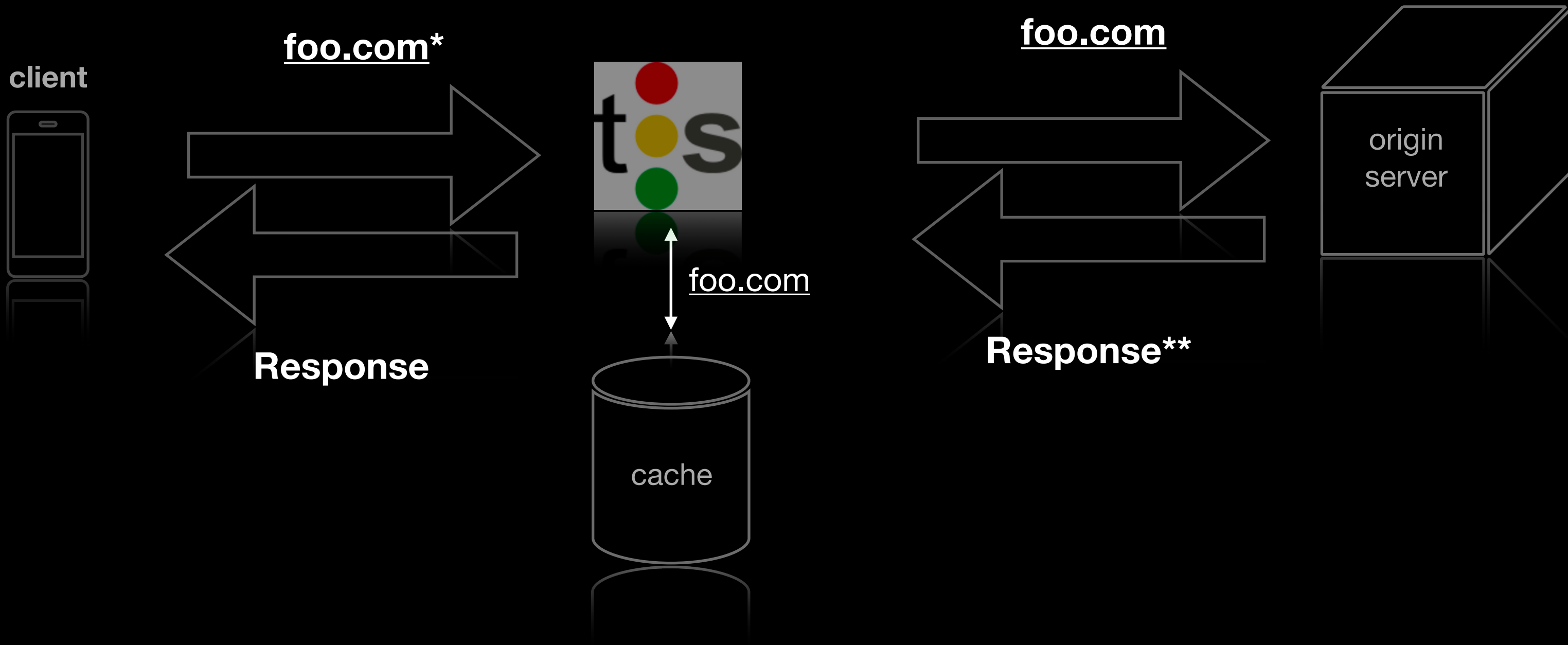


HTTP Cache Optimizations

-Vijay Mamidi

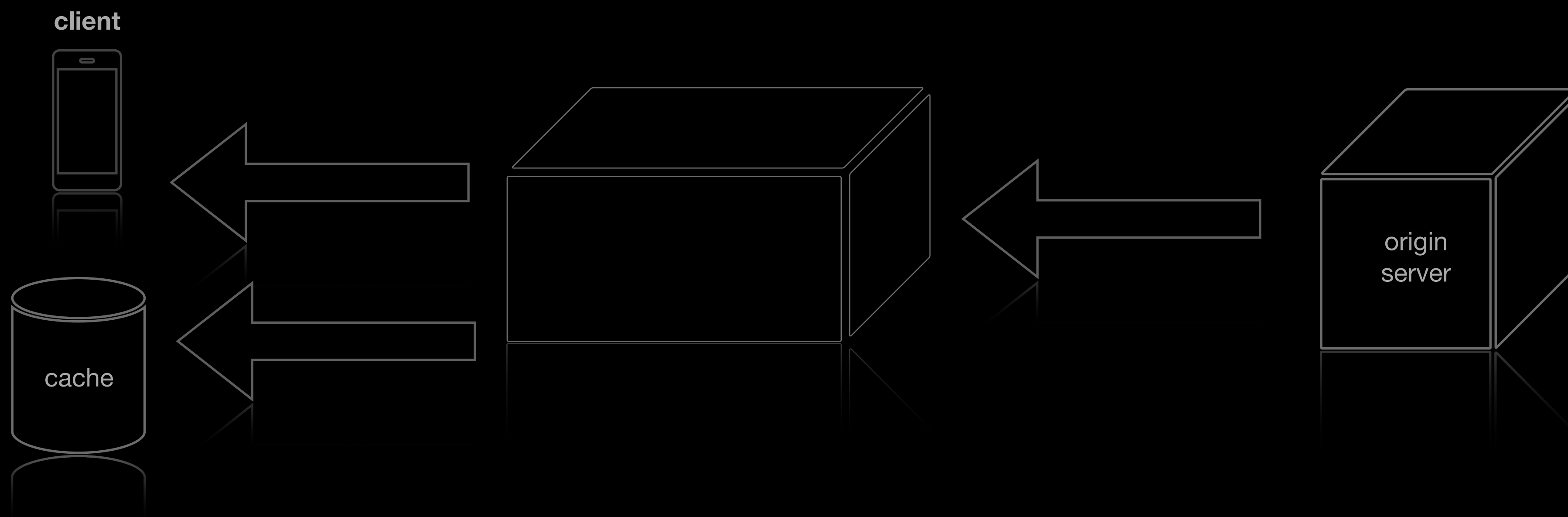
Cache MISS



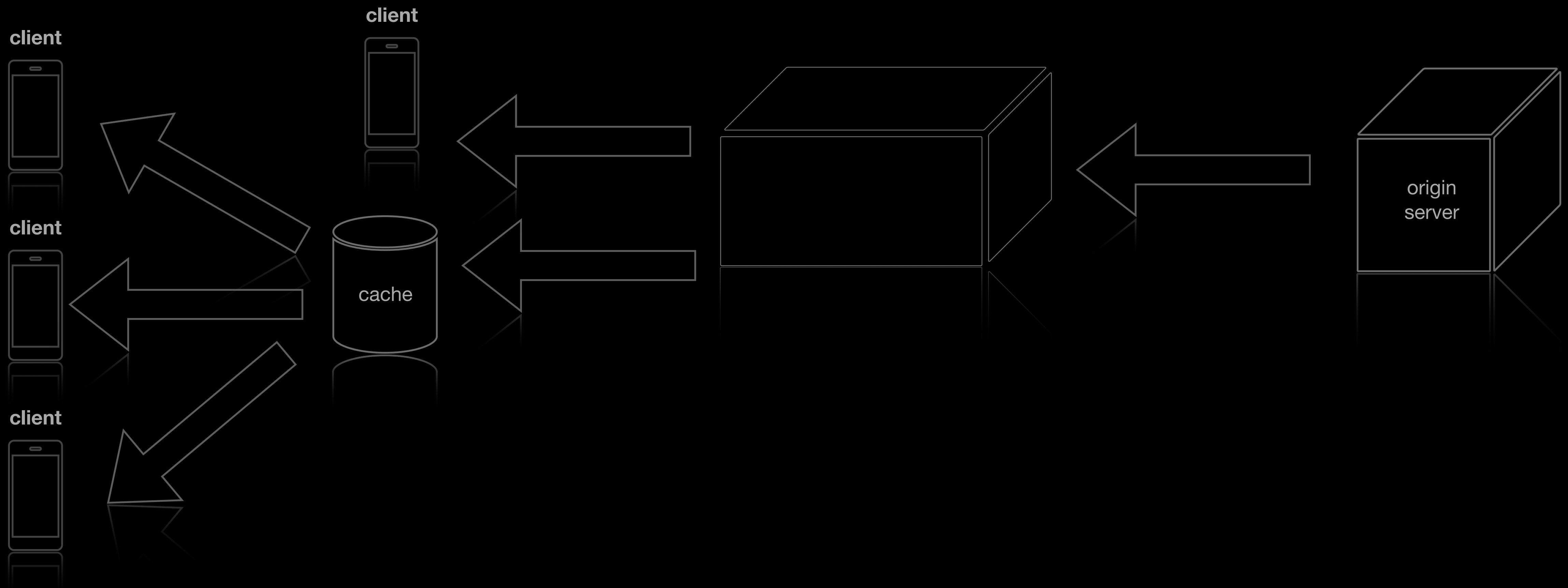
*Request can NOT be served from cache

