RSC & Streaming in Next.js 13

Jan Kott, Full-Stack Engineer
St. Gallen, 2023

Meine erste Bank. St.Galler Kantonalbank

Agenda

1. Next.js
2. Client Components
3. Server Components
4. Server-Side Rendering (SSR)
5. Streaming with Suspense

St.Galler Kantonalbank

What is Next.js?

- A React Framework with building blocks to create web applications
- Handles the tooling and configuration needed for React
- Provides additional structure, features, and optimizations

- Next.js 13 with App Router builds on Server Components, Suspense and more

Code Example

Client Components

St.Galler Kantonalbank

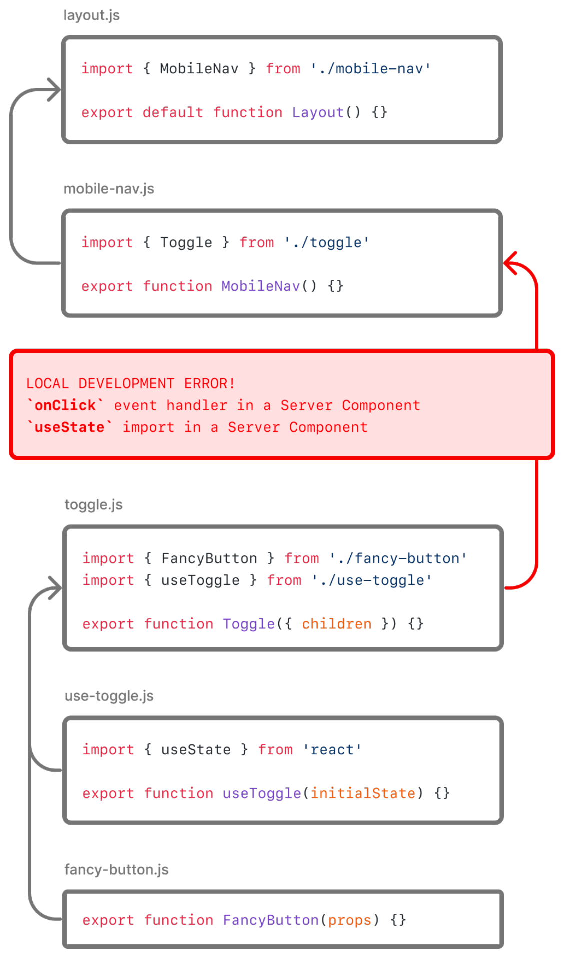
«React components that are fetched and rendered on the client.»

What are Client Components?

Benefits of Client Components

- Interactivity
  - Support state management, effects, and event listeners
  - Enables real-time user feedback and UI updates
- Browser APIs
  - Can leverage browser APIs, e.g., geolocation and localStorage
  - Enables the creation of UI tailored to specific use cases

Before 'use client'



layout.js

```
import { MobileNav } from './mobile-nav'

export default function Layout() {}
```

mobile-nav.js

```
import { Toggle } from './toggle'

export function MobileNav() {}
```

```
LOCAL DEVELOPMENT ERROR!
`onClick` event handler in a Server Component
`useState` import in a Server Component
```

toggle.js

```
import { FancyButton } from './fancy-button'
import { useToggle } from './use-toggle'

export function Toggle({ children }) {}
```

use-toggle.js

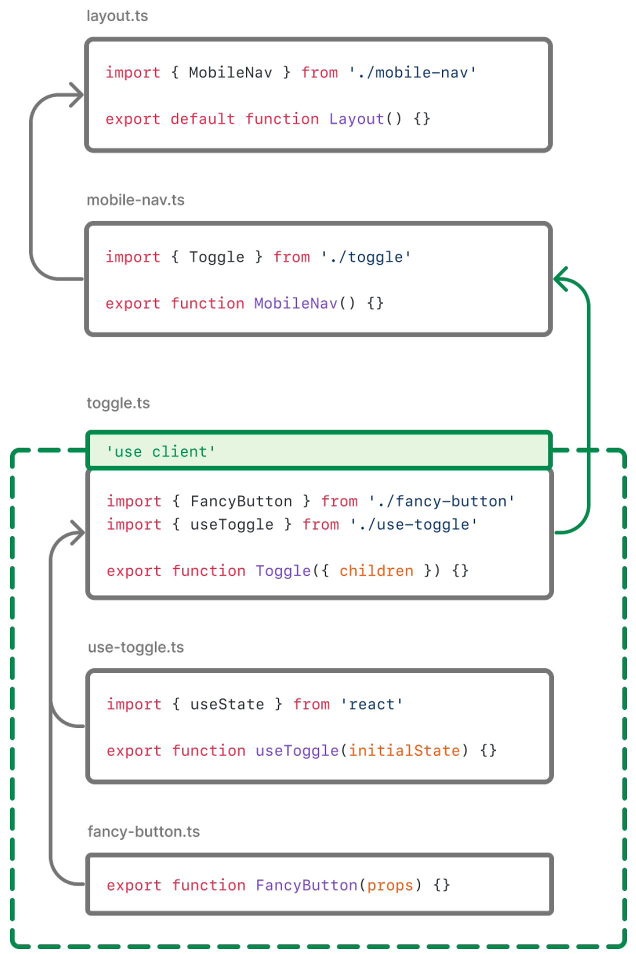
```
import { useState } from 'react'

export function useToggle(initialState) {}
```

