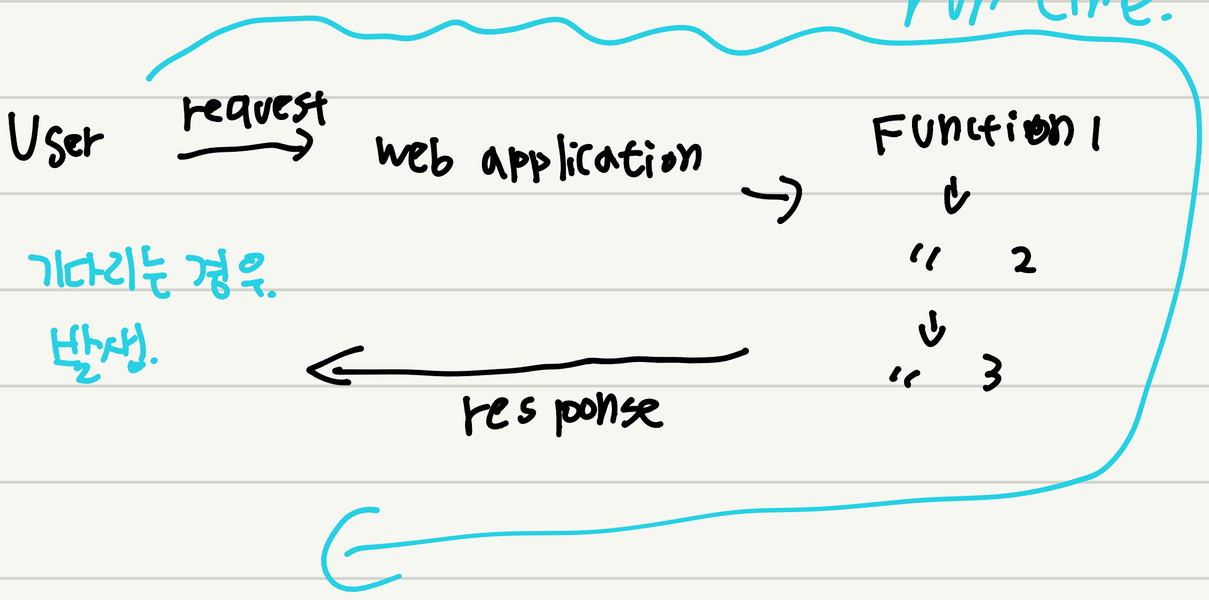
Celery is an asynchronous task queue/job queue based on distributed message passing

- Real-time operation (support scheduling as well)
- Primary used in Python
- key aspect
  - Distribute: workers can be distributed across multiple machines
  - Broker: message broker to mediate between Clients and Workers (redis or RabbitMQ)
  - Backends: store the state and results of tasks (RDB)
  - Tasks: units of work
  - Scheduling: periodic tasks or cron style tasks
  - (동시성) Concurrency: handles concurrency (multiprocessing, Events, or Gevent)
  - Fault Tolerance: designed to be resilient to node failures, with various mechanisms for retrying failed tasks. (재시도)
  - Integration: Django, Flask, and other python webframe works.
  - Workflows: supports complex workflows through Chords, groups and Chains, allowing you to orchestrate tasks
  - Monitoring: Tools like Flower provide real-time monitoring of Celery workers and tasks.

Use Case (사용 경우).

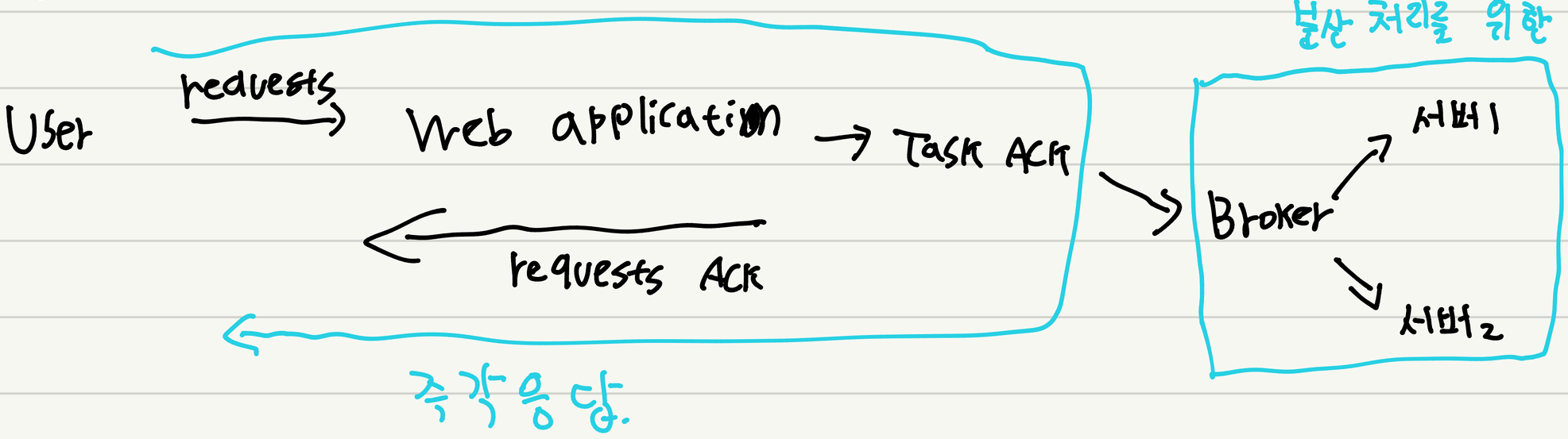
- 머신러닝 시스템.
- email 보낼 때.
- 웹 사이트 스크레이핑, 크롤링
- 리포트 생성.
- 데이터 분석, 프로세싱.

셀러리를 사용 하지 않을 경우.



셀러리를 사용하는 경우

셀러리와 함께 브로커를 사용하는 경우.



일반적인 아키텍처

Publisher — Broker — Consumer & worker — Backend.

Django — Redis — Celery — Database