

CONCEPTOS CLAVE DISEÑO SISTEMAS A GRAN ESCALA

---

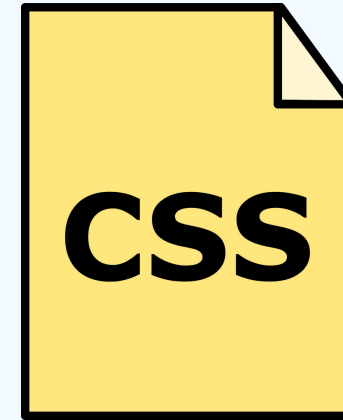
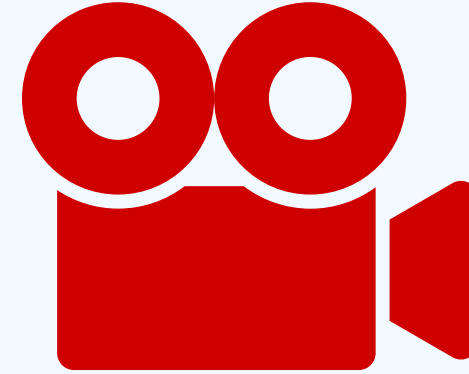
CDN

---

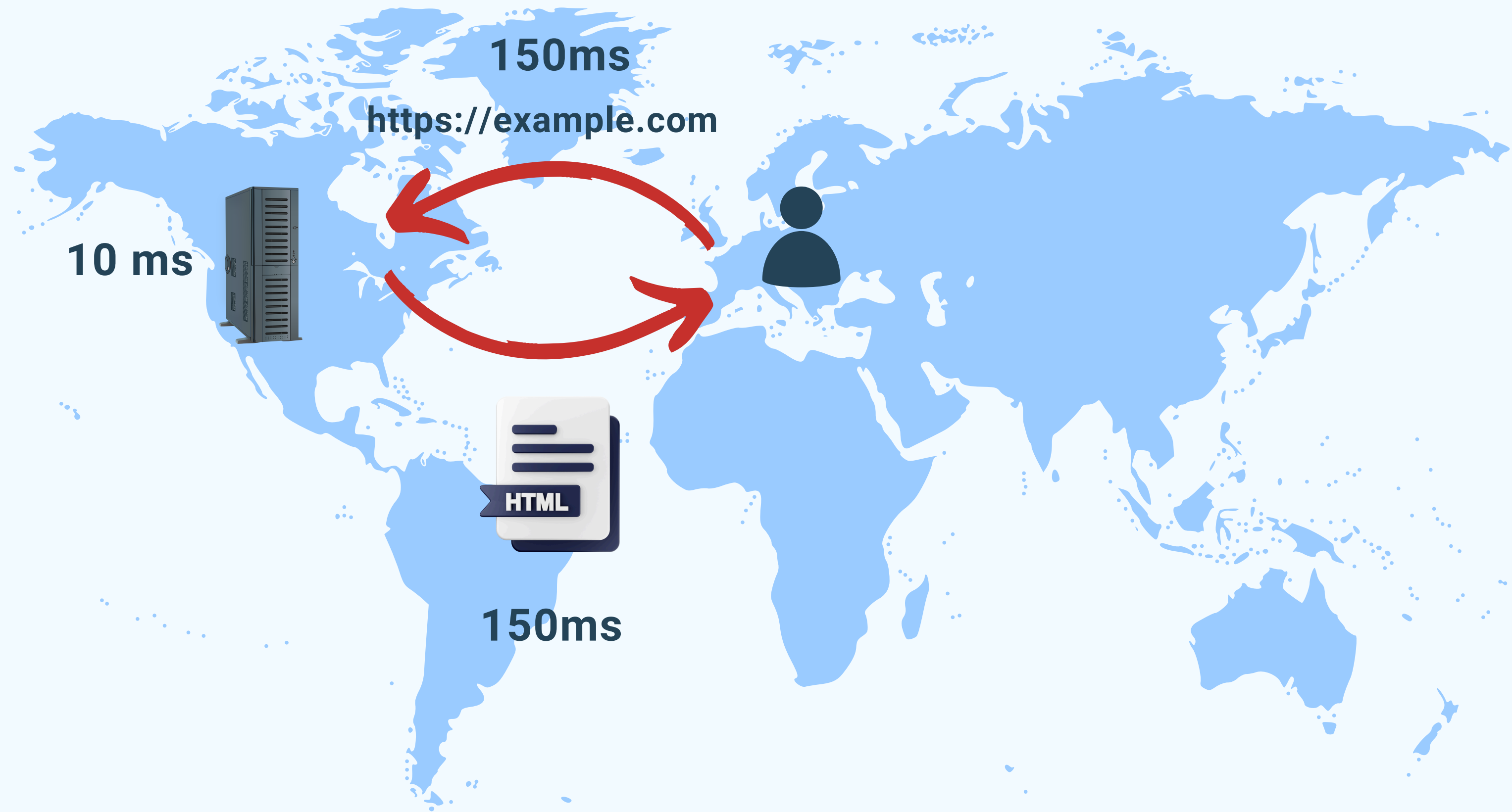
Daniel Blanco Calviño

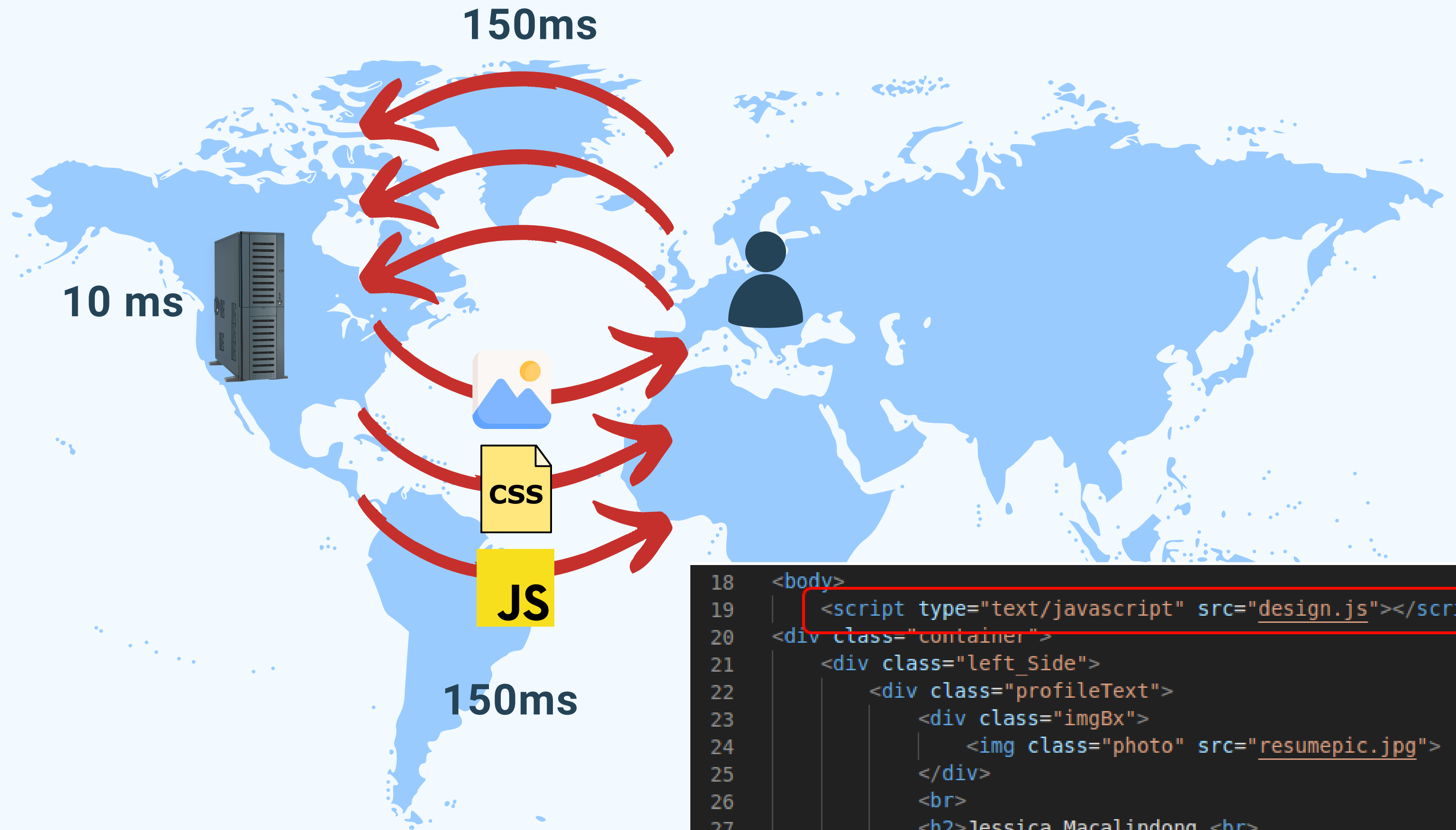
# RECURSOS ESTÁTICOS

- Imágenes.
- Vídeos.
- CSS y JavaScript.

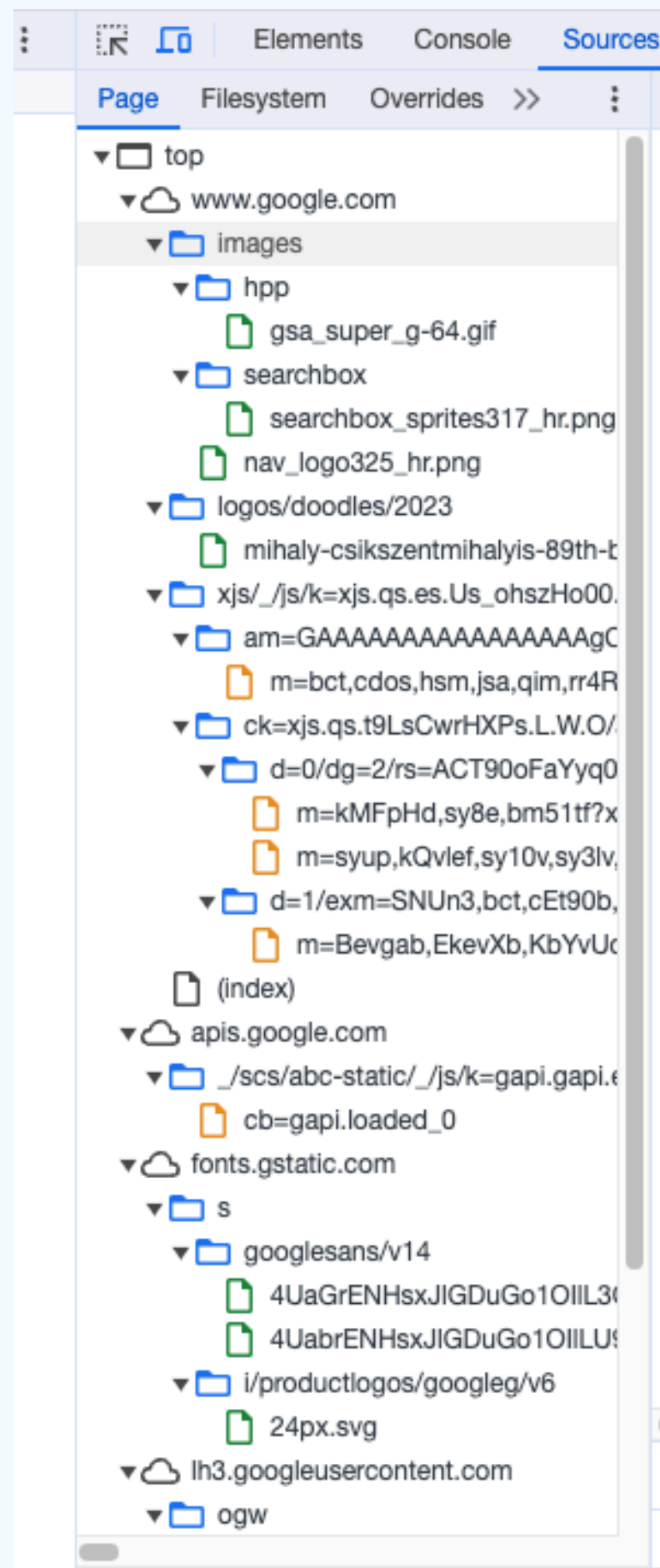


Si estos recursos se cargan **lentamente**, el usuario tendrá una **muy mala experiencia**. No importa que la primera petición y procesamiento sean rápidos.





```
18 <body>
19 |   <script type="text/javascript" src="design.js"></script>
20 <div class="container">
21 |   <div class="left_Side">
22 |     <div class="profileText">
23 |       <div class="imgBx">
24 |         
25 |       </div>
26 |       <br>
27 |       <h2>Jessica Macalindong <br>
28 |     </div>
```