fancy-button.js

```
export function FancyButton(props) {}
```

After 'use client'



layout.ts

```
import { MobileNav } from './mobile-nav'

export default function Layout() {}
```

mobile-nav.ts

```
import { Toggle } from './toggle'

export function MobileNav() {}
```

toggle.ts

```
'use client'
import { FancyButton } from './fancy-button'
import { useToggle } from './use-toggle'

export function Toggle({ children }) {}
```

use-toggle.ts

```
import { useState } from 'react'

export function useToggle(initialState) {}
```

fancy-button.ts

```
export function FancyButton(props) {}
```

Server Components

«React components that are fetched and rendered on the server.»

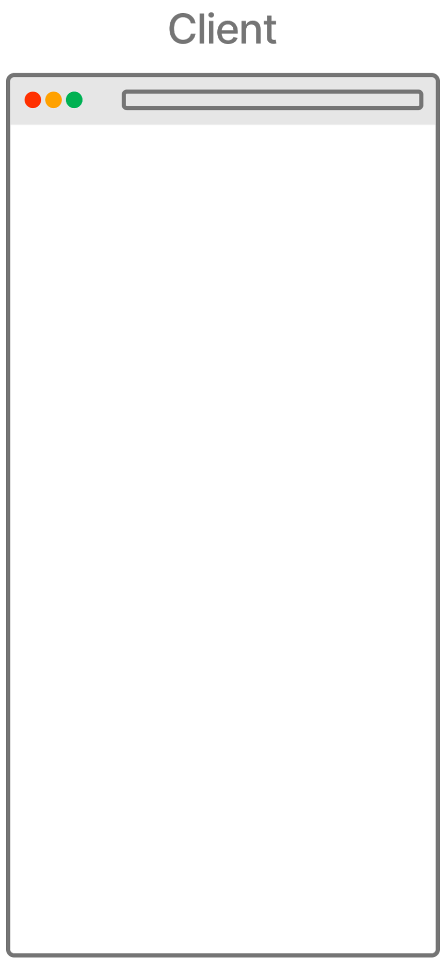
What are Server Components?

Benefits of Server Components

- Data Fetching
  - Facilitate data fetching on the server
  - Reduces data fetching time and number of client requests
- Security
  - Keep sensitive data and logic on the server
- Bundle Size
  - Reduce the impact of large dependencies on client JavaScript bundles
- Initial Page Load & First Contentful Paint (FCP)
  - Eliminates the need for clients to download, parse, and execute JavaScript
- Streaming
  - Enable chunked rendering and streaming to clients

