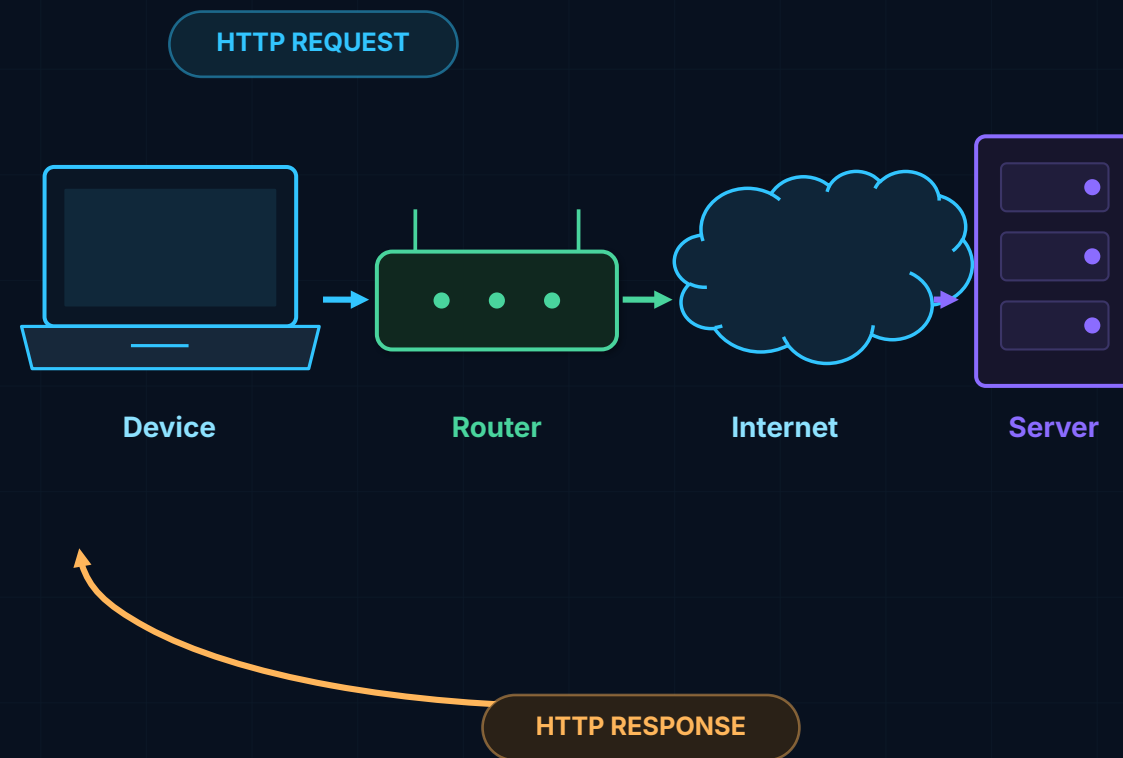


Lifecycle of a Web Request

From pressing Enter to pixels on your screen—and every DNS lookup, packet hop, handshake, server call, and browser task in between.

REQUEST → INTERNET → RESPONSE

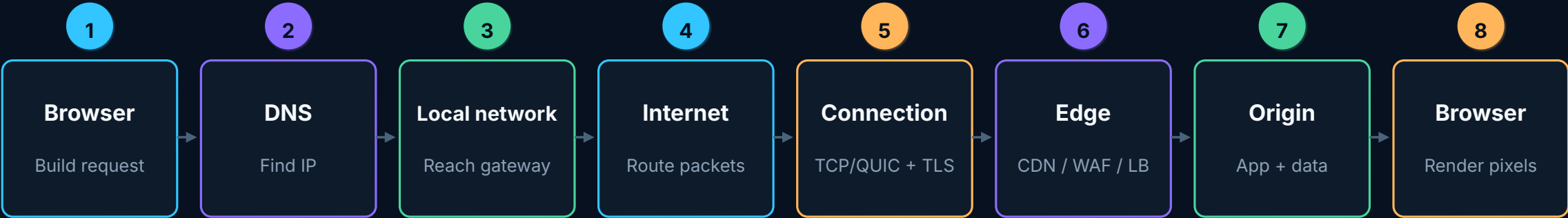


Core idea

HTTP is the message. DNS finds an address. IP routes packets. Transport delivers bytes. TLS protects them.