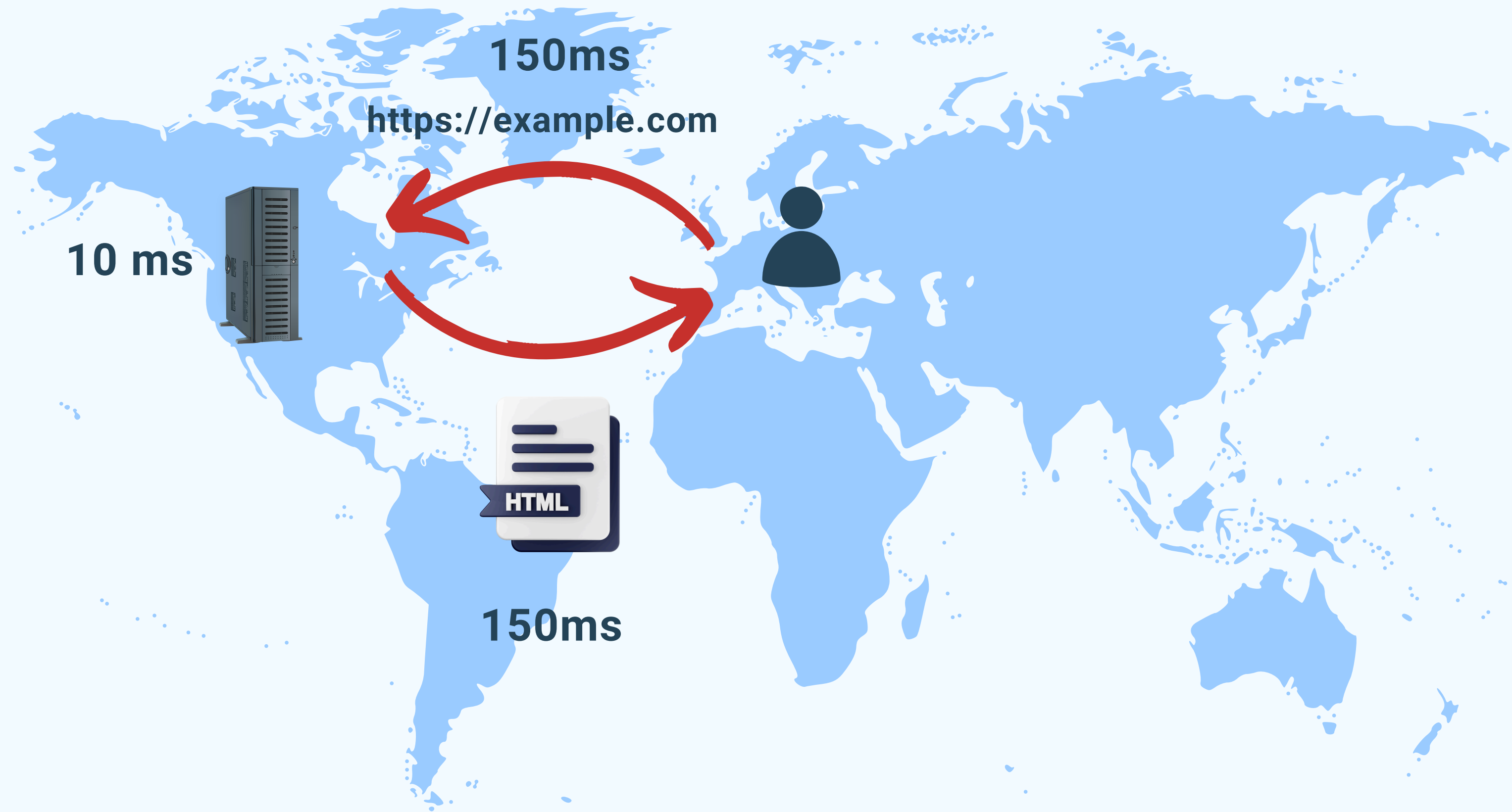


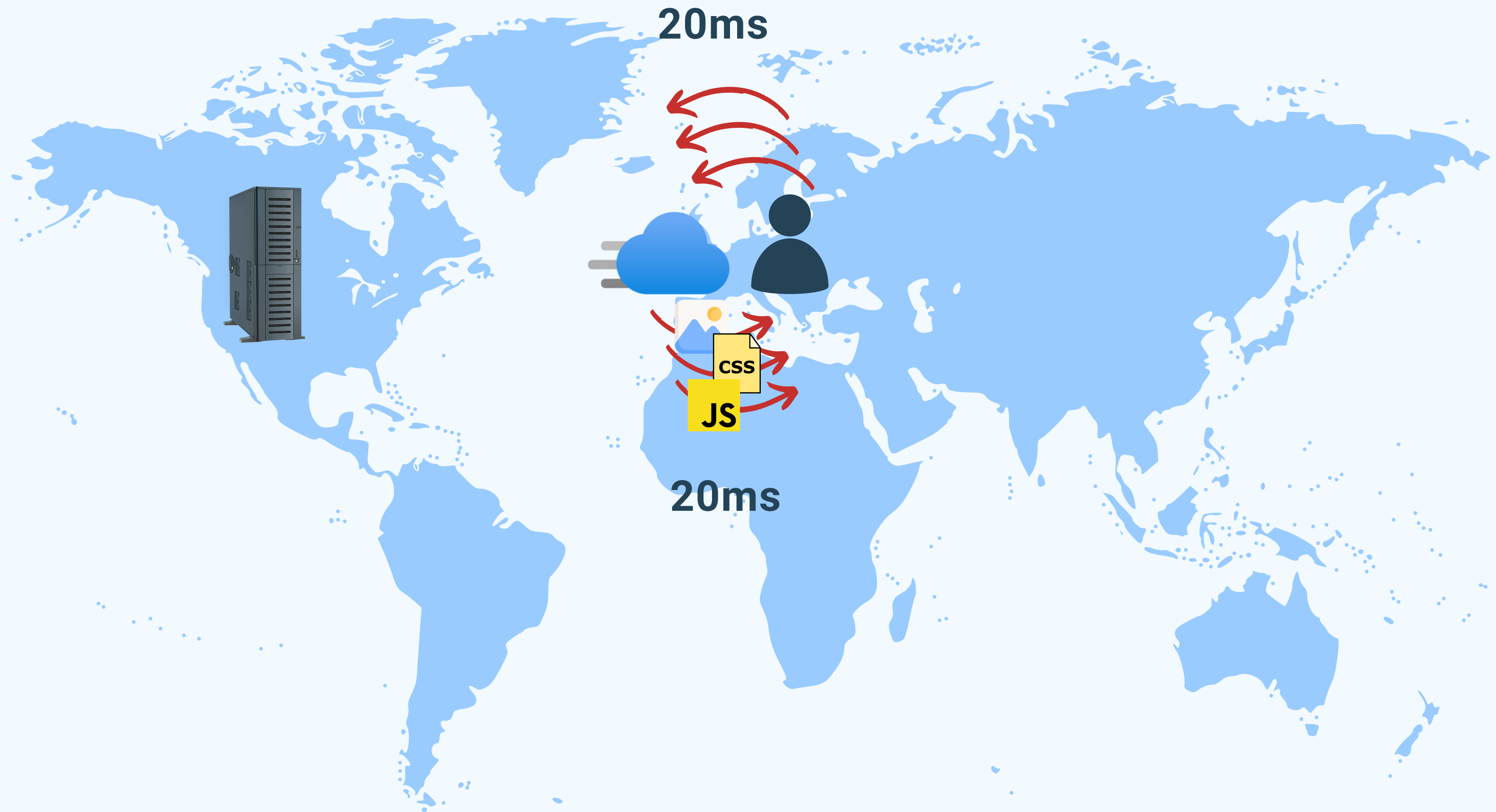
# CDN

- Infraestructura distribuida compuesta por servidores en **diversas ubicaciones**.
  - Diseñados para **almacenar y entregar recursos de manera rápida y eficiente**.
