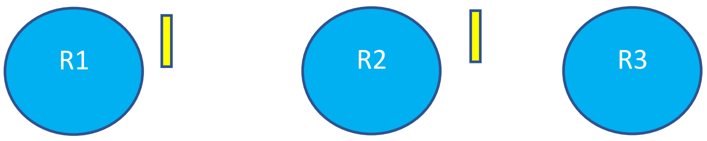
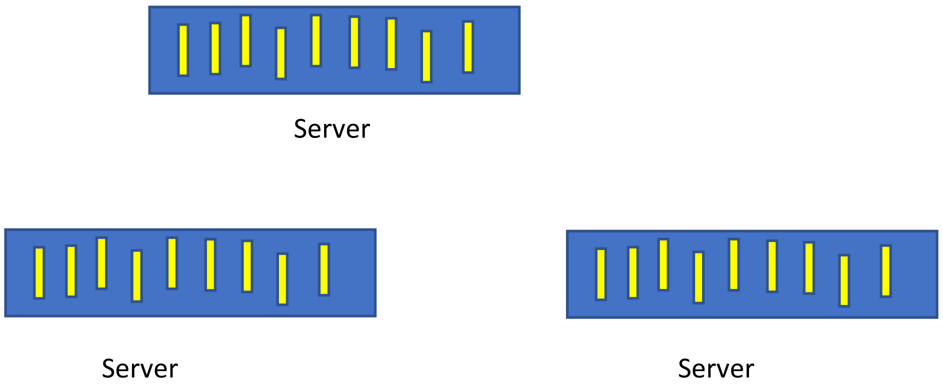
Node applications are **highly scalable, data-intensive** and **real-time** apps this is because of the **non-blocking Asynchronous** nature of Node.

The problem with blocking/synchronous architecture

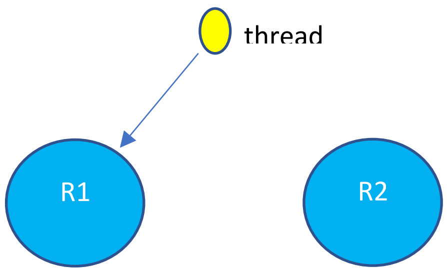
When we receive a request on the server a thread is allocated for handling that request, as part of handling that request, it is likely to quarry a database, as you know some times it may take a little while until the result is ready. When the database is executing the query that thread is waiting. It can't be use to serve other request, so then a new thread serve the request,

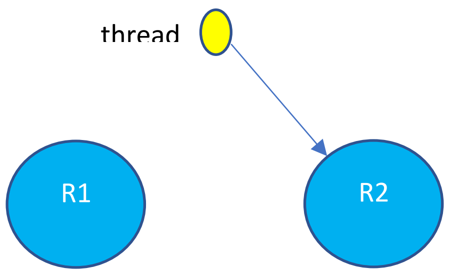


Imagine what we have a large number of concurrent clients at some point we are going to use all thread to serve all the clients, so new client has to wait until free threads are available or



(if we don't want them to wait, we need more hardware, in this type of architecture we are not utilizing the resources efficiently this is the problem with blocking/synchronous architecture). This is how framework like asp.net work by default (we can make it asynchronous we have to work for it.)





But Node applications are asynchronous by default, In Node there is a single thread to handle requests, so single thread is used to handle multiple requests. If we need to quarry a database our thread doesn't have to wait for the dB return the data, while that dB is executing the query the thread is used to serve another client, when the dB prepares the result it put a message in the Event queue, node is continuously monitoring these ques in the background when it finds an event in the queue it will take it out and process it. This kind of architecture make node ideal for building applications that include lot of network access we can serve more clints with out the need of slowing more hard ware and that's why node applications are highly scalable.

Node should not be used for CPU intensive applications, like video encoding or image manipulation applications.

- It will be slow if there are more calculations.