One click, many systems

A single page load crosses multiple layers—and usually triggers many more requests.



The same data is wrapped at each layer

Endpoints understand the full HTTP message. Routers in the middle mostly forward IP packets.

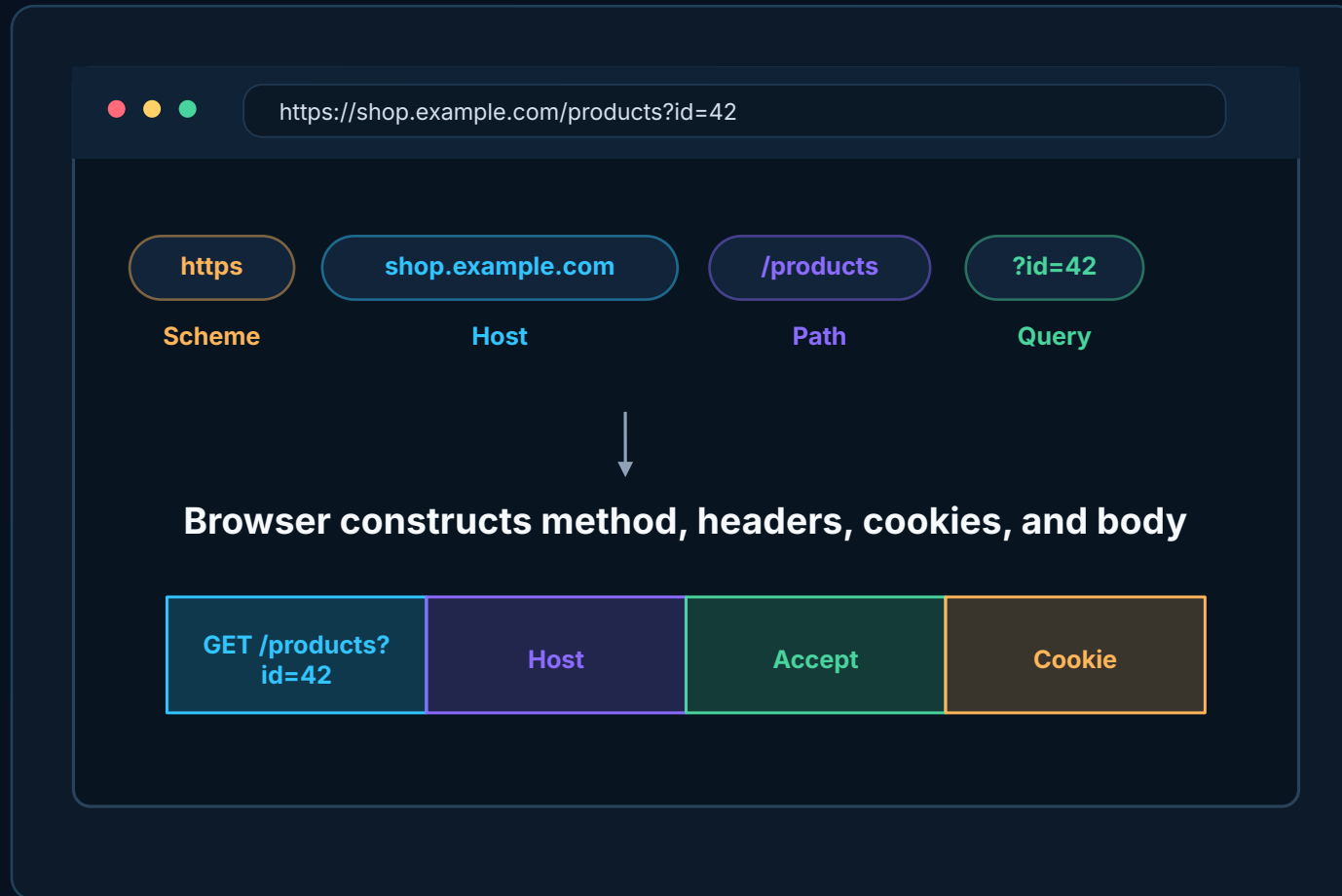


A PAGE IS A REQUEST GRAPH

HTML often triggers CSS, JavaScript, images, fonts, API calls, analytics, and more.

