𝗟𝗼𝗮𝗱 𝗕𝗮𝗹𝗮𝗻𝗰𝗲𝗿 𝘃𝘀. 𝗥𝗲𝘃𝗲𝗿𝘀𝗲 𝗣𝗿𝗼𝘅𝘆 𝘃𝘀. 𝗔𝗣𝗜 𝗚𝗮𝘁𝗲𝘄𝗮𝘆  
  
All three are used to optimize and manage web traffic. However, they vary in their function and use cases:  
  
𝐋𝐨𝐚𝐝 𝐁𝐚𝐥𝐚𝐧𝐜𝐞𝐫: A load balancer is a device that distributes network or application traffic across a number of servers to ensure that no single server bears too much demand. This helps to increase concurrency, and reliability of applications by 'balancing' the load among various servers.  
  
𝐑𝐞𝐯𝐞𝐫𝐬𝐞 𝐏𝐫𝐨𝐱𝐲: A reverse proxy, also known as an "inbound" proxy, is a server that receives requests from the Internet and forwards them to a small set of servers, typically located on an internal network. It provides a point of control and can provide security, logging, and even load balancing. The client is unaware it is communicating with a set of servers instead of a single server.  
  
𝐀𝐏𝐈 𝐆𝐚𝐭𝐞𝐰𝐚𝐲: An API gateway is an API management tool that sits between a client and a collection of backend services, acting as a single point of entry for a defined group of microservices. In addition to accommodating direct requests, it can also invoke multiple back-end services and aggregate the results, perform format transformations, handle real-time processing, and implement security policies like OAuth.  
  
𝐃𝐢𝐟𝐟𝐞𝐫𝐞𝐧𝐜𝐞 𝐢𝐧 𝐏𝐮𝐫𝐩𝐨𝐬𝐞:  
🔹 A load balancer distributes traffic for the purpose of optimal resource utilization, maximizing throughput, minimizing response time, and avoiding system overload.  
🔹 A reverse proxy protects servers from traffic by intercepting requests and managing them.  
🔹 An API Gateway, apart from handling requests, also manages and coordinates multiple microservices running behind it.  
  
𝐃𝐢𝐟𝐟𝐞𝐫𝐞𝐧𝐜𝐞 𝐢𝐧 𝐔𝐬𝐞 𝐜𝐚𝐬𝐞𝐬:  
🔹 Load balancers are used when there is heavy traffic to servers and we need to ensure the load is evenly distributed.  
🔹 Reverse proxies are used to control and protect access to servers in internal networks, manage SSL encryption, or serve static content.  
🔹 API Gateways are used in microservices architecture where there are a number of service endpoints and there's a need for functionalities like request routing, composition, and protocol translation.  
  
Ref  
1. 𝗟𝗼𝗮𝗱 𝗕𝗮𝗹𝗮𝗻𝗰𝗲𝗿 𝘃𝘀. 𝗥𝗲𝘃𝗲𝗿𝘀𝗲 𝗣𝗿𝗼𝘅𝘆 𝘃𝘀. 𝗔𝗣𝗜 𝗚𝗮𝘁𝗲𝘄𝗮𝘆: 𝗗𝗲𝗺𝘆𝘀𝘁𝗶𝗳𝘆𝗶𝗻𝗴 𝗪𝗲𝗯 𝗔𝗿𝗰𝗵𝗶𝘁𝗲𝗰𝘁𝘂𝗿𝗲𝘀: <https://lnkd.in/gfzyT42k>

2. 𝗚𝗿𝗼𝗸𝗸𝗶𝗻𝗴 𝗦𝘆𝘀𝘁𝗲𝗺 𝗗𝗲𝘀𝗶𝗴𝗻 𝗙𝘂𝗻𝗱𝗮𝗺𝗲𝗻𝘁𝗮𝗹𝘀: <https://lnkd.in/gtcCT-dJ>

-------  
Follow [Tauseef Fayyaz](https://www.linkedin.com/in/ACoAACogvGkBsaFqvSPBkBwBwBpj-QjPY9YEKss) for more amazing content!  
  
Book your 1:1 sessions for more personalized tips and mentorship at [topmate.io/tauseeffayyaz](http://topmate.io/tauseeffayyaz)

Credits: [Design Gurus](https://www.linkedin.com/company/designgurus/)  & [Arslan Ahmad](https://www.linkedin.com/in/ACoAAACdXlIBSIWCXfeStv-9Dd9-ej-_9oH-Fxc)  
  
[#loadbalancer](https://www.linkedin.com/feed/hashtag/?keywords=loadbalancer&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7126459526760869888) [#proxy](https://www.linkedin.com/feed/hashtag/?keywords=proxy&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7126459526760869888) [#apigateway](https://www.linkedin.com/feed/hashtag/?keywords=apigateway&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7126459526760869888) [#systemdesign](https://www.linkedin.com/feed/hashtag/?keywords=systemdesign&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7126459526760869888) [#interviewtips](https://www.linkedin.com/feed/hashtag/?keywords=interviewtips&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7126459526760869888) [#learning](https://www.linkedin.com/feed/hashtag/?keywords=learning&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7126459526760869888) [#architecture](https://www.linkedin.com/feed/hashtag/?keywords=architecture&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7126459526760869888) [#microservices](https://www.linkedin.com/feed/hashtag/?keywords=microservices&highlightedUpdateUrns=urn%3Ali%3Aactivity%3A7126459526760869888)

Activate to view larger image,