**Request/Response can be cached

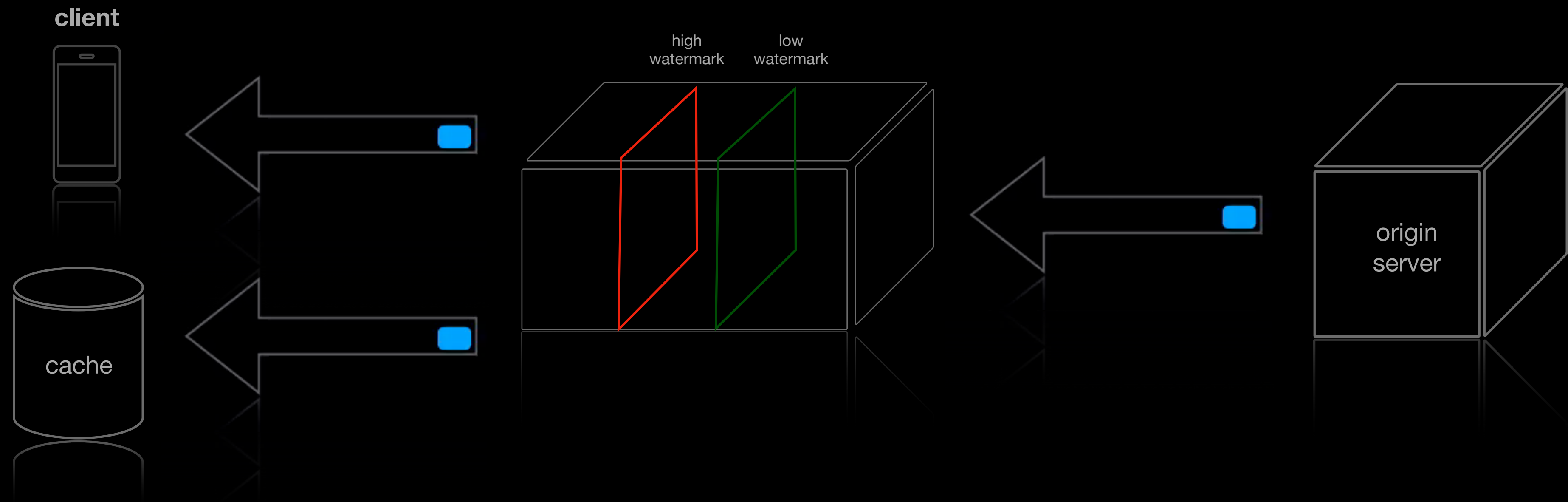
Tunneling



Tunneling with Read While Write



Flow Control

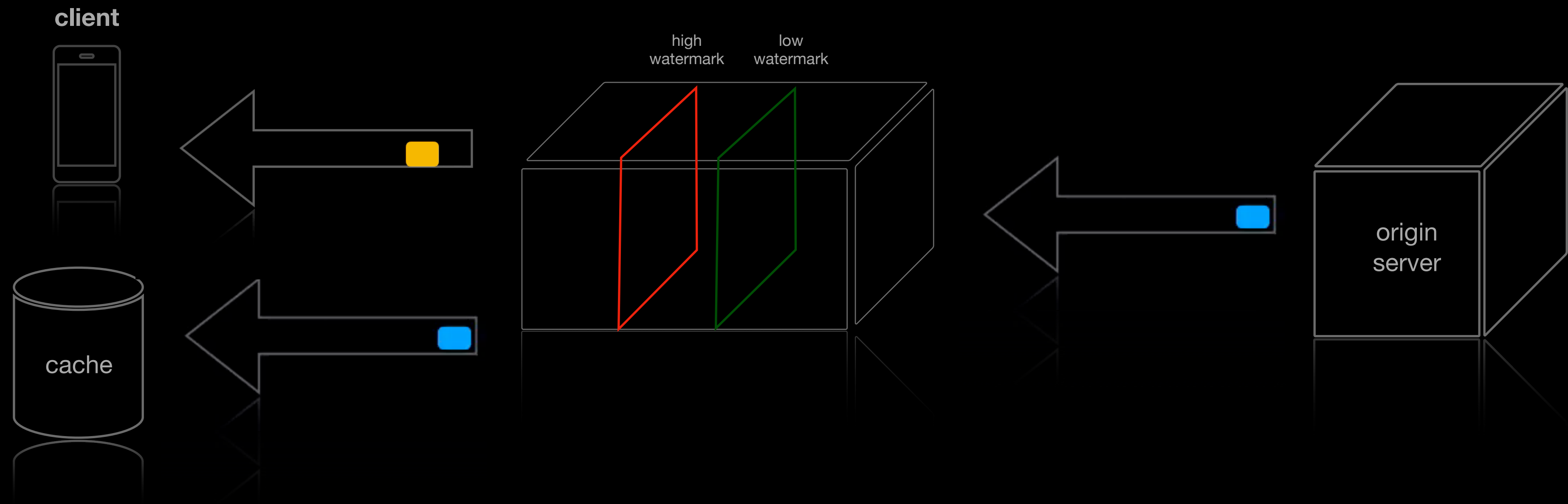


Flow Control

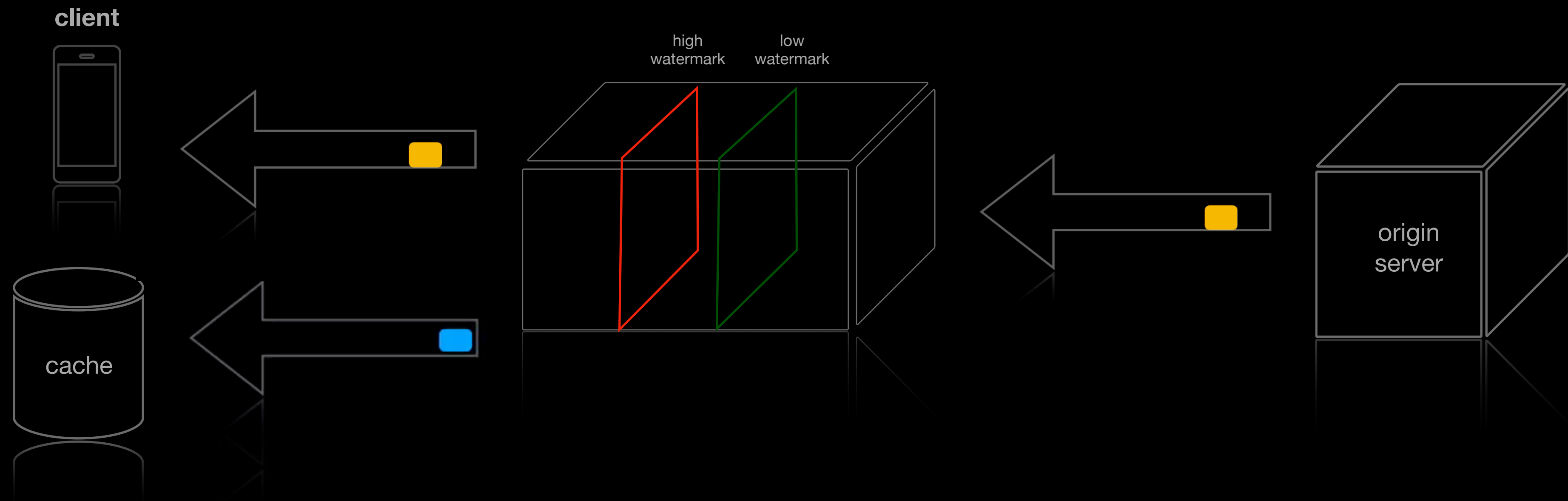
- HttpTunnel Flow Control
 - proxy-config-http-flow-control-high-water
 - proxy-config-http-flow-control-low-water
- IOBuffer Flow Control
 - proxy-config-http-default-buffer-water-mark

Both the scenarios throttle to prevent additional external data when the buffer space in use exceeds the water mark

