Responsibilities, services or components? of…

1) Client 2) Carrier/Internet 3) Server

Client = Browser	Internet, TCP/IP, HttpRequest, HttpResponse, Https, SSL, …	Server = Web server, Application container, Database server
• Responsibilities, characteristics, features, services, capabilities, and/or components?	• Responsibilities, characteristics, features, services , capabilities, and/or components?	• Responsibilities, characteristics, features, services , capabilities, and/or components?

Client = Browser and front-end application	Internet, TCP/IP, HttpRequest, HttpResponse, Https, SSL, …	Server = Web server, Application container, Database server
• Sends the requests based on link or send button clicks and JavaScript events (AJAX) • Parses and renders HTML, CSS • Downloads and shows images and media • Maintains the DOM • Provides UI for navigation (links, etc.) • Provides form controls and form submissions (usually HTTP POST) • UI event handling, like mouse dragging etc. • Runs JavaScript • Runs AJAX request/responses too • Validation of input (client-side) • Session management (client-side) • Saves cookies • Caching. And Local storage. Debugger. • Can fetch content from many servers • Maintains hidden info sent and later again accessed by the server, usually encrypted, (like antiforgerytokens, auth_key, api_key) and so on. • Support for the carrier security (client-side)	• Security: Encryption of data • Security: We know that the ends are who they say they are • Routing packages • Reliability, package handling • Cache proxies	• Routing, Dispatching • Hiding the implementation • Loose coupling of URL to implementing components, • Restful URIs • Runs code, maintains object structures • Interprets and parses requests (incl. AJAX) • Writes responses e.g. as HTML/JSON/XML • Validation of input (server-side) • Session management (server-side) • Authentication of users **and** client apps **and** requests • Authorization (user->items, role->actions) • Conventions/Annotations/Configurations • Templates helping (Spring Boot, MVC..) • Database, Database connection, Database user authentication and authorization, (Database encryption) • Support for the carrier security (server-side) • Web Services, sockets, …

MOST LIKELY FEATURES STILL MISSING FROM ALL OF THESE LISTS!

AMAZED?