Two independent columns

Left and right columns hold two unrelated essays. A naive left-to-right text extraction that ignores column boundaries will interleave sentences from both and produce nonsense; a correct one keeps each column intact.

Caching strategies for web APIs

Caching trades staleness for speed by storing a response so a later request can skip the work that produced it. The hardest part is rarely writing the cache. It is deciding when an entry stops being valid and picking a key that will not collide with a slightly different request.

HTTP gives you two matching mechanisms for this: validators like ETag and Last-Modified let a client ask "has this changed?" and get a cheap 304 back instead of the full body, while Cache-Control headers let a server declare how long a response can be reused without asking at all.

Application-level caches add another layer on top, usually keyed by something more specific than a URL (a user id, a query hash, a feature flag combination) because two requests to the same path can still deserve two different cached answers.

Why timezones are hard for software

A timezone is not a fixed offset from UTC; it is a set of rules that can change based on legislation, and those rules are applied differently depending on the exact date and time in question, not just the place.

Daylight saving transitions create two kinds of trouble: a "spring forward" gap where a clock time like 2:30am never occurs at all, and a "fall back" overlap where a clock time like 1:30am occurs twice on the same night, once before the shift and once after.

Storing an instant as UTC and converting only for display sidesteps most of this. But any software that stores a *local* wall-clock time, like a recurring meeting at 9am or an alarm, has to re-resolve that local time against current timezone rules every time it is read.