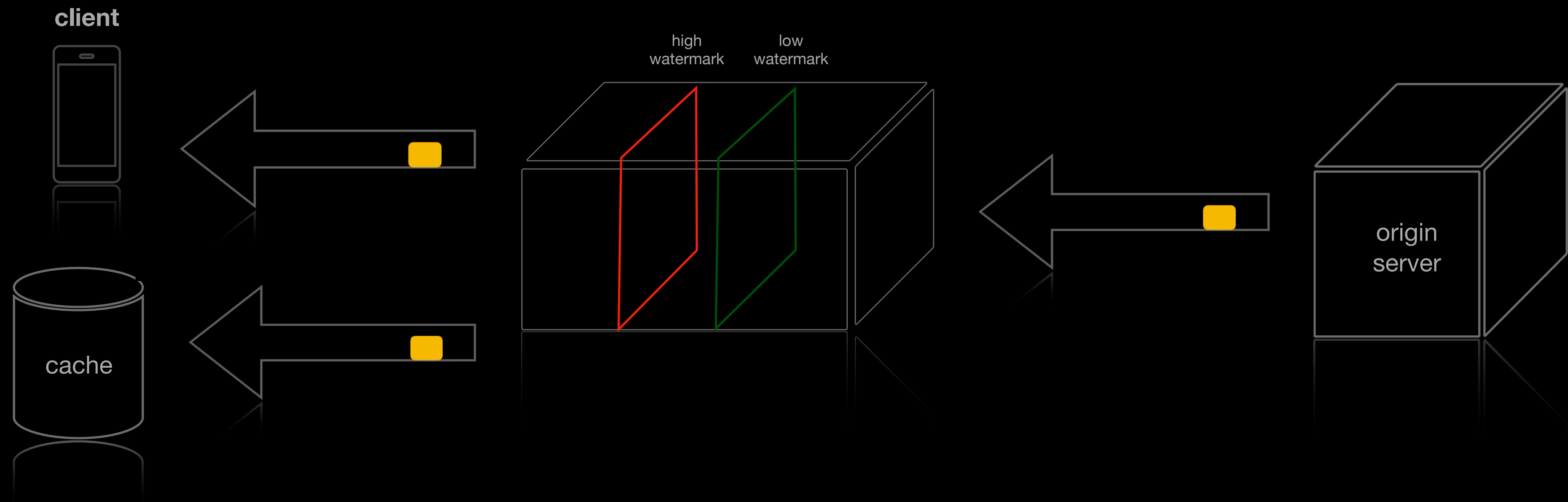
Slow Client



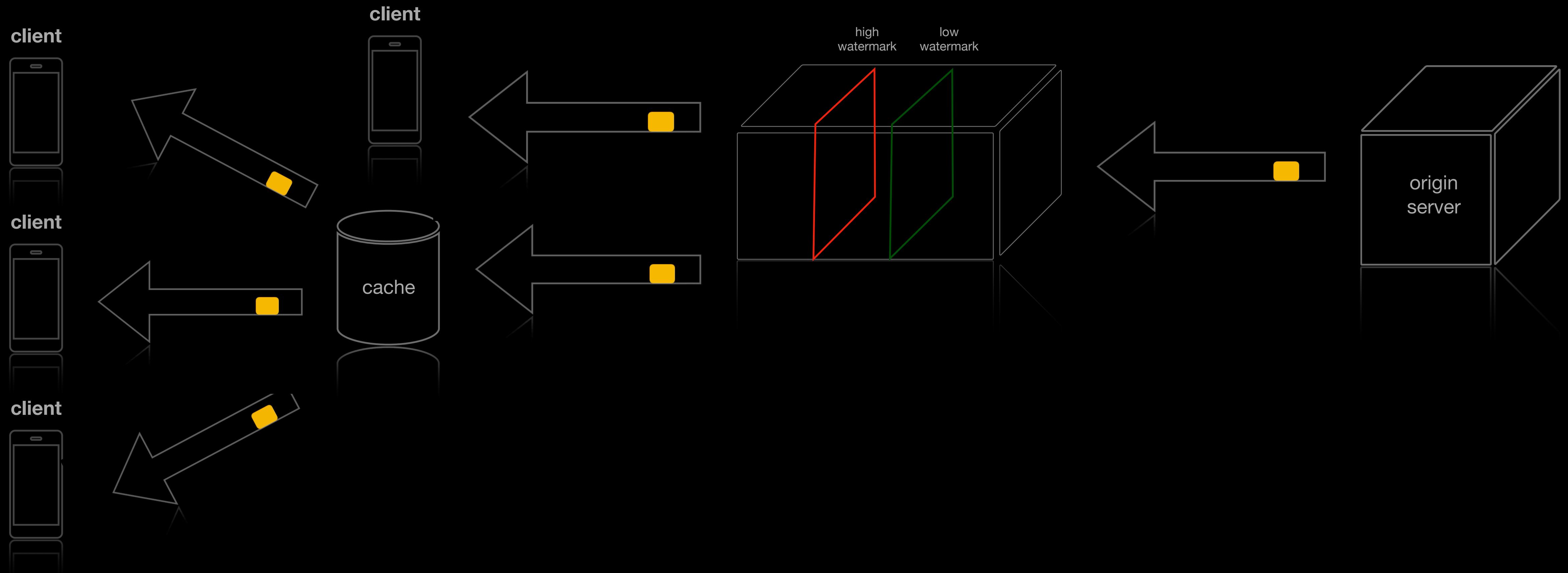
Slow Client throttles Origin Server



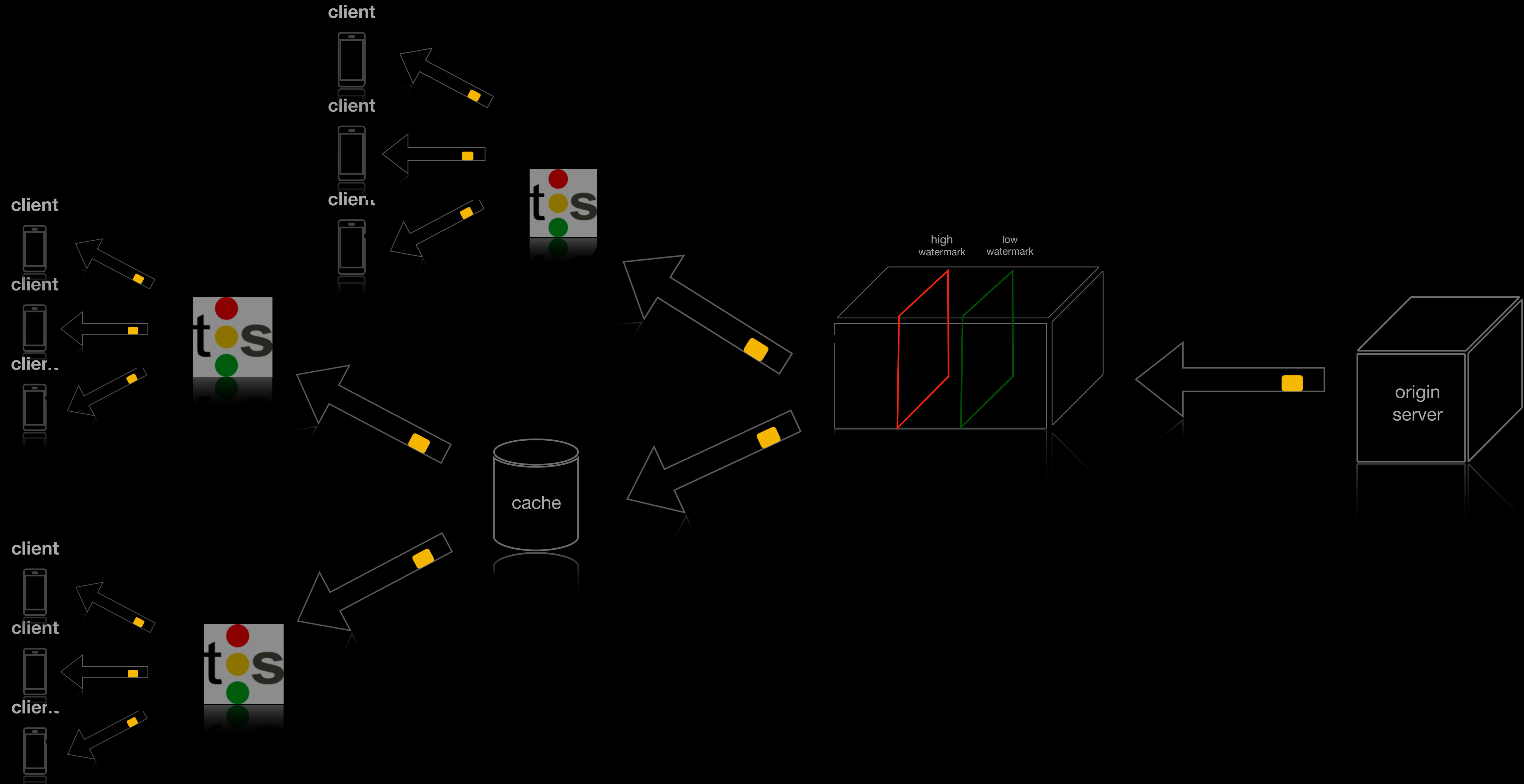
Slow Client throttling Cache



Slow Client eventually throttling RWW clients



Slow Client behavior in a cache hierarchy



Demo

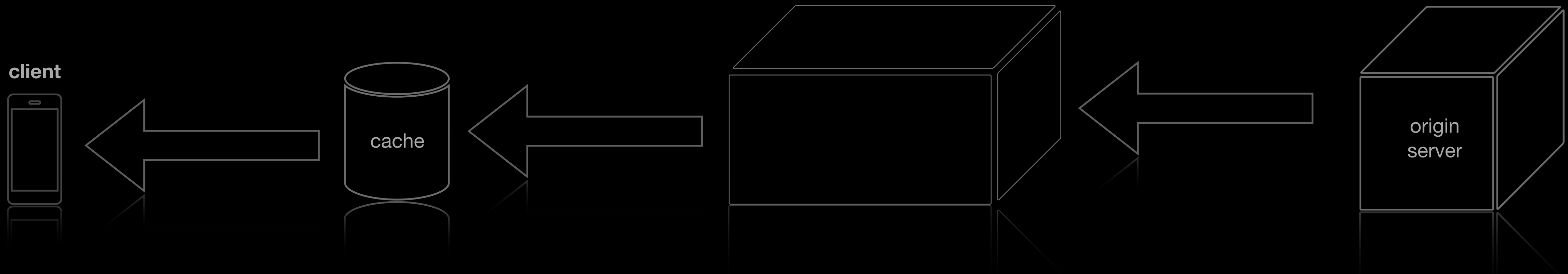
The image shows a terminal window with a dark background and light text. At the top, there is a status bar with system information: 11% battery, 25 GB memory usage, network activity (8.2 kB down, 9.2 kB up), date and time (10/24, 12:18 PM), and the shell (zsh). The terminal is split into four panes. The top-left pane is titled 'traffic_server (zsh)' and contains three lines of text: 'vmamidi@vijays-MacBook-Pro ~ %', 'vmamidi@vijays-MacBook-Pro ~ %', and 'vmamidi@vijays-MacBook-Pro ~ %'. The top-right pane is titled 'Slow Client (zsh)' and contains one line: 'vmamidi@vijays-MacBook-Pro ~ %'. The middle-right pane is titled 'Fast Client - 1 (zsh)' and contains one line: 'vmamidi@vijays-MacBook-Pro ~ %'. The bottom-right pane is titled 'Fast Client-2 (zsh)' and contains one line: 'vmamidi@vijays-MacBook-Pro ~ %'. Each pane has a close button (X) and a menu icon (three horizontal lines) on its top edge. A mouse cursor is visible at the bottom center of the terminal window.