1. The browser prepares the request

Before any packet leaves, the browser parses the URL and checks whether it already has the answer.



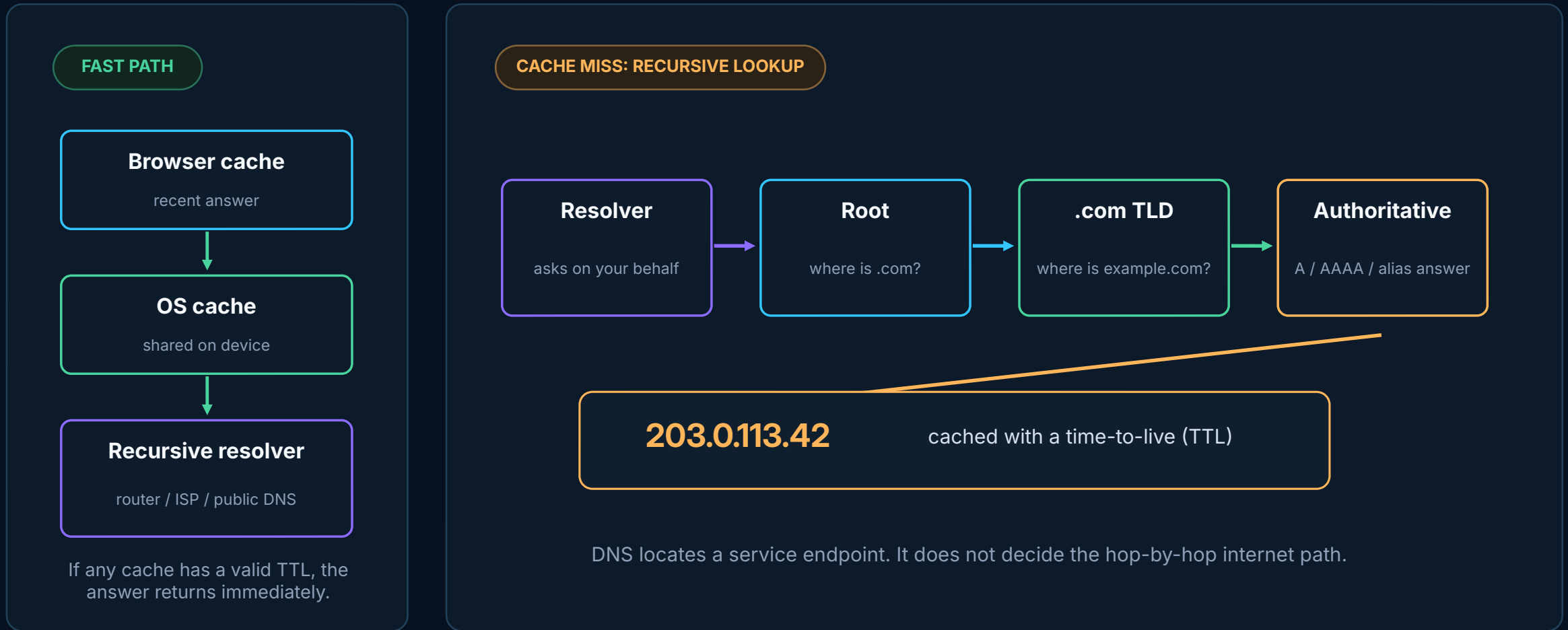
CHECKS BEFORE NETWORK

- Memory or disk cache
- Service worker interception
- HTTPS and HSTS rules
- Existing DNS / connection state
- Content security and cookie policy

A cache hit can end the journey here.

