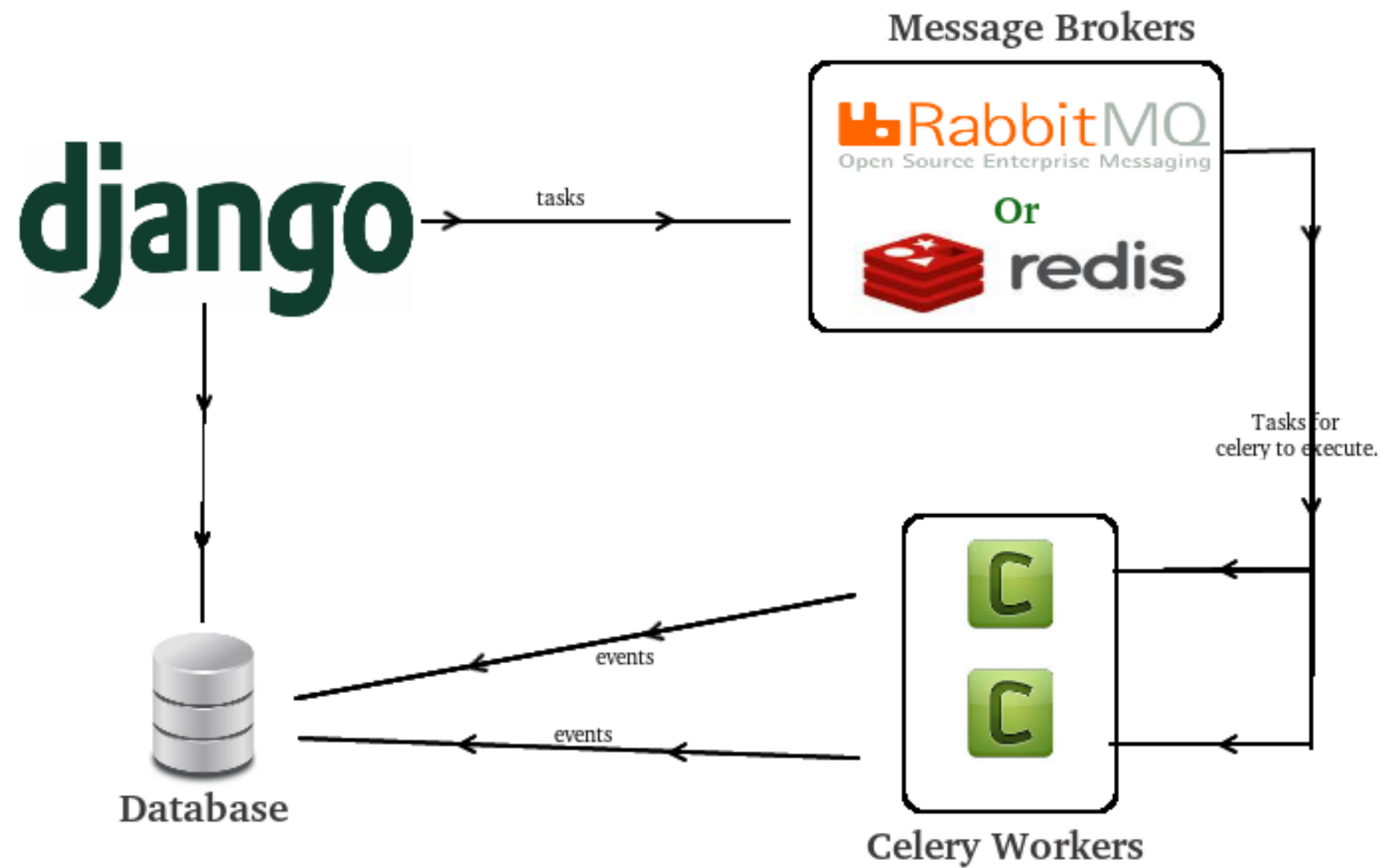




Celery - Distributed Task Queue

- Представляет собой библиотеку для работы с асинхронными задачами
- Имеет широкую популярность (в силу своей необходимости)
- Хорошо работает с такими брокерами как: RabbitMQ, Redis
- Может хранить результаты выполнения задач в базе данных



Что такое broker?

- Брокер хранит в себе очереди задач
- Брокер передает сообщения, с помощью которых общается Celery
- У одного приложения могут быть несколько брокеров

Что такое worker?

- worker - процесс, который выполняет задачи (task), после получения сигнала от брокера
- Складывает результаты выполнения задачи в конечную базу данных (или нет)
- Их может быть много, каждый может обрабатывать разные очереди

Что такое task?

- task - единица выполнения для Celery, чаще всего функция
- task может принадлежать разным очередям (<http://docs.celeryproject.org/en/latest/userguide/routing.html>)
- task может быть выполнен синхронно или асинхронно
- task может быть выполнен в какое-то определенное время или по таймеру (как cron), см.: <http://docs.celeryproject.org/en/latest/userguide/periodic-tasks.html>

```
from celery import Celery

app = Celery('hello', broker='amqp://guest@localhost//')

@app.task
def hello():
    return 'hello world'
```

Минимальное приложение

celery beat

- Команда для запуска периодических задач
- <http://docs.celeryproject.org/en/latest/reference/celery.beat.html>

django-celery

- Модуль для интеграции Django и Celery
- Позволяет использовать дополнительные Django команды: `celerycam`, `celerymon`

celerycam

- Сканирует текущий статус выполнения задач, добавляет результаты внутрь Django
- ``python manage.py celerycam``

celerymon

- Позволяет мониторить статусы выполнения задач в реальном времени
- Устарел, теперь используется Flower: <https://github.com/mher/flower>