Flow Control Tuning

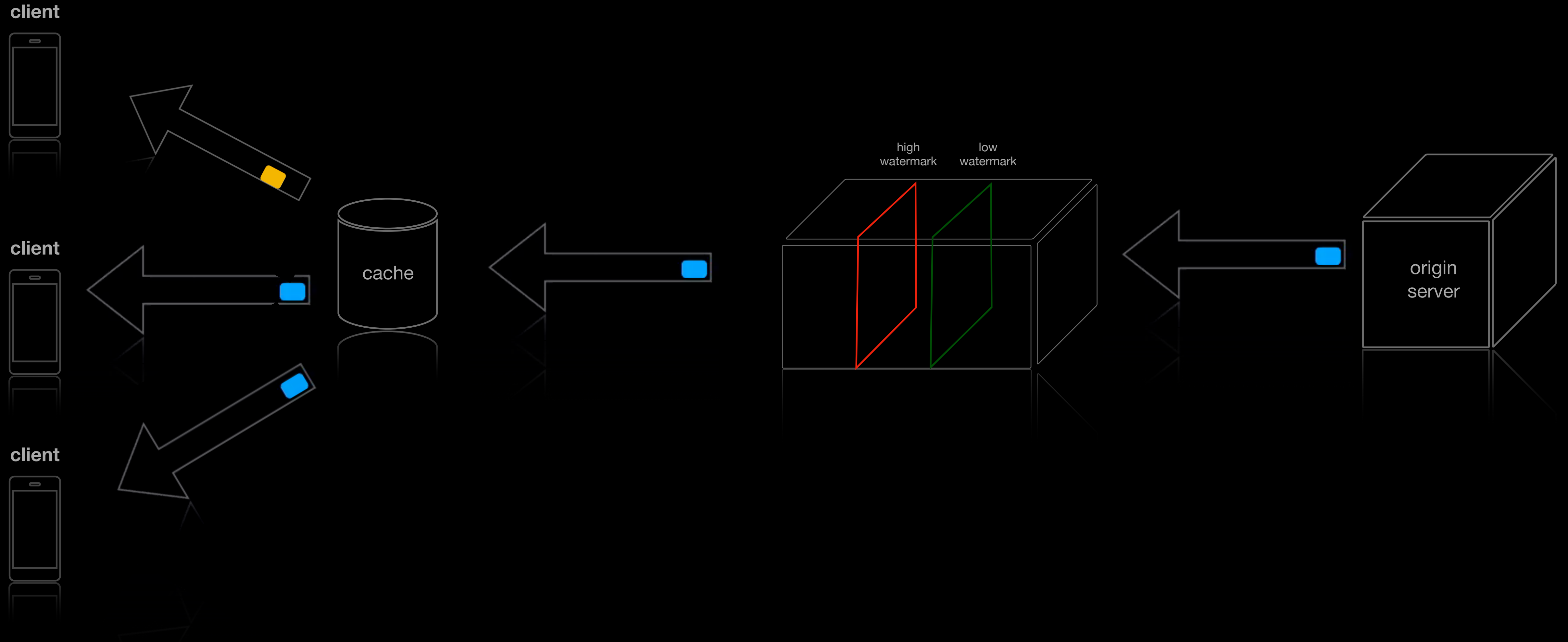
- Disable Flow Control
 - proxy-config-http-flow-control-enabled
- Tune Http Tunnel Flow Control
 - proxy-config-http-flow-control-high-water
 - proxy-config-http-flow-control-low-water
- Tune IOBuffer Flow Control
 - proxy-config-http-default-buffer-water-mark

How to figure out the right values for water-mark?