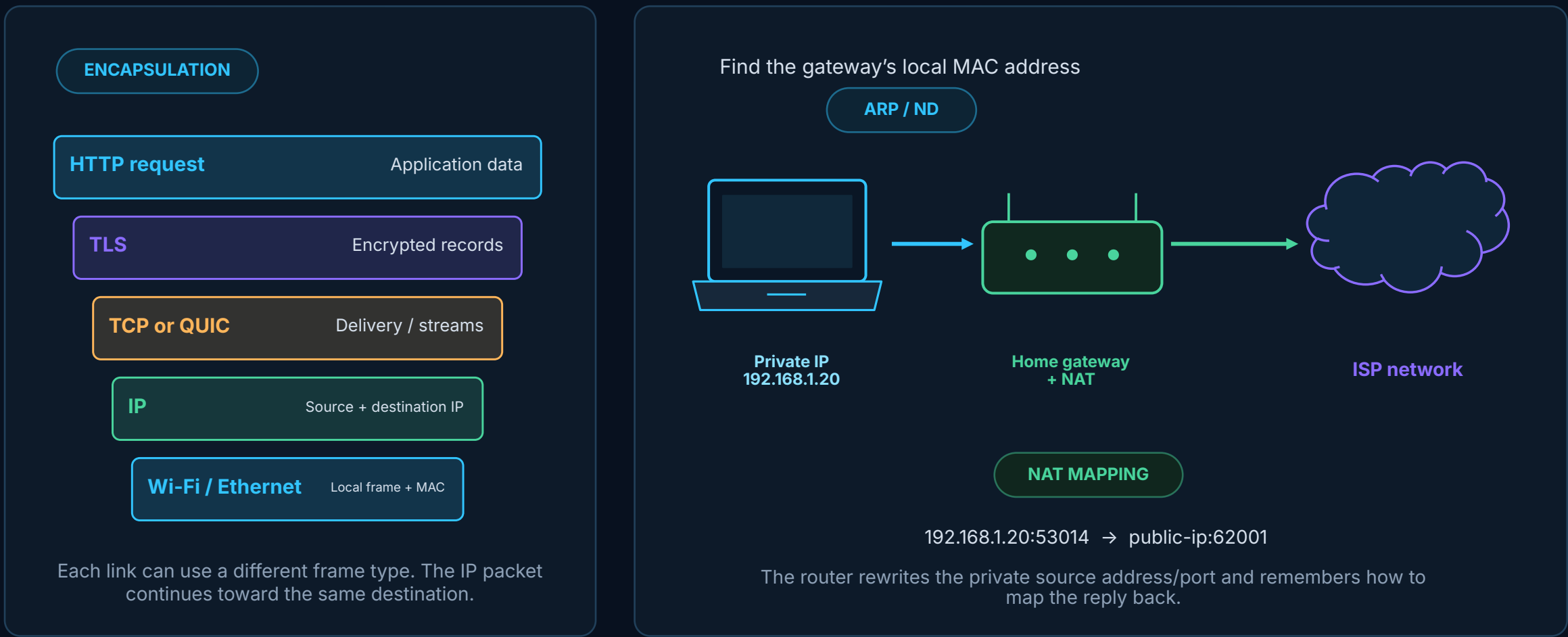
2. DNS turns a name into an IP address

The browser needs a routable address for the host before it can connect.



3. Your device reaches the local gateway

The operating system wraps data in layers and sends a frame over Wi-Fi or Ethernet.



4. Routers move packets across the internet

Every router chooses a next hop; no single machine plans the entire journey.



Destination prefix: 203.0.113.0/24

Each hop consults a routing table and forwards the packet to a neighboring router.

Routing table

Longest-prefix match chooses the next hop.

BGP

Networks advertise which IP prefixes they can reach.

