

TaskBunny

Background Job Processing with RabbitMQ

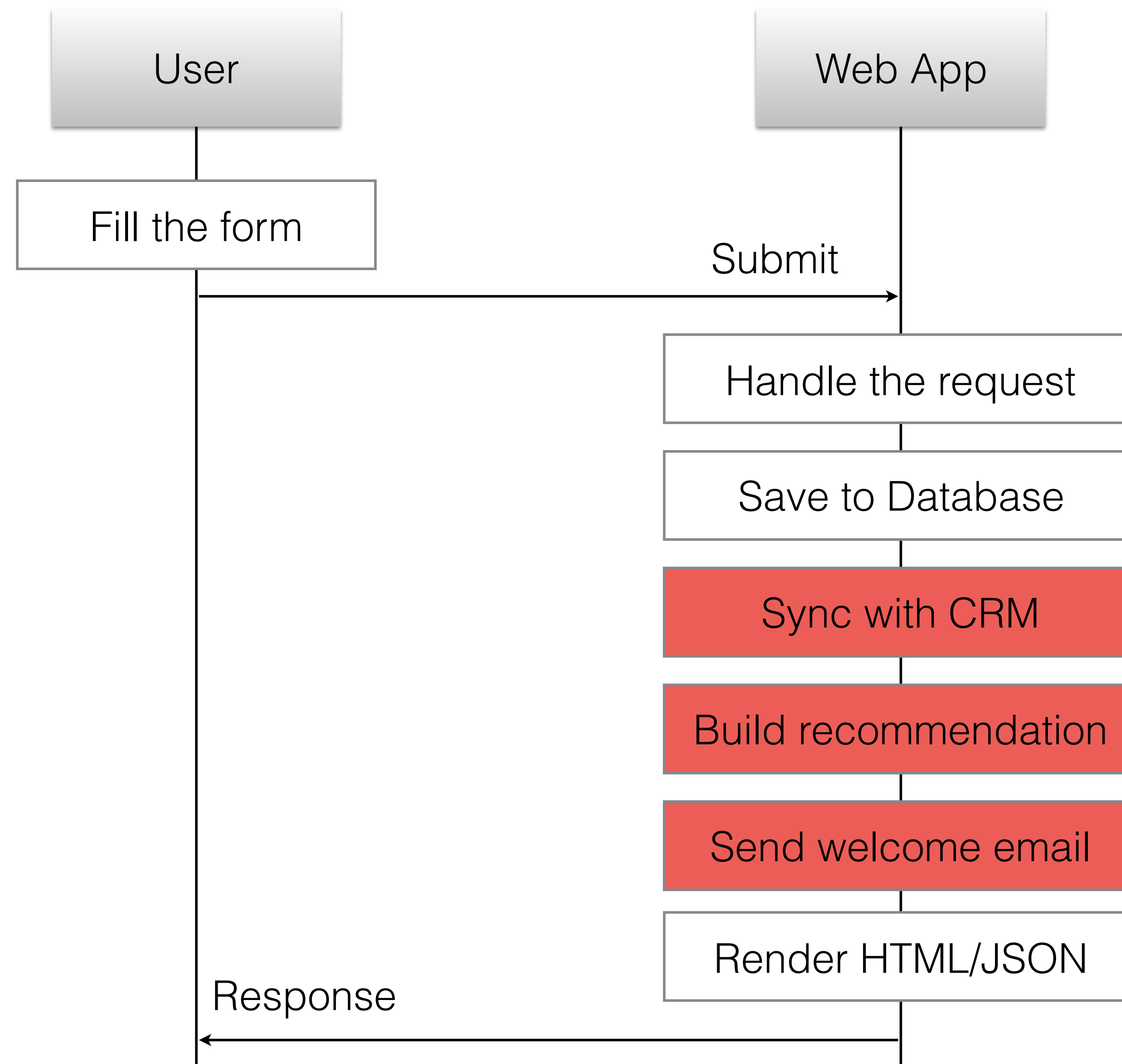
Tatsuya Ono (@ono)

What is Background Job?