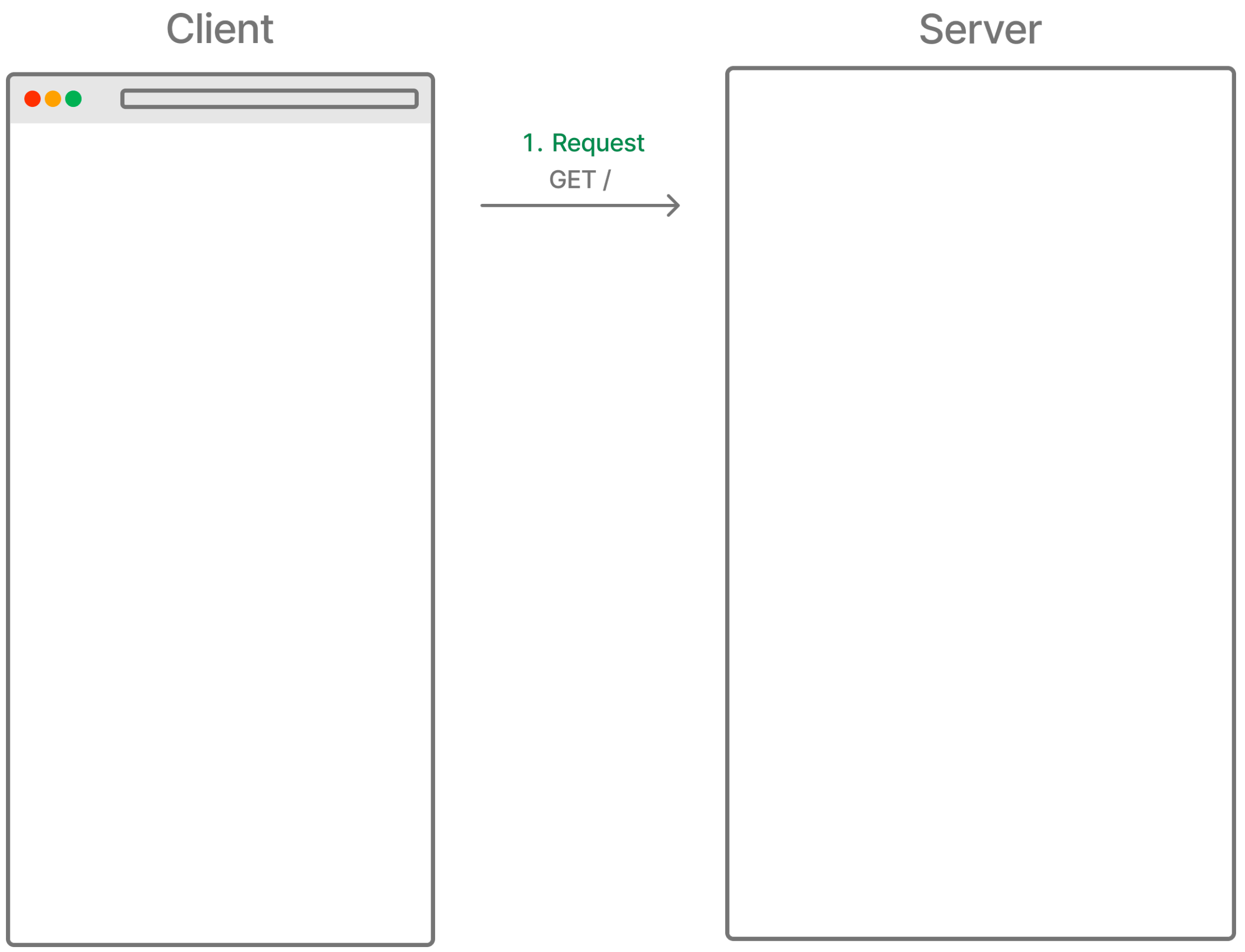
St.Galler Kantonalbank

How are Server Components rendered?



