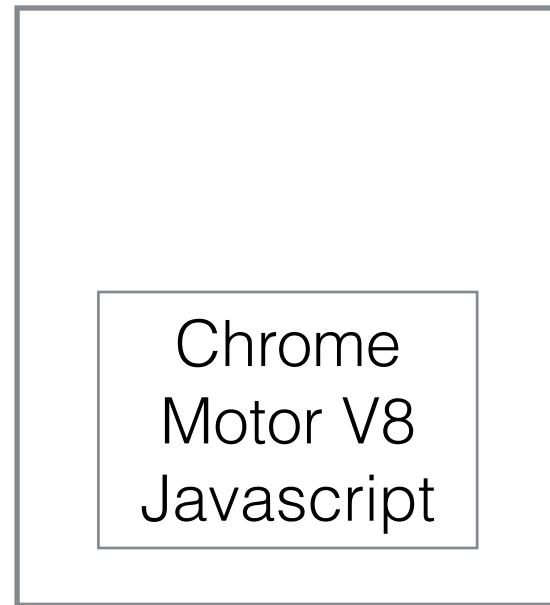
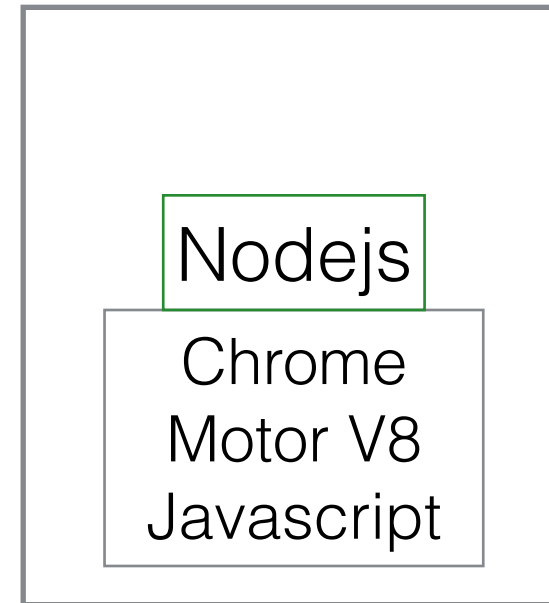


Node.js

- Plataforma construida sobre el Javascript runtime de Chrome (<https://code.google.com/p/v8/>) para fácilmente construir aplicaciones rápidas y escalables.
- Usa un modelo dirigido por eventos, Non-blocking I/O que lo hace liviano y eficiente, perfecto para aplicaciones de datos intensivos en real-time a través de dispositivos distribuidos.



Browser



Servidor

Cómo funciona?

- Trabaja con un único hilo de ejecución. Este hilo organiza todo el trabajo que le mandamos a ejecutar.