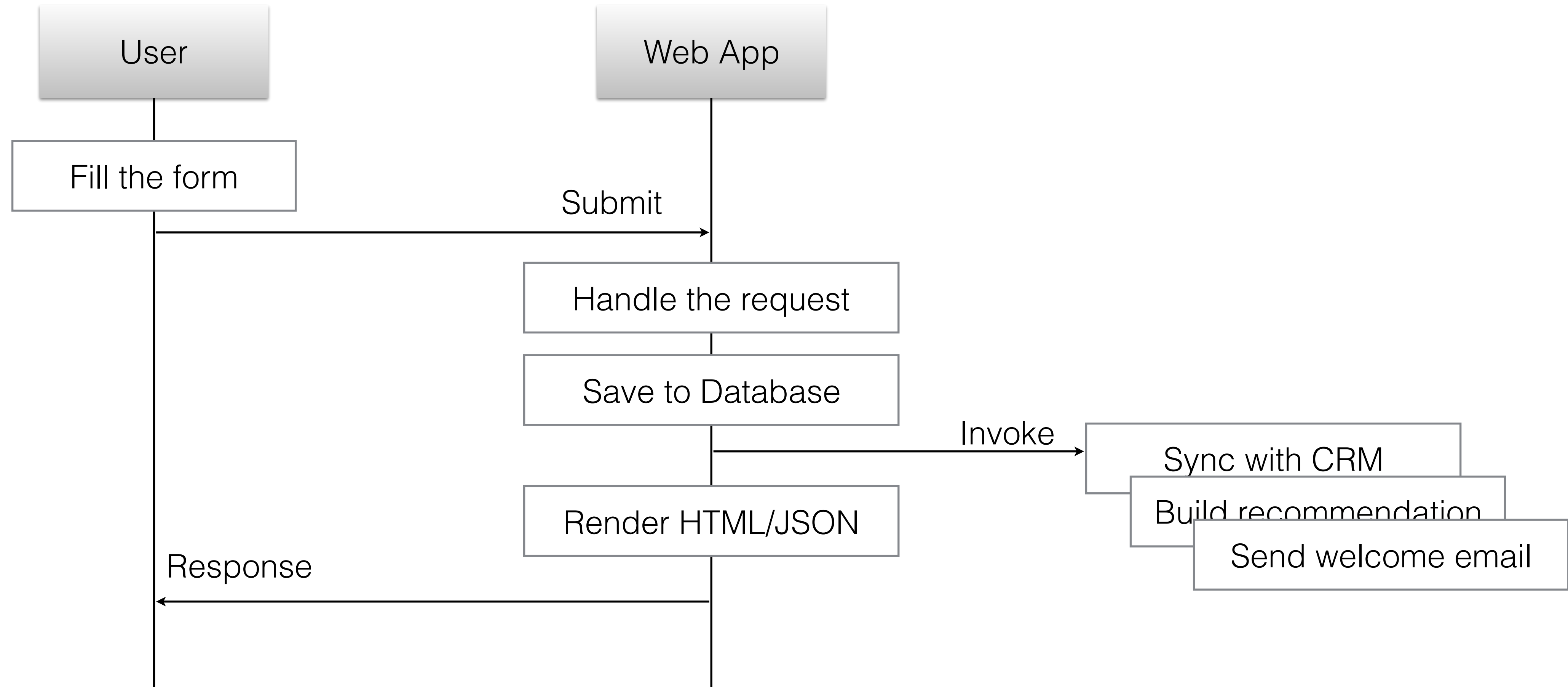
“Background job is **a non-interactive process** that runs behind the normal interactive operations. They run **in parallel** and **do not disturb(block)** interactive processes and operations.”