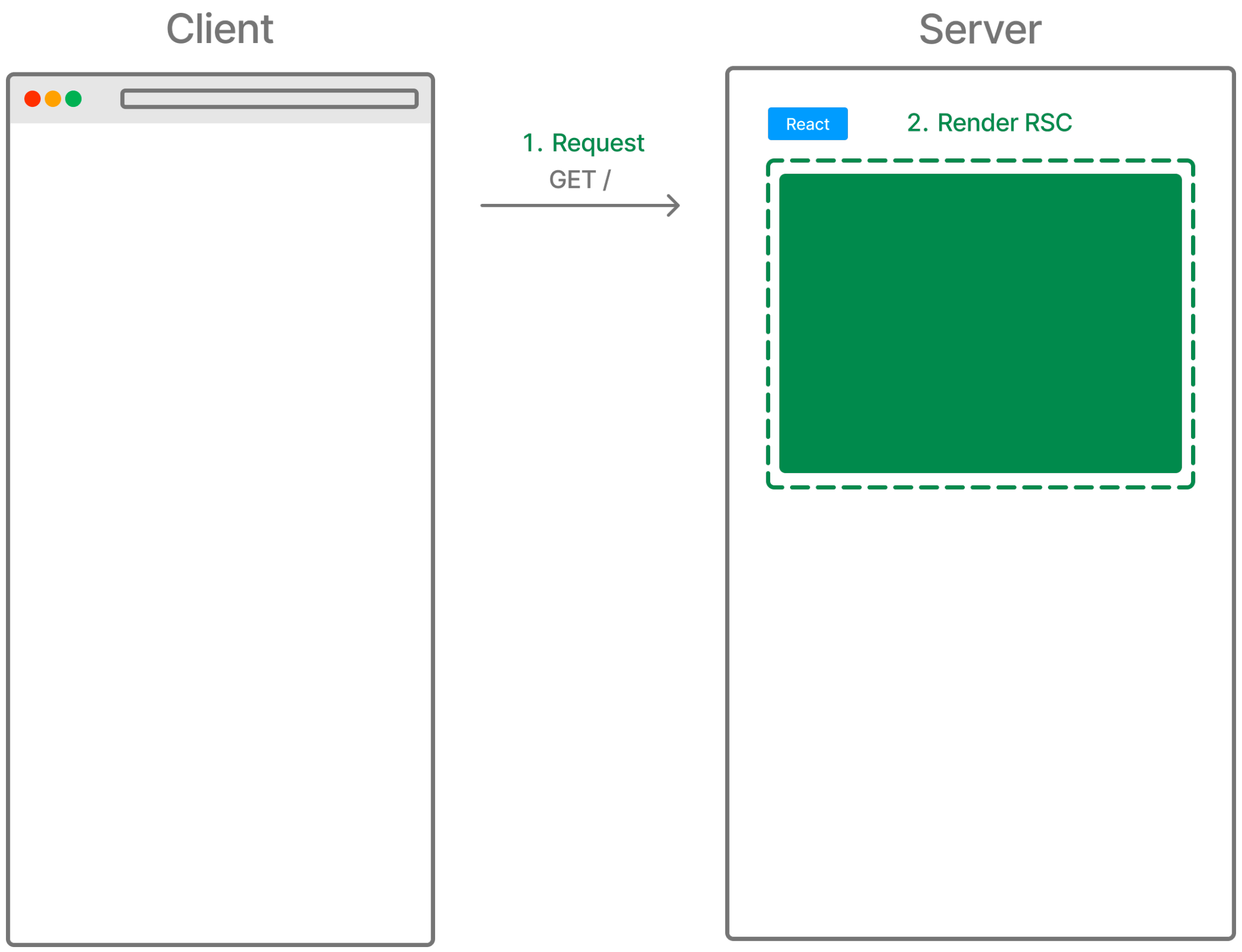
St.Galler Kantonalbank

How are Server Components rendered?

