

The next one is the layered system constraint. This is a pretty easy one, actually.

A REST-based solution can be compromised of multiple architectural layers, just as almost all application architectures we use today. These layers can be modified, added, removed, can be physical or logical, but no one layer can directly access a layer that's beyond the next one.

That also means that a client cannot tell whether it's directly connected to the final layer or to another intermediary along the way. So, REST restricts knowledge to a single layer, the outer-facing one, and that reduces the overall system complexity.