- guru99.com

Example: User Registration



User Registration with BJ



Background Job in Elixir

```
user = Users.insert params
```

```
spawn MyCRM, :sync, [user]
```

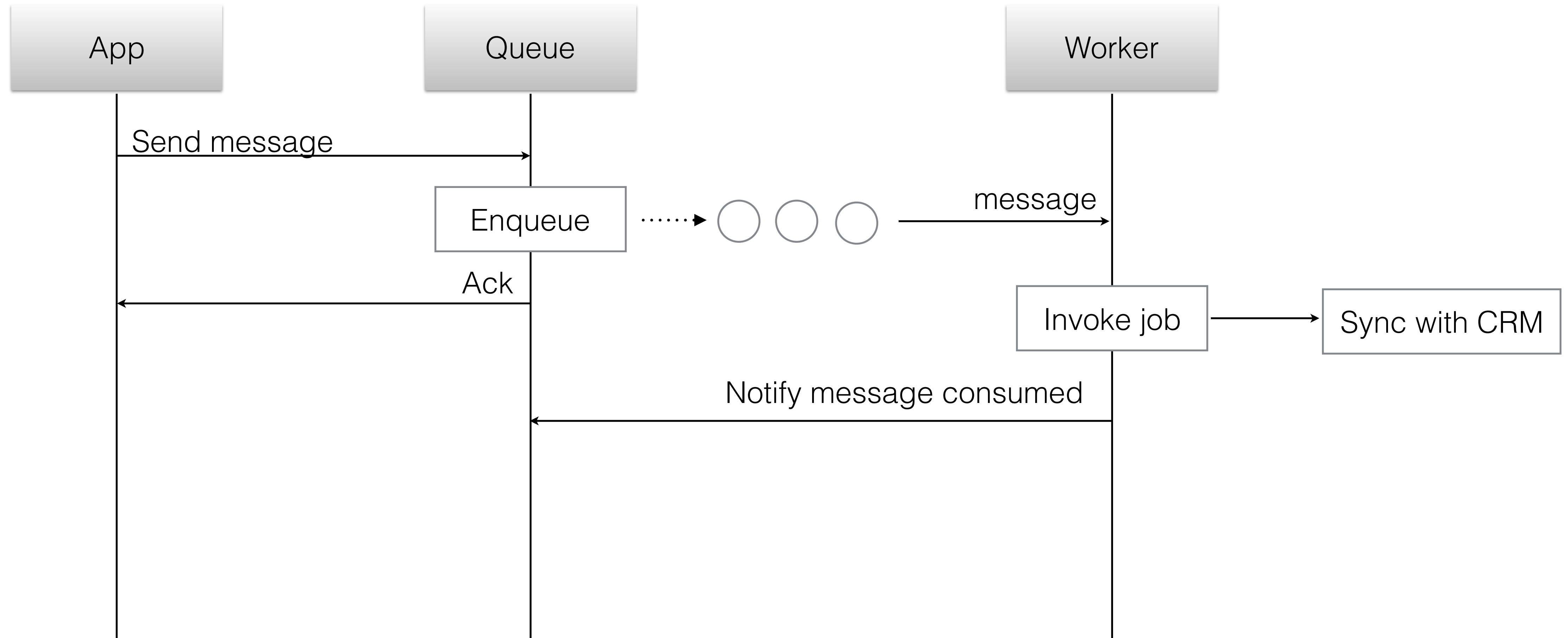
```
render_user user
```

```
user = Users.insert params
```

```
GenServer.cast MyCRM.pid(), {:sync, user}
```

```
render_user user
```

Background Job with Queue



Benefits

- Decouple the request and processing
- Persist requests as messages

It allows you introduce some strategies on job processing

TaskBunny

- Queue based background job processing library
- Using RabbitMQ
- New! - Released 0.1 in April