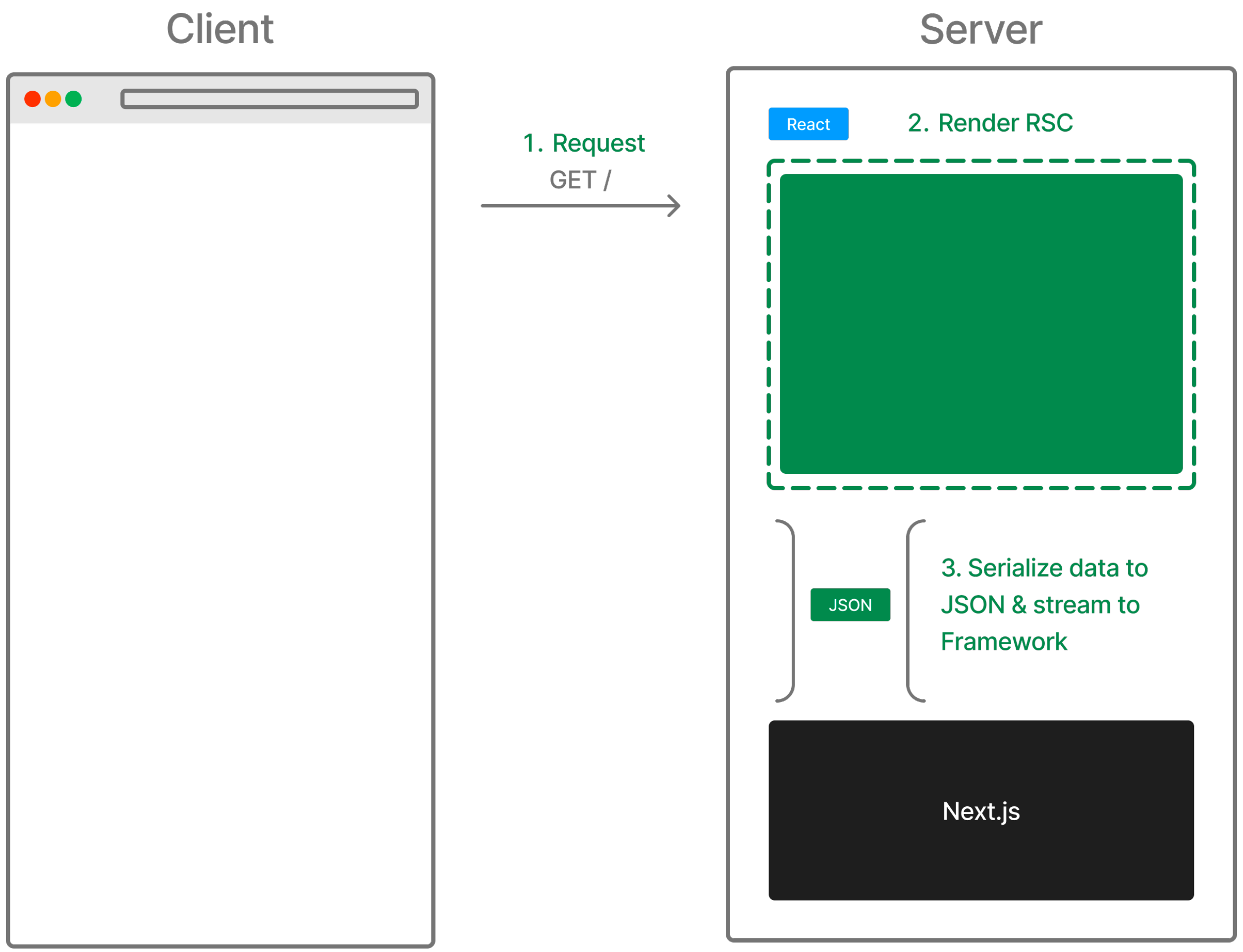


St.Galler Kantonalbank

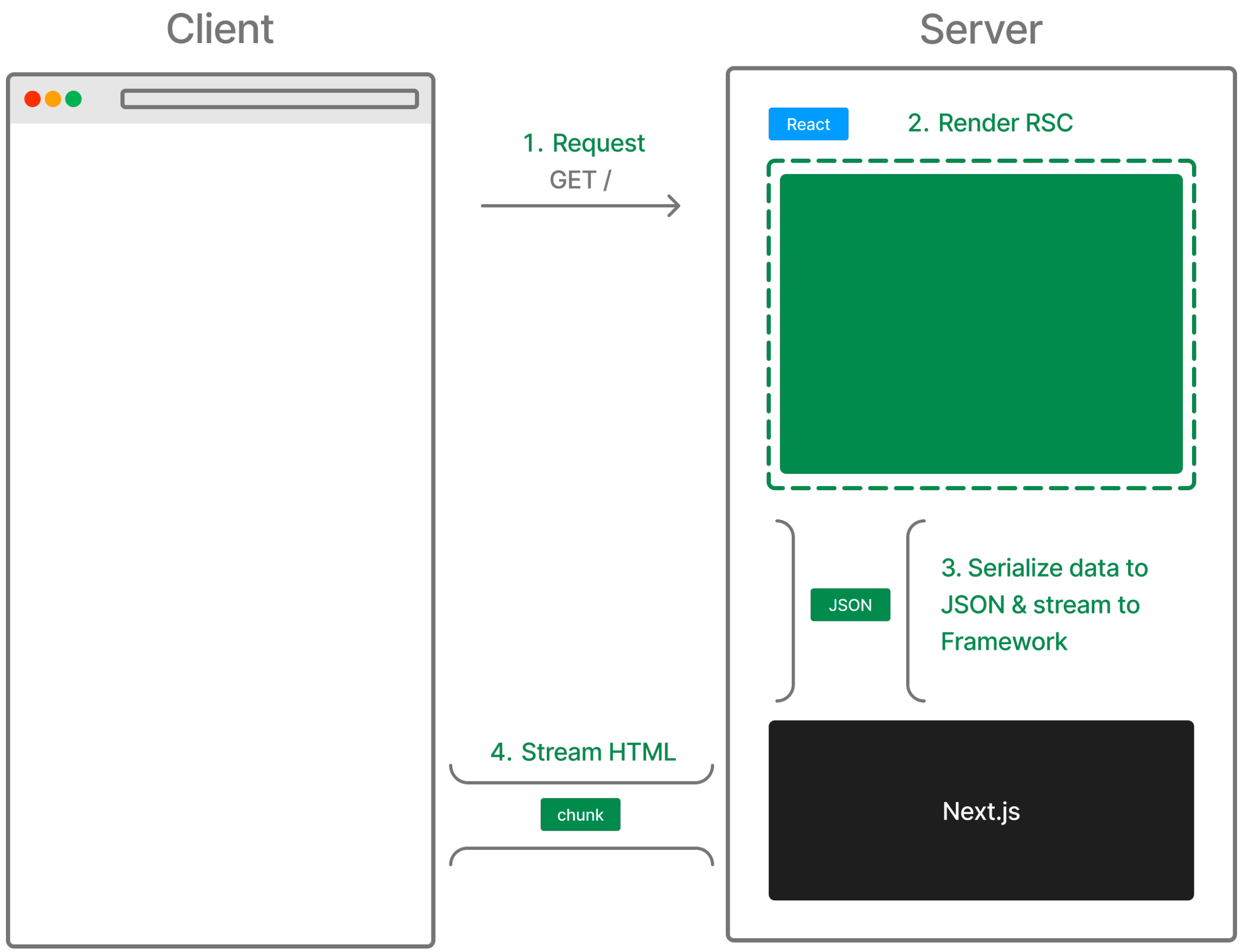
How are Server Components rendered?