Hop limit

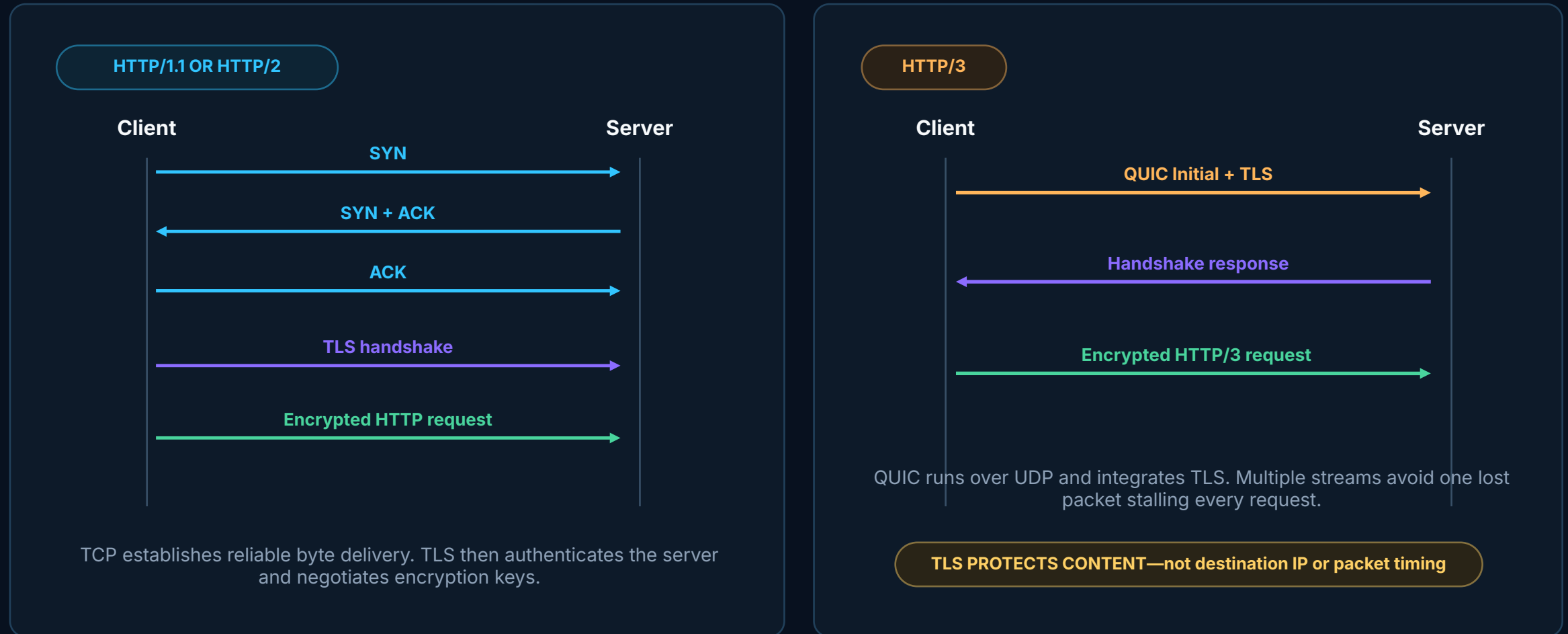
TTL / Hop Limit prevents packets looping forever.

Asymmetry

The response may take a different route back.

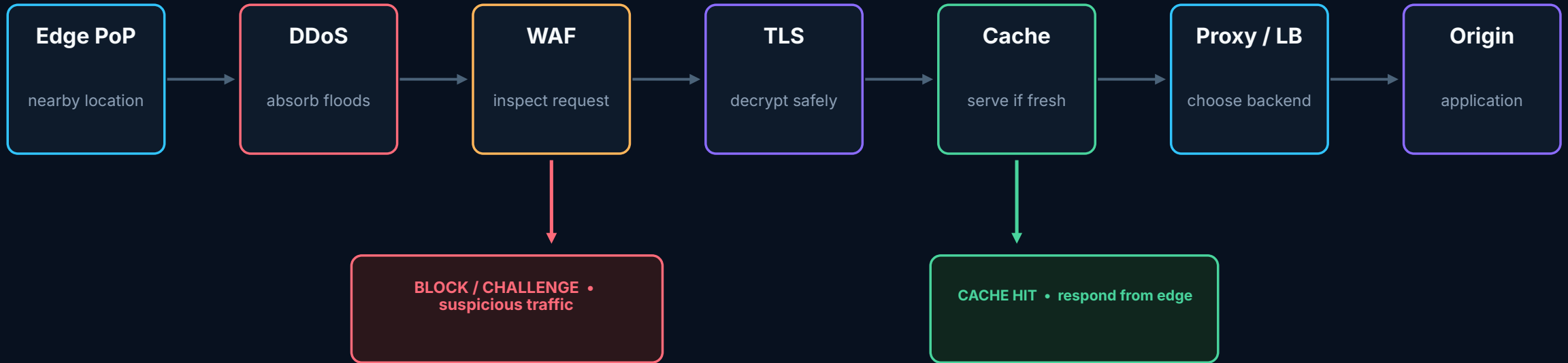
5. The client establishes a secure connection

The exact handshake depends on whether the site uses HTTP/1.1, HTTP/2, or HTTP/3.