  - Optimizan la experiencia del usuario, **reduciendo la latencia**.
  - Mejoran la **disponibilidad** del contenido.
  - Optimizado para cachear recursos. Algoritmos de **compresión** (gzip) y **minificación** de ficheros JS y CSS.



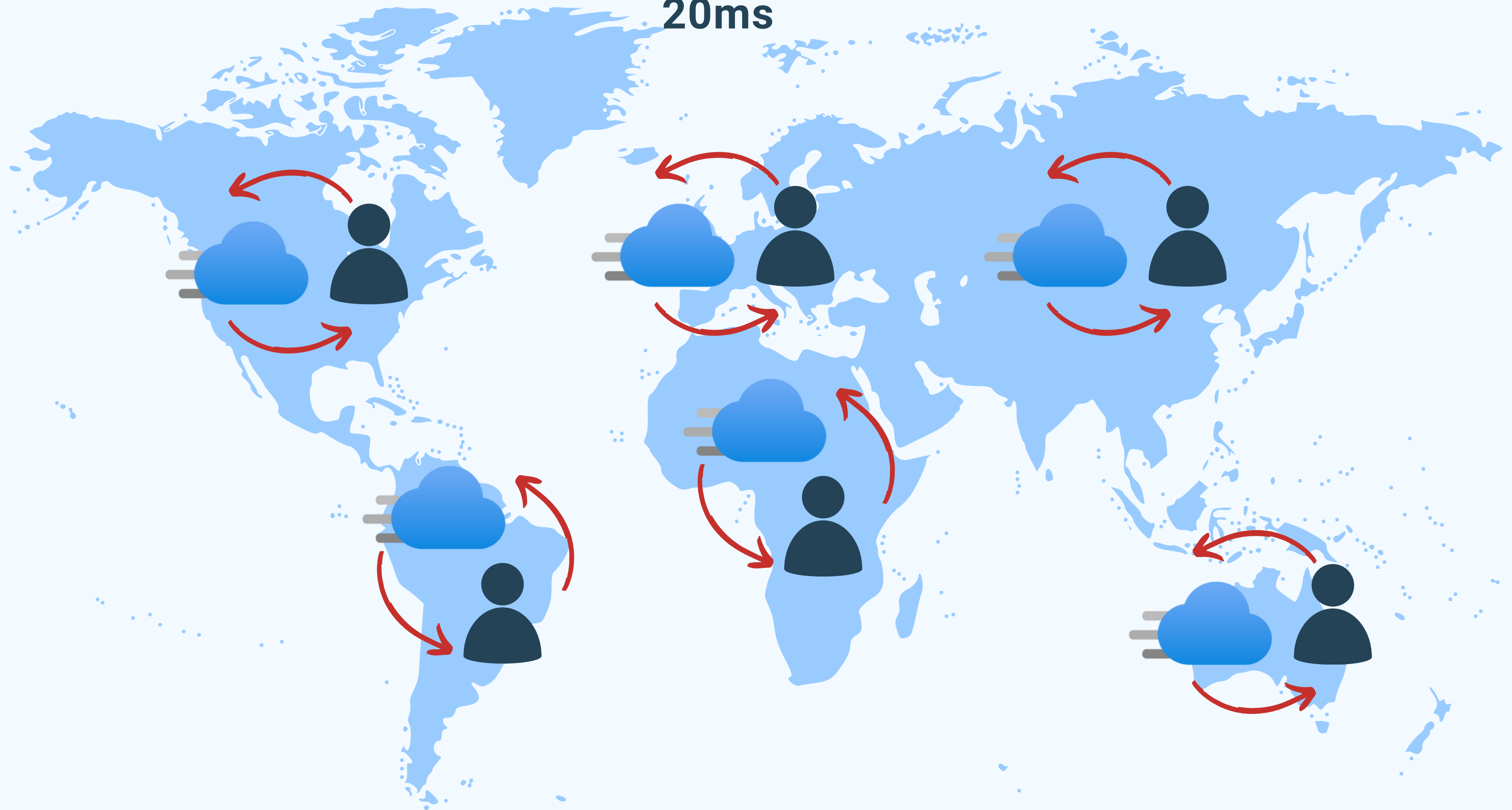
Son cachés para recursos estáticos.