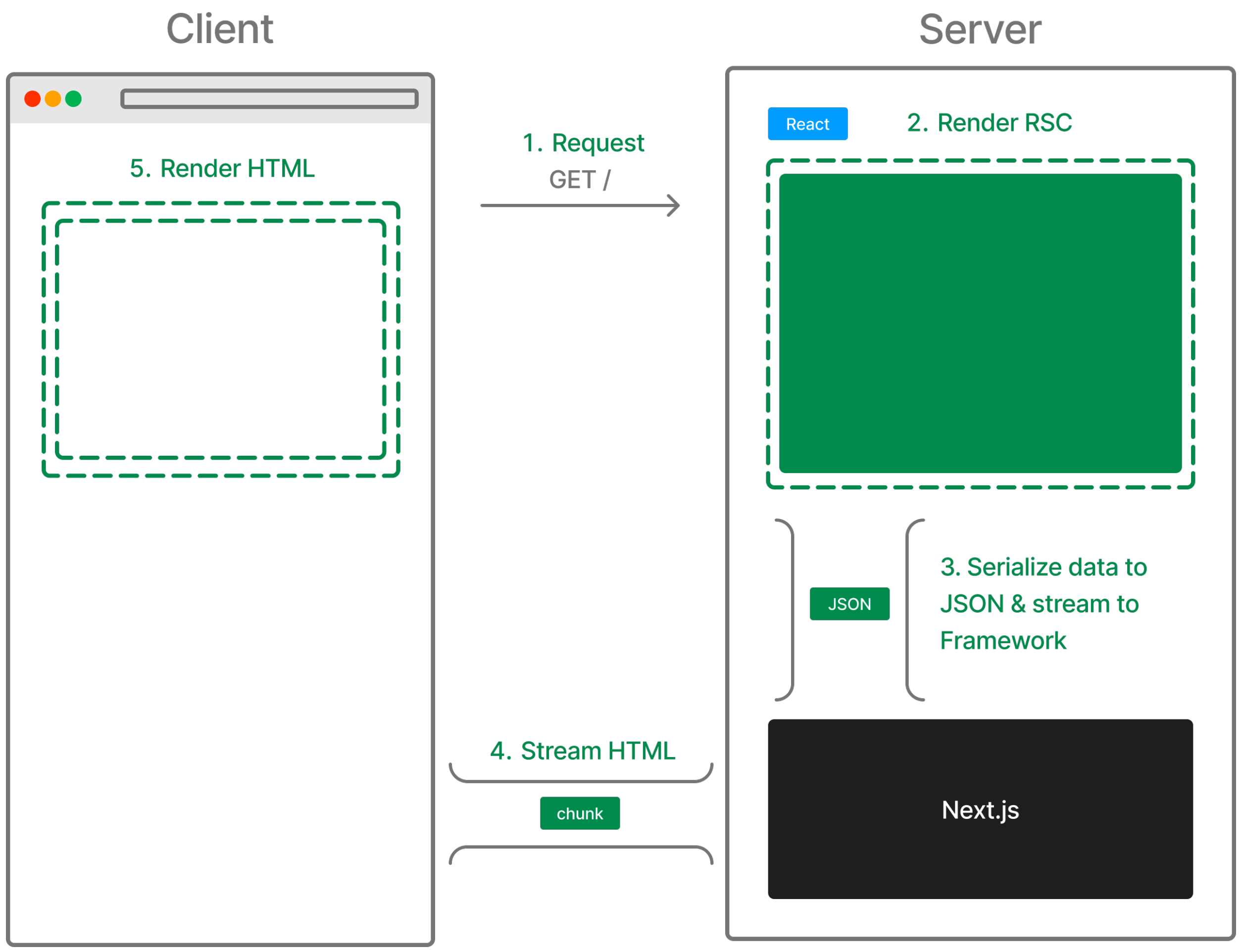
How are Server Components rendered?