Proposed Solution



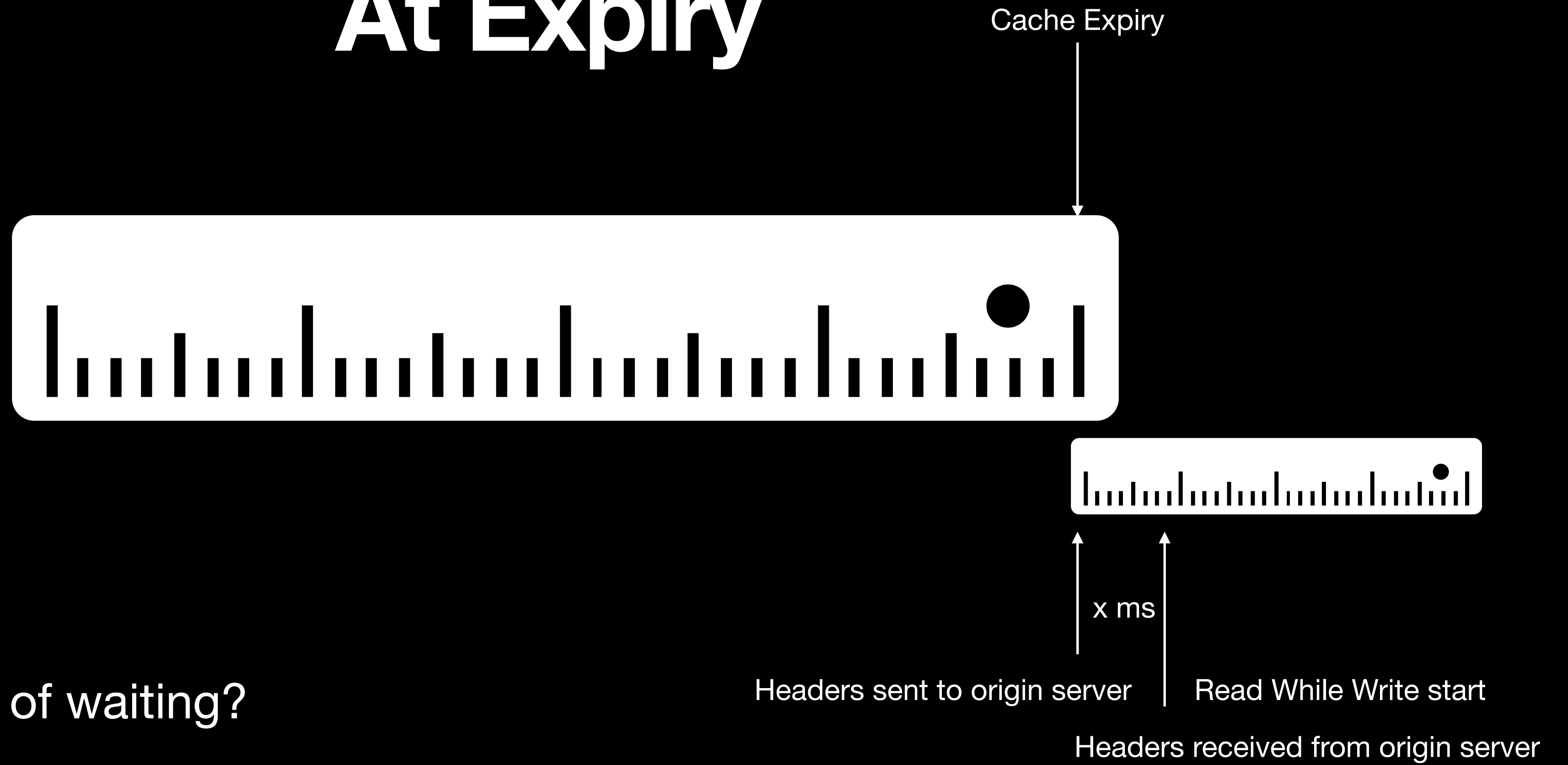
Proposed Solution



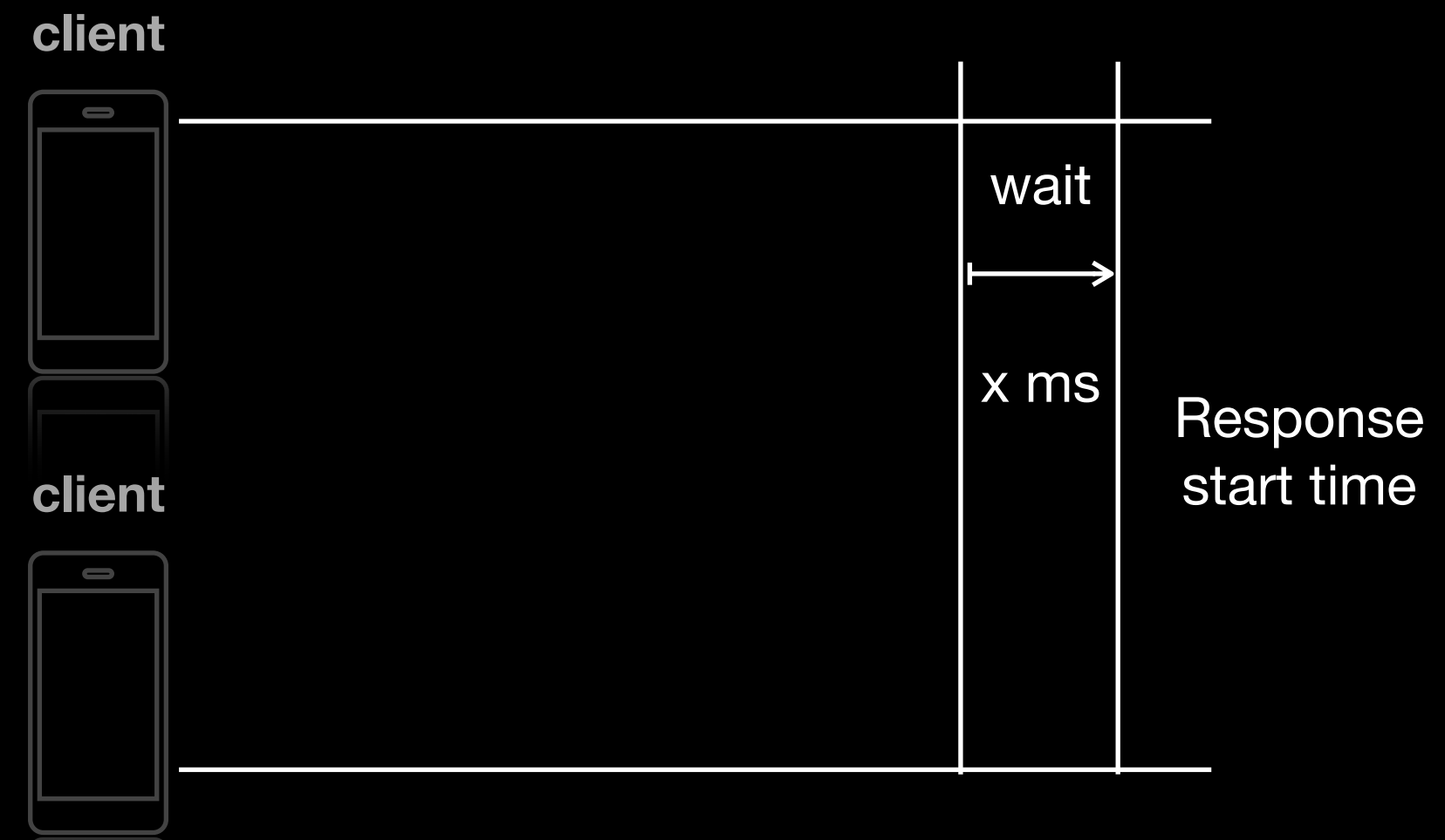
Proposed Solution

- Advantages
 - Flow control only depends on cache speed
 - Updating cache doesn't depend on client speed
 - Reduces latency of objects that are reading while object is written to cache
- Disadvantages
 - Doubles the size of buffer for the request that updates cache
 - May increase the latency of the request that is updating the cache