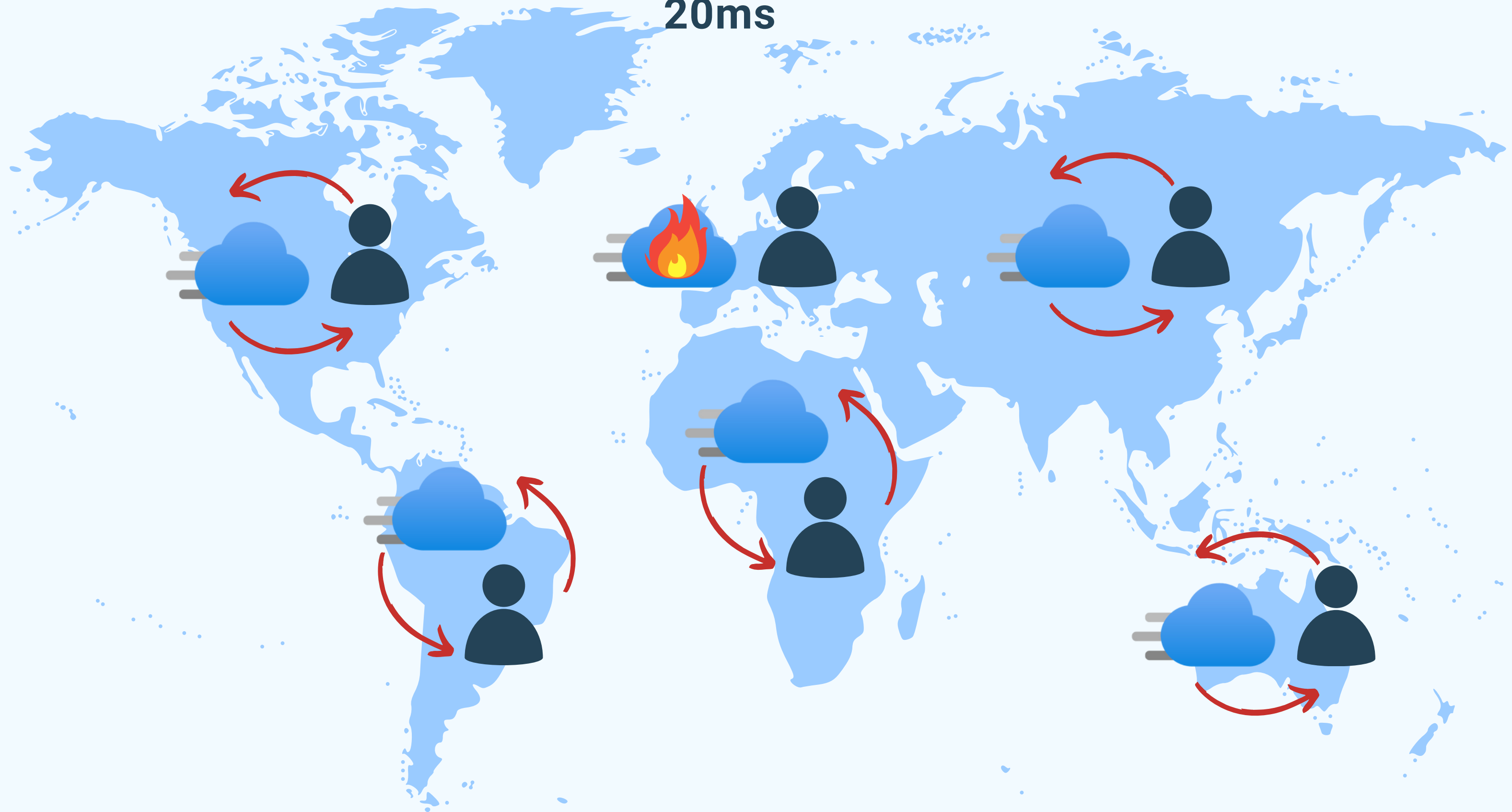




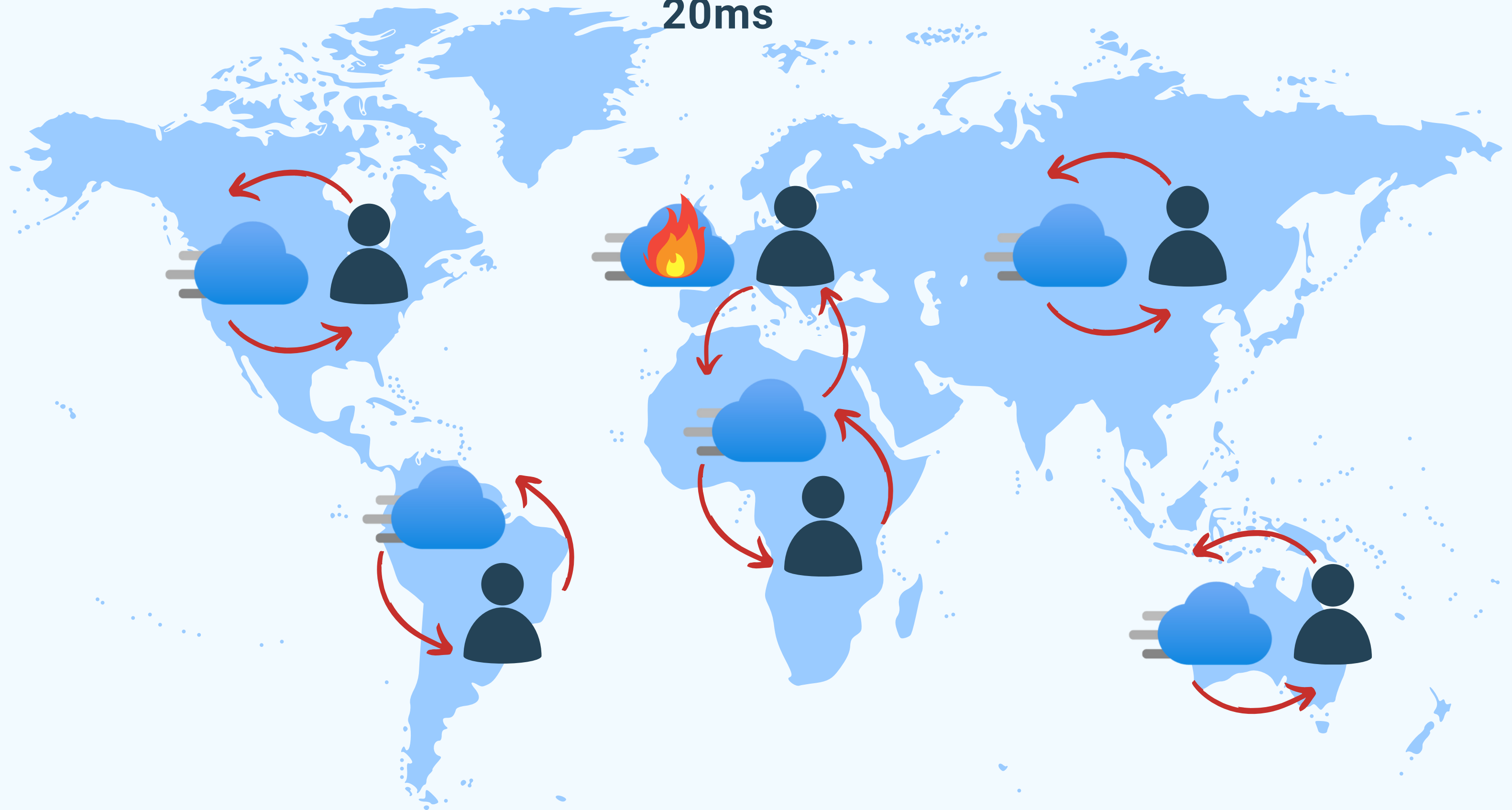
20ms



20ms

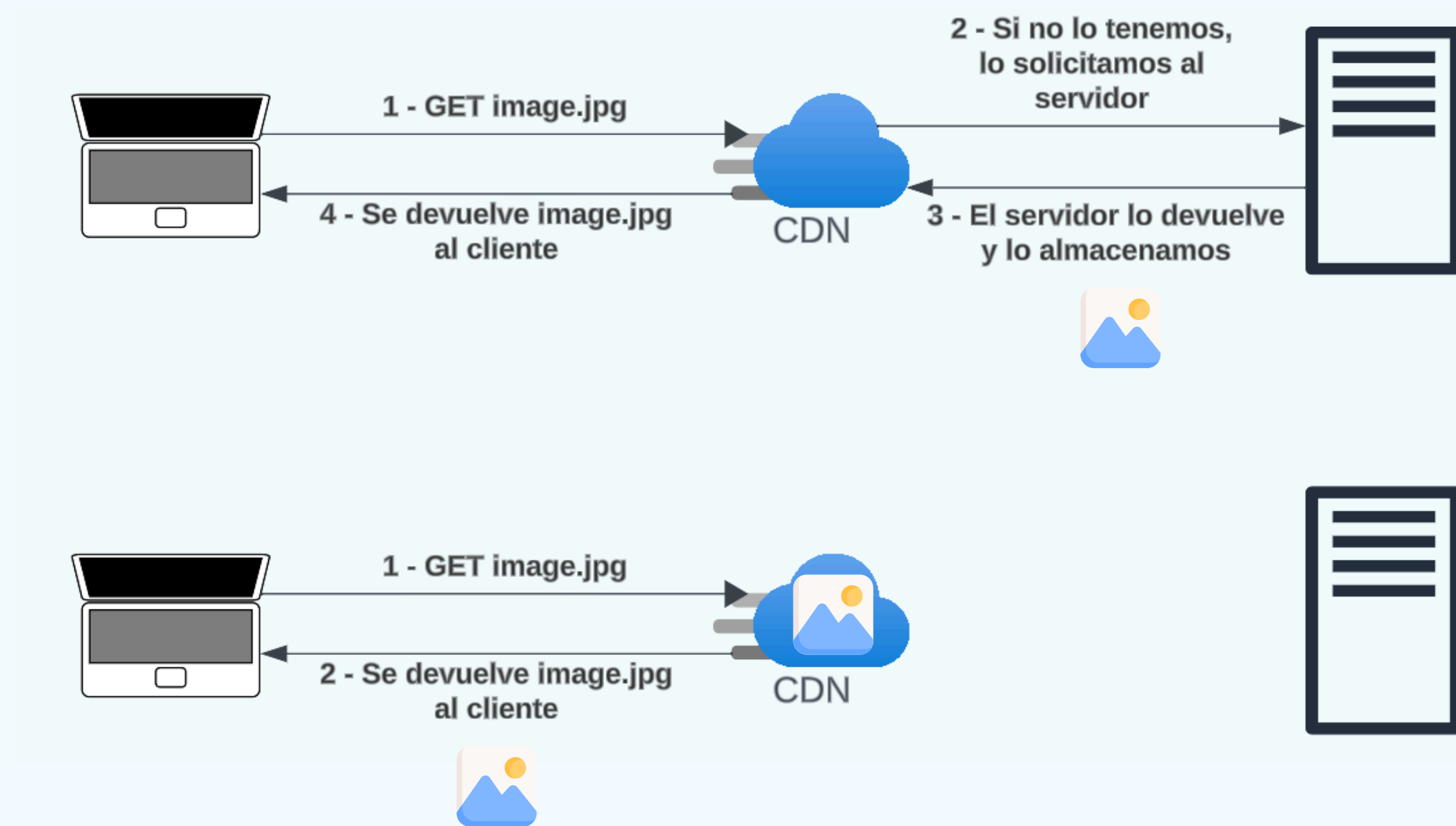


20ms



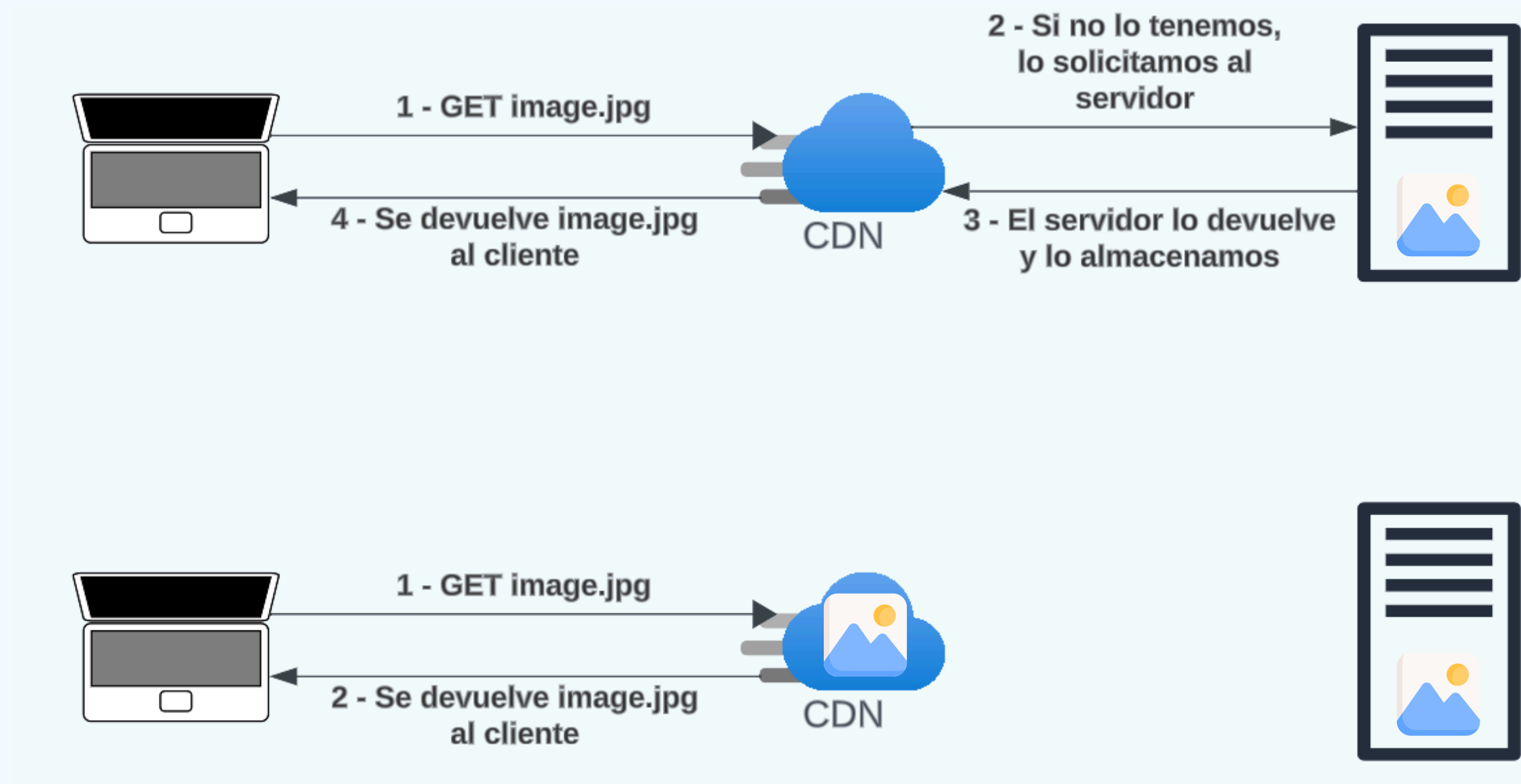