How are Server Components rendered?

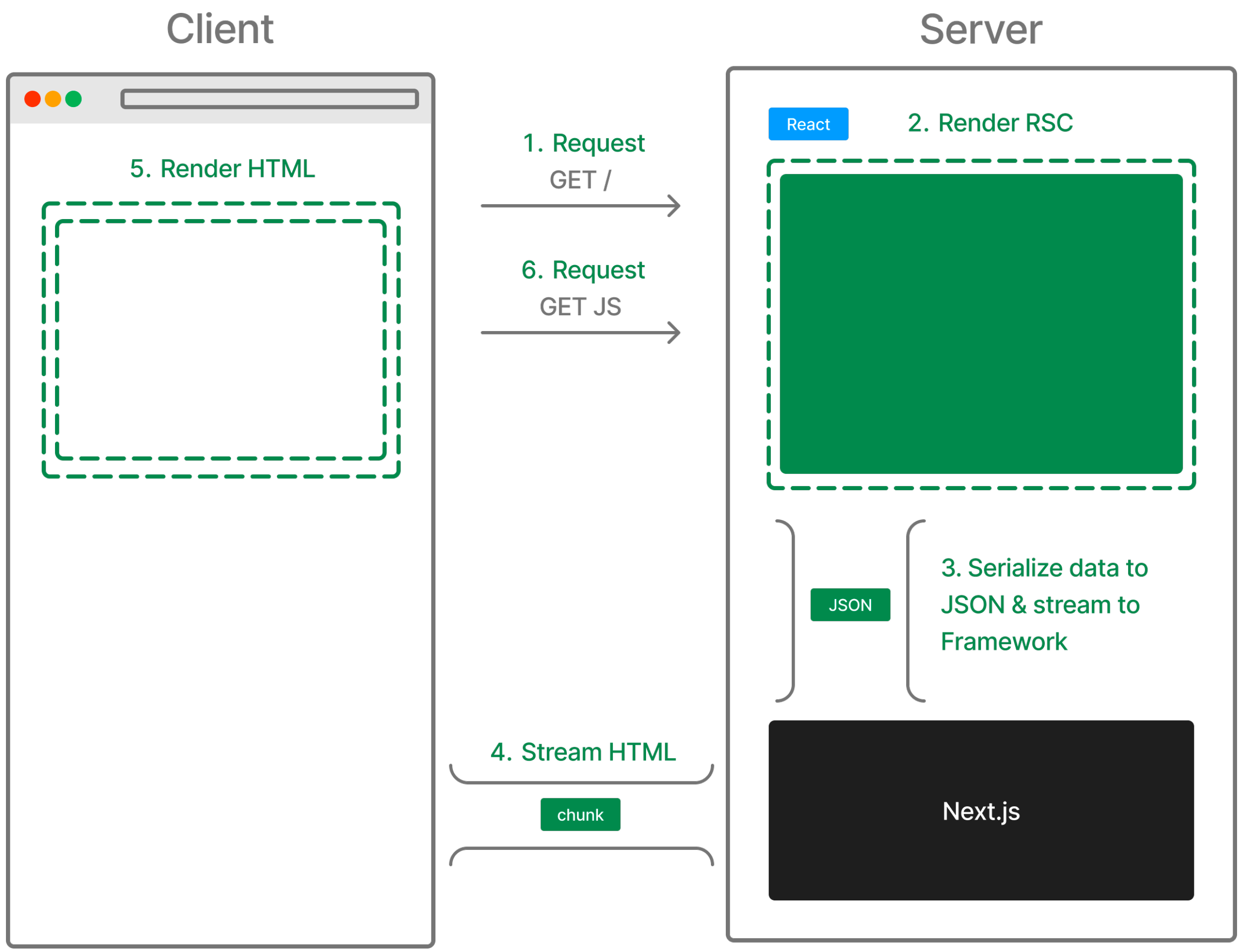


How are Server Components rendered?