At Expiry



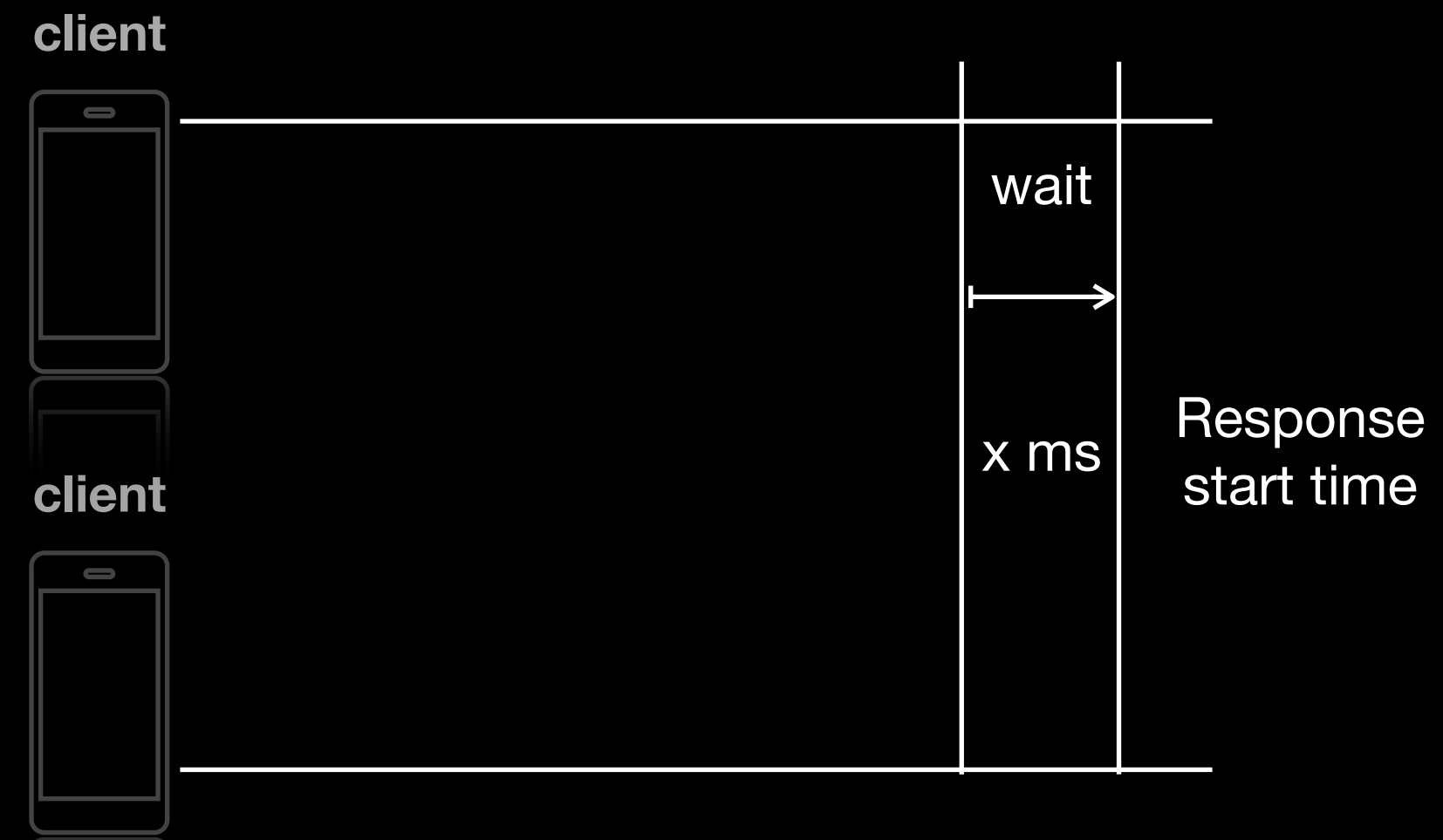
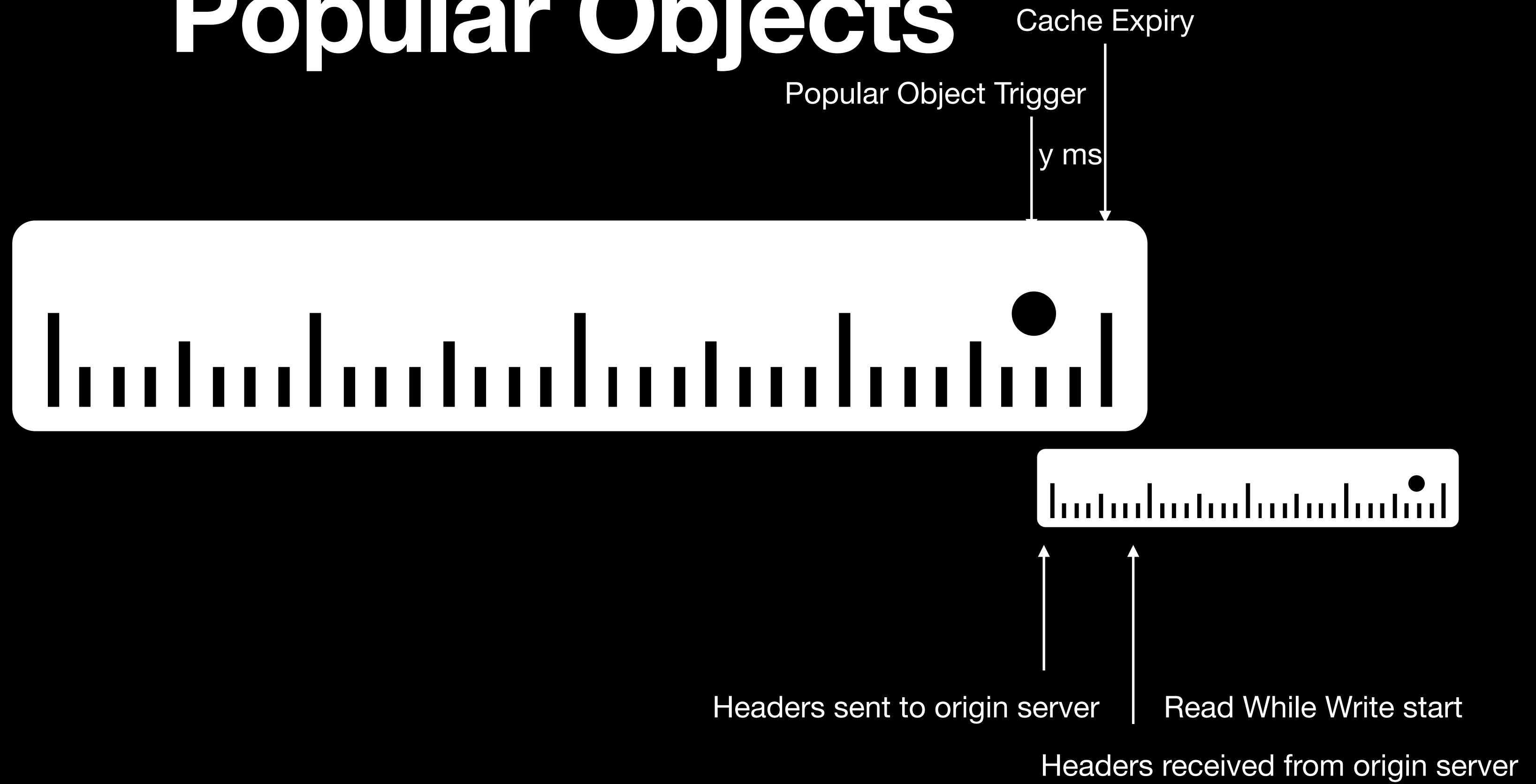
💡 Serve Stale instead of waiting?



Popular Objects

- Popular objects can be updated independent of incoming requests
 - Requires a plug-in to initiate pre-warming of the popular objects
 - Requires an algorithm to find the popular objects

Popular Objects



Popular Objects

- Cache refresh causes cache reads to wait until the headers are updated
- This introduces latency for the requests for the object at the time of updation
- How about atomically updating cache?

Discussion

- How do we find other scenarios
 - Increase Debug performance such that it can be enabled with less performance overhead
 - SystemTap