6. Edge infrastructure receives the request

Many sites terminate the connection at a nearby edge before forwarding anything to the origin.



Nearer edge

reduces round-trip time

Shared protection

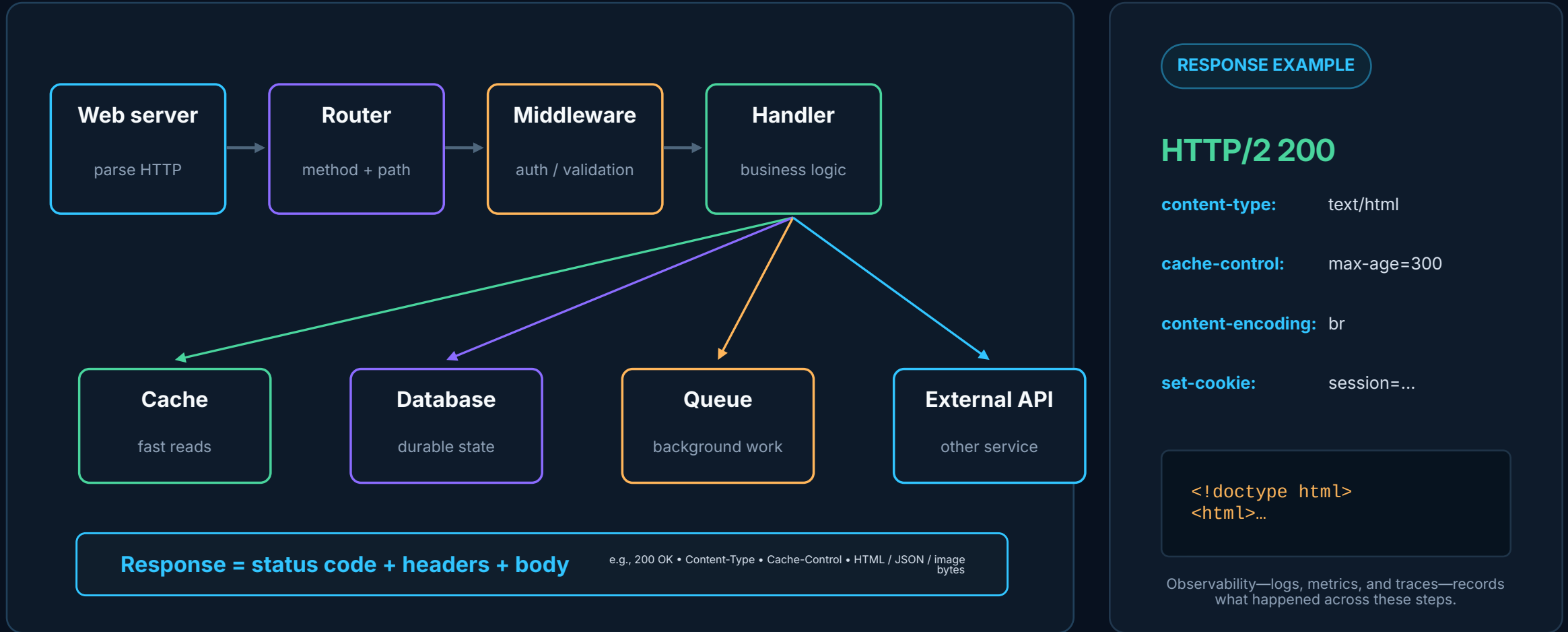
filters abuse before origin

Load balancing

spreads traffic across healthy servers