Background Job with TaskBunny

App

Send message

Ack

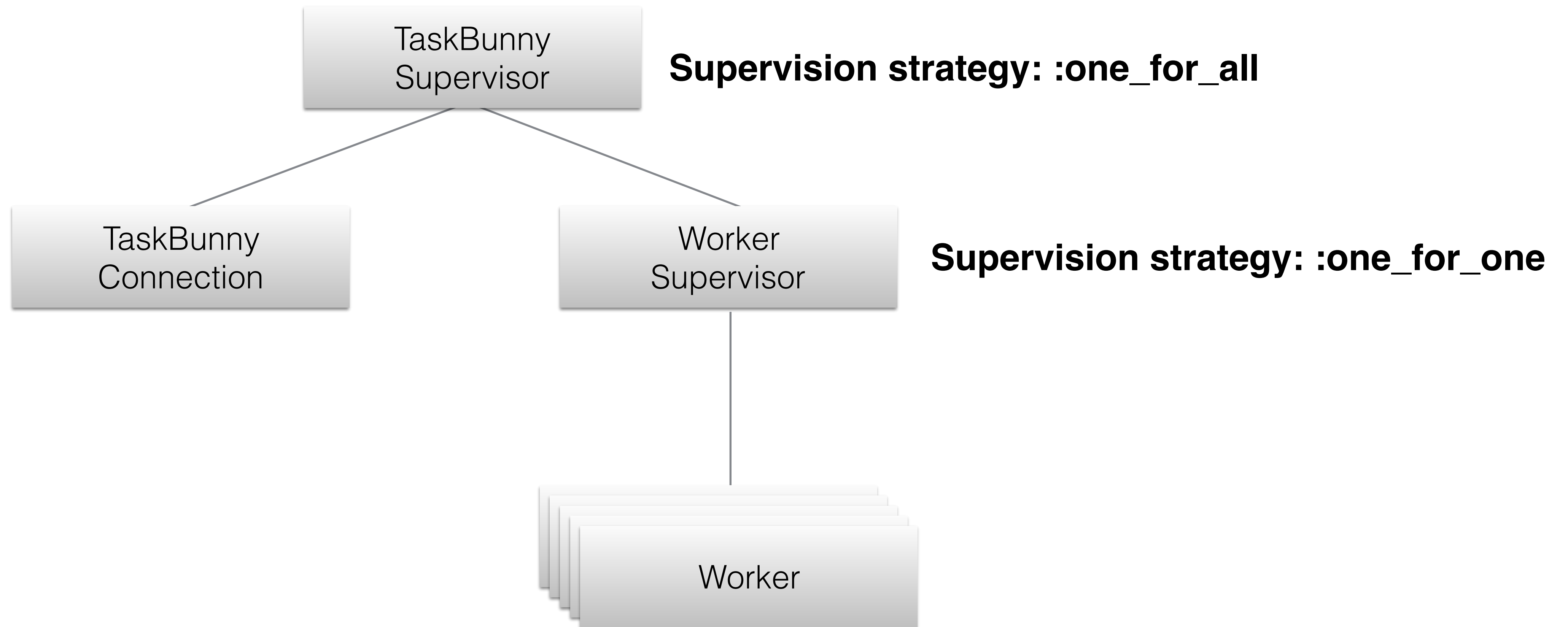


Sync with CRM

Demo - getting started

Demo - concurrency and queues

Supervisor tree



And more...

- Handle failures (:error, exception, exit or timeout) and retry job automatically
- Let you customise retry interval and max retry
- Auto reconnect to RabbitMQ when the connection is gone and back
- Schedule a job e.g. execute the job in 10 minutes
- Wobserver integration (Ian will be talking about it this afternoon)
- https://hexdocs.pm/task_bunny/readme.html

In Future

- Take benchmark and performance optimisation if needed
- Failure backend
- Global concurrency settings among clusters
- Try out and give us feedback!

Thanks

SQUARE ENIX for supporting the development

(They are hiring!)