# FUNCIONAMIENTO CDN

1. El cliente solicita un recurso al CDN.
2. Si el CDN no tiene el recurso, lo solicita al servidor y lo almacenamos.
3. Se devuelve el recurso al cliente de forma transparente.



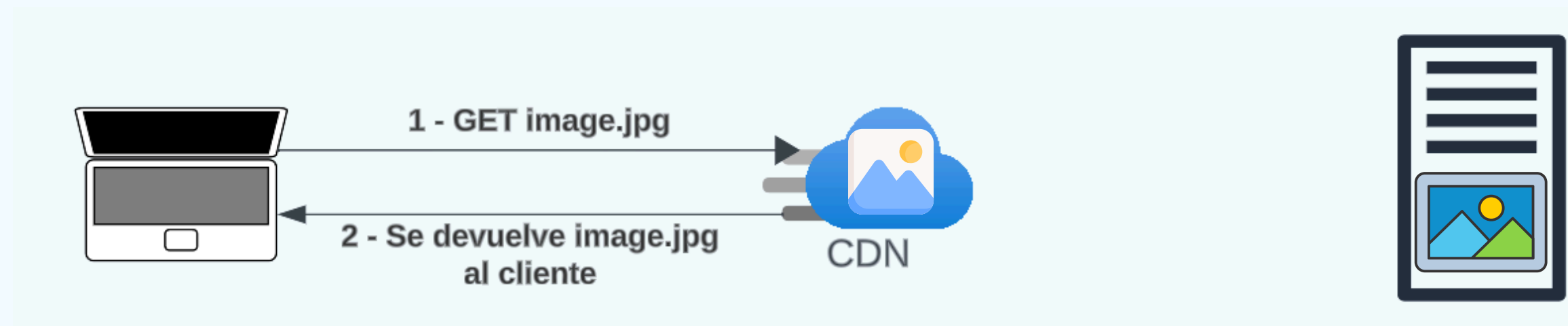
# ESTRATEGIA PULL

- El CDN solicita **bajo demanda** los recursos al servidor.