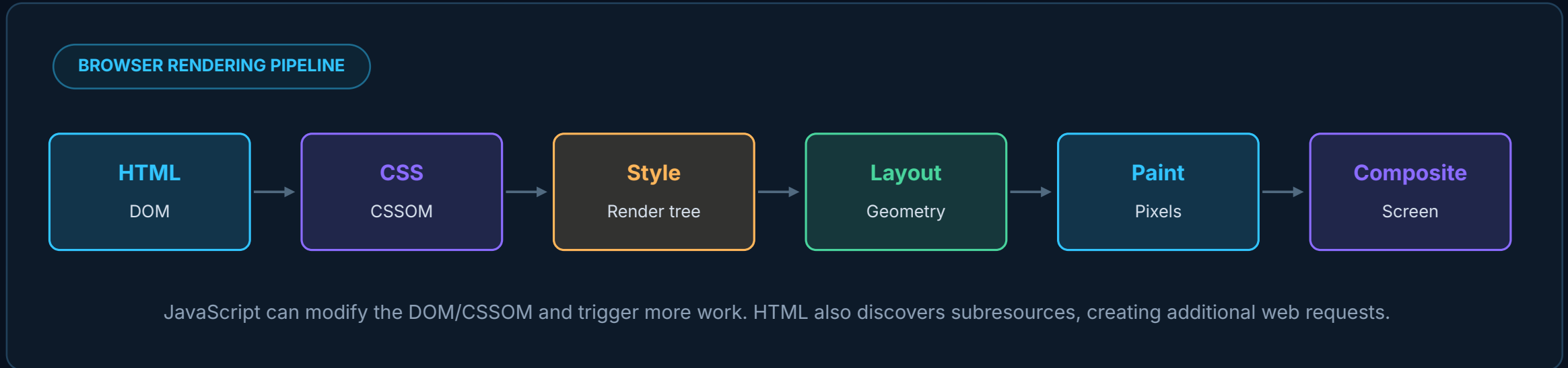
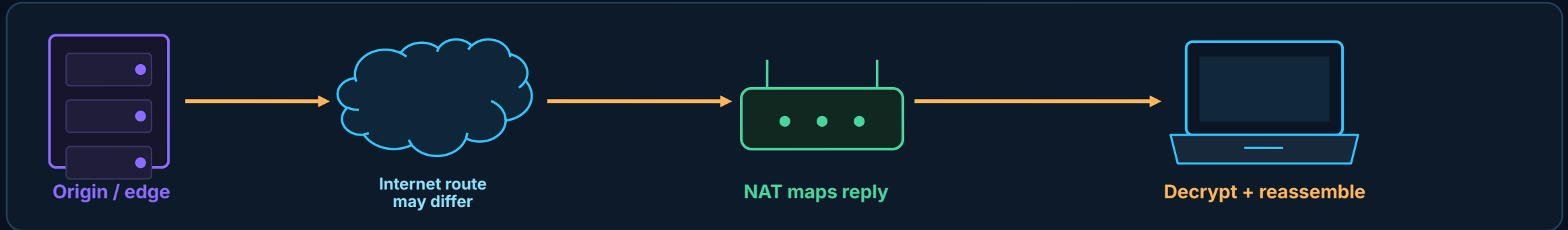
7. The server creates the response

The request is routed through application logic, caches, databases, queues, and external services.



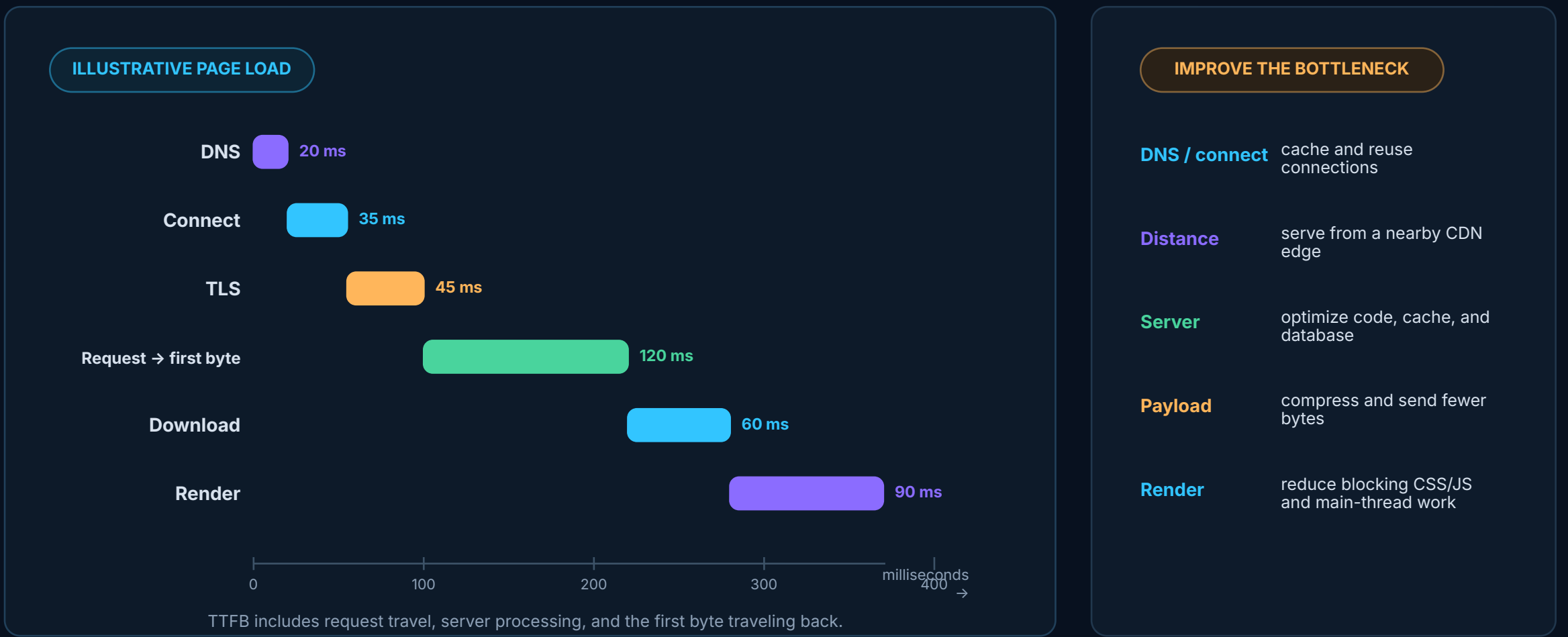
8. The response returns—and the browser renders it