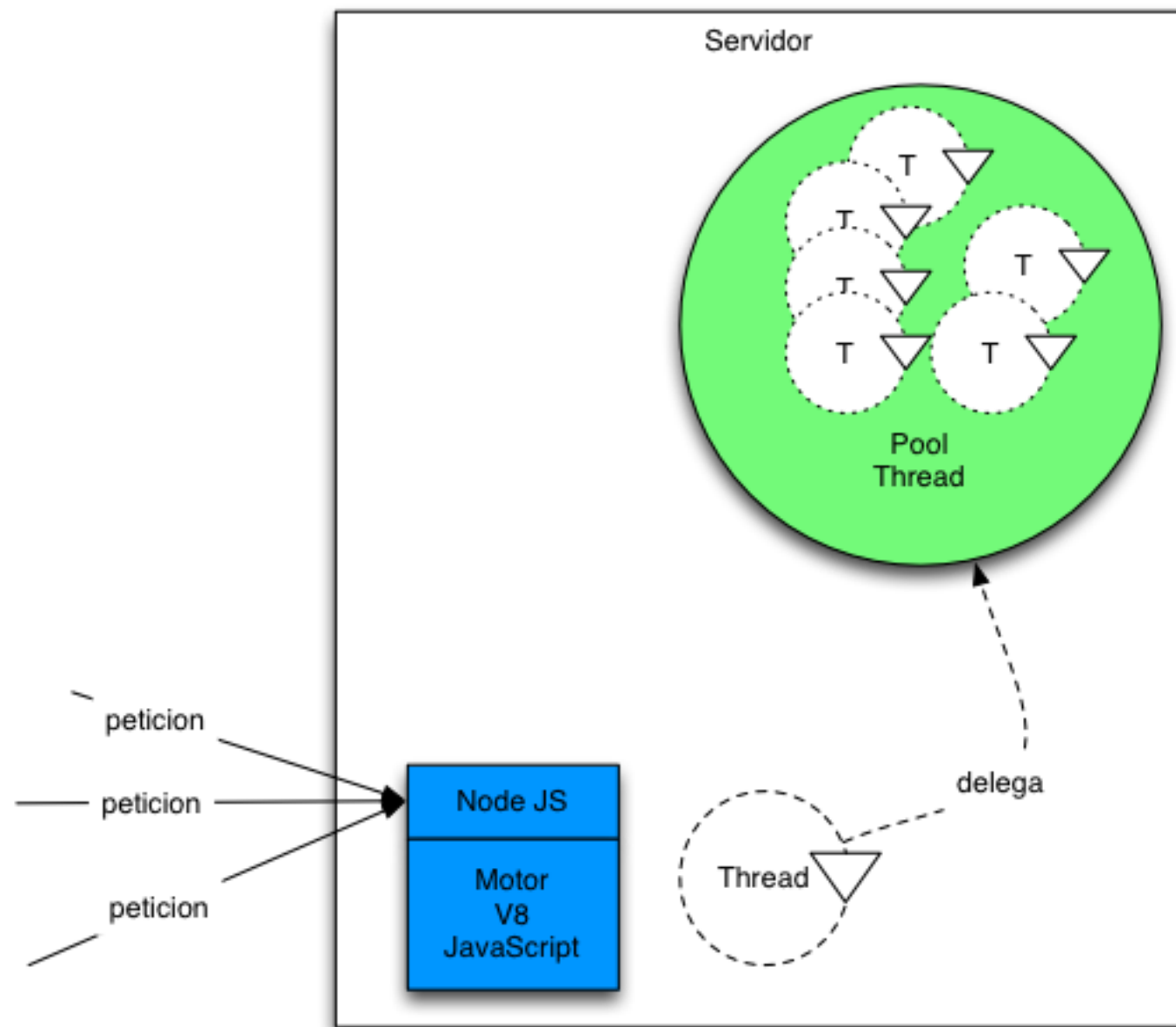
Un solo hilo?

Si una de las tareas que
e mandamos a realizar
toma mucho tiempo?

Respuesta:

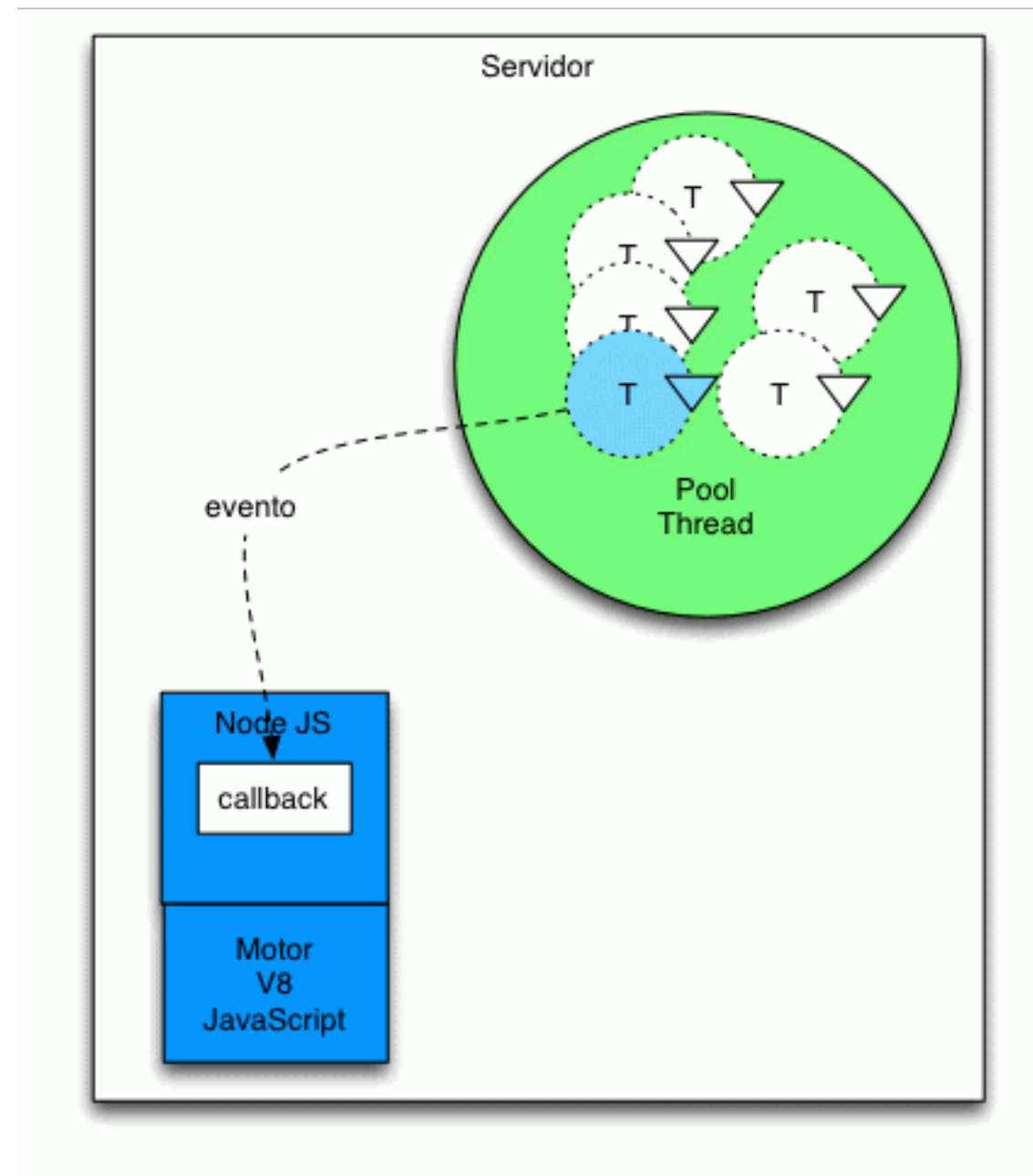
Ejecuciones Asíncronas

- Nodejs delega el trabajo a un pool de threads



libuv

- Dispone de su propio entorno de multi thread asíncrono.
- Nodejs envía el trabajo a realizar al pool.
- Libuv emitirá un evento que será recibido por Nodejs, una vez el trabajo haya sido terminado.



Ejemplo:
node server.js

```
1  var http = require("http");
```

- Requiere al módulo http.
- Este viene incluido en Nodejs.
- Es accesible a través de la variable “http”.

```
3 ▼ http.createServer(function(request, response) {
```

- http crea un servidor a través de *createServer*.
- Esta función retorna un objeto.
- Este objeto tiene una función llamada *listen*.


```
7    }).listen(8888);
```

- *listen* recibe un valor numérico que representa al puerto que http va a escuchar.

Otra forma:

```
var http = require("http");  
var server = http.createServer();  
server.listen(8888);
```

Orientado al evento

Callbacks!

Gracias

Comentarios, preguntas, sugerencias, críticas...?