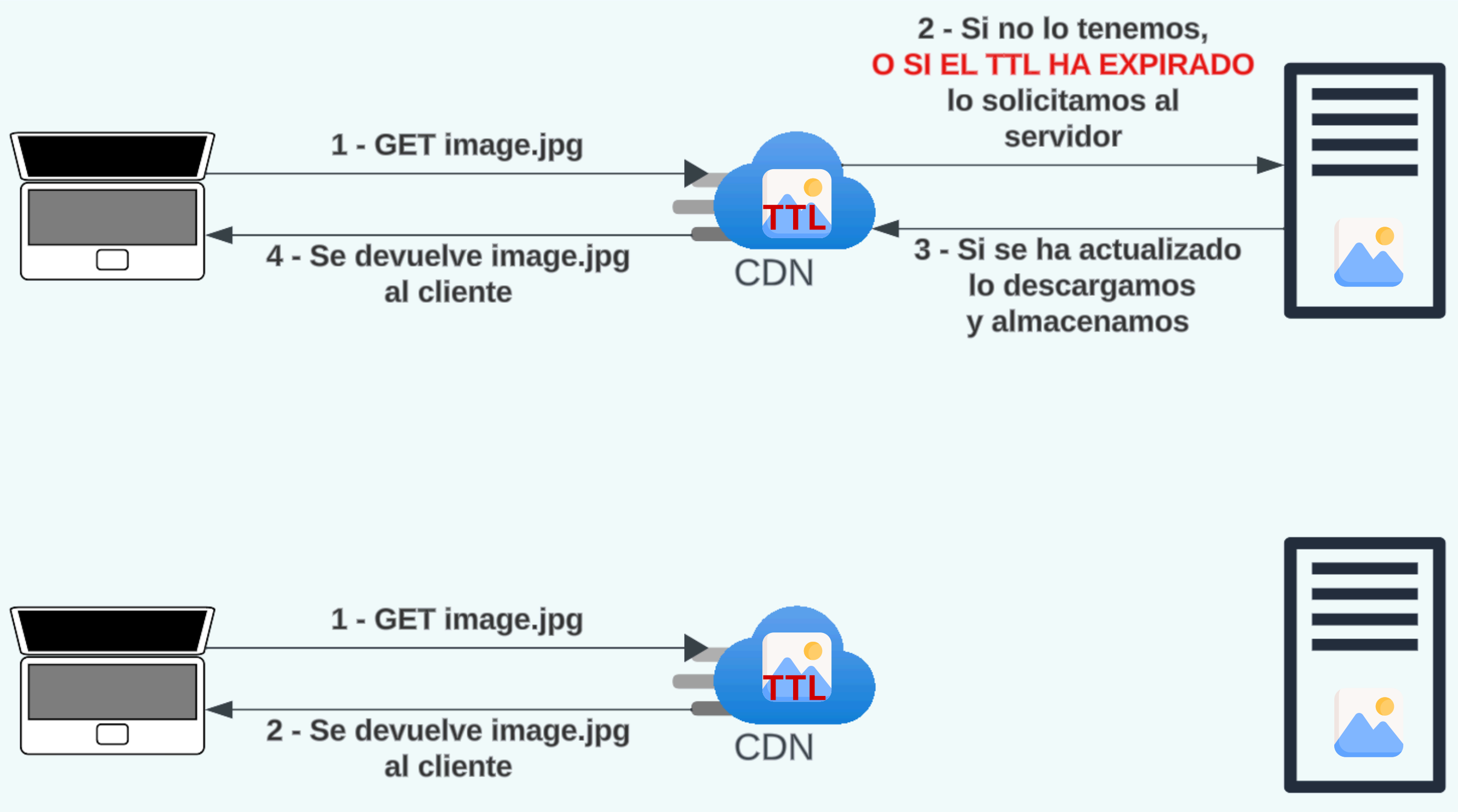


# ESTRATEGIA PULL

- El CDN solicita **bajo demanda** los recursos al servidor.
  - Si se modifica un recurso en el servidor estaríamos devolviendo un recurso desactualizado.
  - Debemos establecer un **TTL**.



# ESTRATEGIA PULL



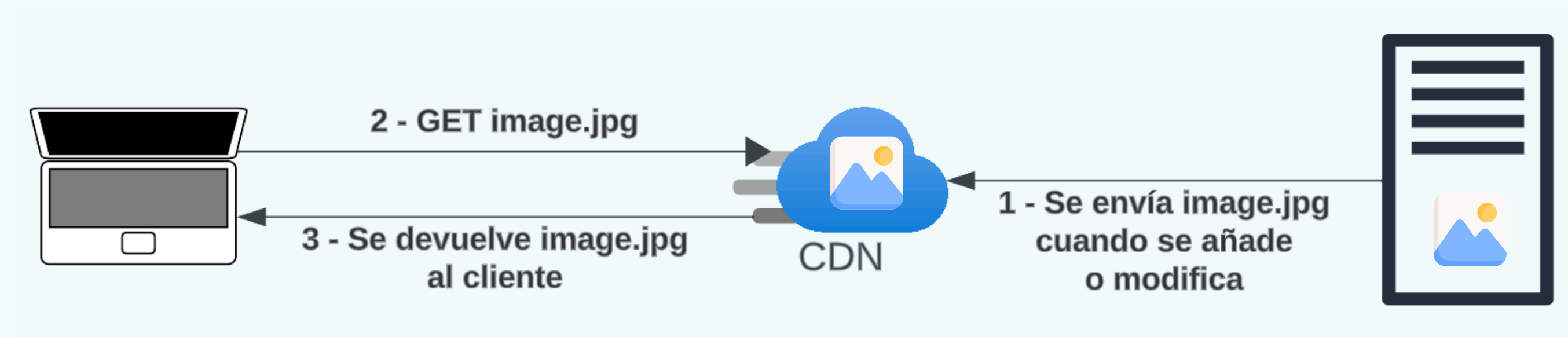
# PROS Y CONTRAS ESTRATEGIA PULL

- Ventajas:
  - No necesita de **mantenimiento** por nuestra parte.
    - El CDN se encarga de solicitar los recursos cuando el TTL expire.
- Desventajas:
  - Algunas peticiones no estarán cacheadas.
  - Complejo establecer un **TTL óptimo**.
    - Si es muy alto los usuarios verán contenido desactualizado.
    - Si es muy bajo, el CDN hará demasiadas peticiones innecesarias al servidor.



# ESTRATEGIA PUSH

- El servidor envía al CDN los recursos cuando se añaden o modifican.



# PROS Y CONTRAS ESTRATEGIA PUSH

- Ventajas:
  - El contenido siempre estará **actualizado**.
  - **Todas las peticiones estarán cacheadas**, incluso la primera.
- Desventajas:
  - El servidor tiene la **responsabilidad** de enviar recursos al CDN.
    - Necesita de mayor mantenimiento por nuestra parte.

# CONSIDERACIONES CDN

- El CDN es un **servicio** que tiene un **coste** importante.
  - Varía dependiendo de:
    - Número de ficheros almacenados y su tamaño.
    - Cantidad de datos transmitidos (tanto hacia los usuarios como desde nuestro servidor)
  - Si nuestros usuarios están en la misma región seguramente no sea necesario.

# CONSIDERACIONES CDN

- El CDN es un **servicio** que tiene un **coste** importante.
  - Varía dependiendo de:
    - Número de ficheros almacenados y su tamaño.
    - Cantidad de datos transmitidos (tanto hacia los usuarios como desde nuestro servidor)
  - Si nuestros usuarios están en la misma región seguramente no sea necesario.
- **Debemos soportar caídas** en el servicio de CDN.
  - Si el CDN no responde debemos detectarlo y **solicitar los recursos directamente al servidor.**

# EJEMPLOS CDN

- Cloudflare
- Akamai
- Amazon CloudFront
- Google Cloud CDN
- Microsoft Azure CDN