Packets travel back, are decrypted and reassembled, then become DOM, layout, paint, and pixels.



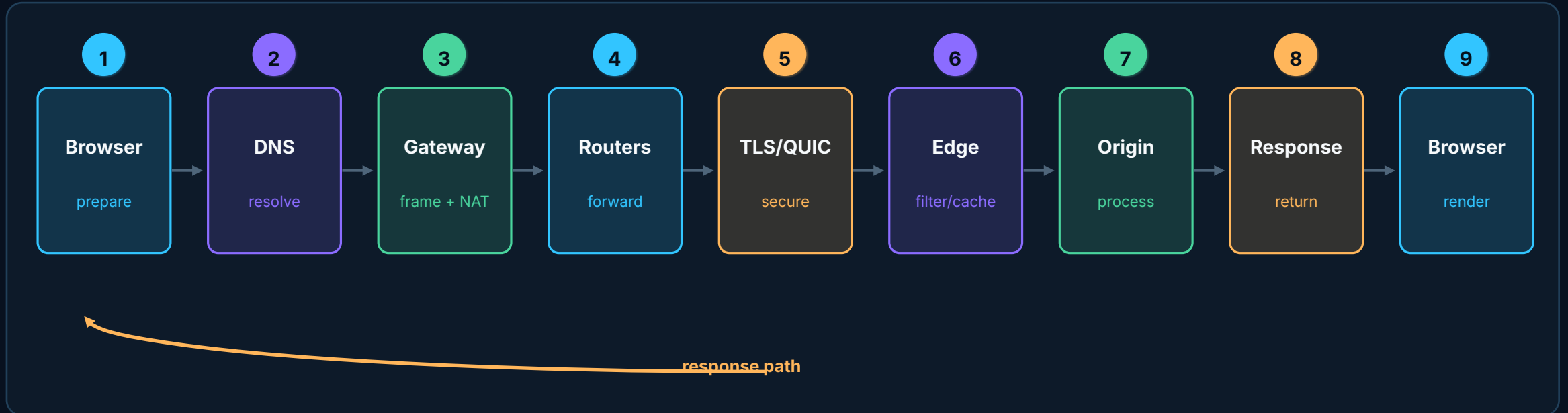
Where does the time go?

An illustrative waterfall separates network setup, server wait, transfer, and browser work.



End-to-end recap

A web request is a layered round trip—not a direct conversation between a browser and one server.



Names → addresses

DNS gives an endpoint; routing moves packets toward it.

Layers have different jobs

HTTP describes meaning, TLS protects, transport delivers, IP routes.

One page means many requests

The first HTML response starts a graph of dependent resources.