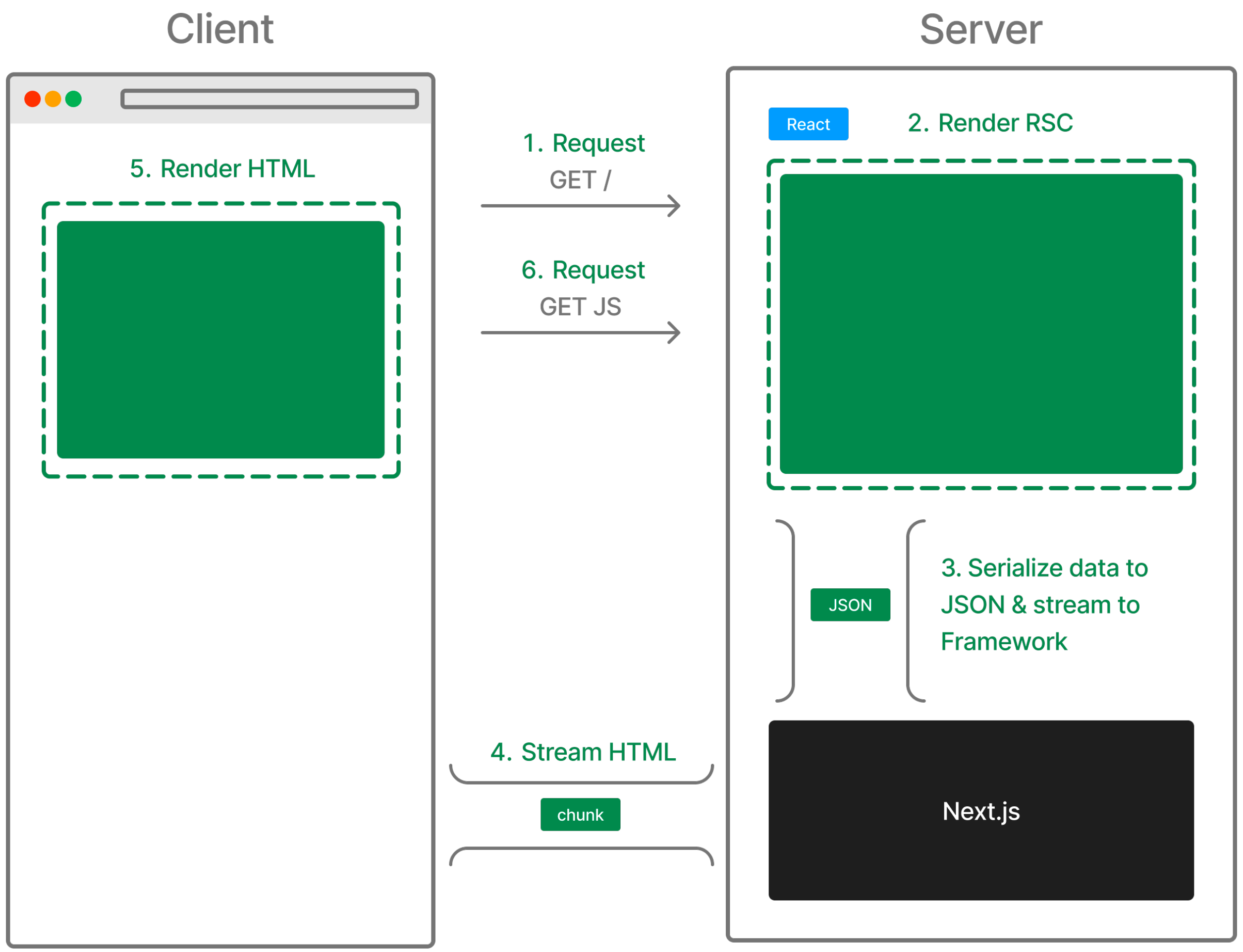
St.Galler Kantonalbank

How are Server Components rendered?

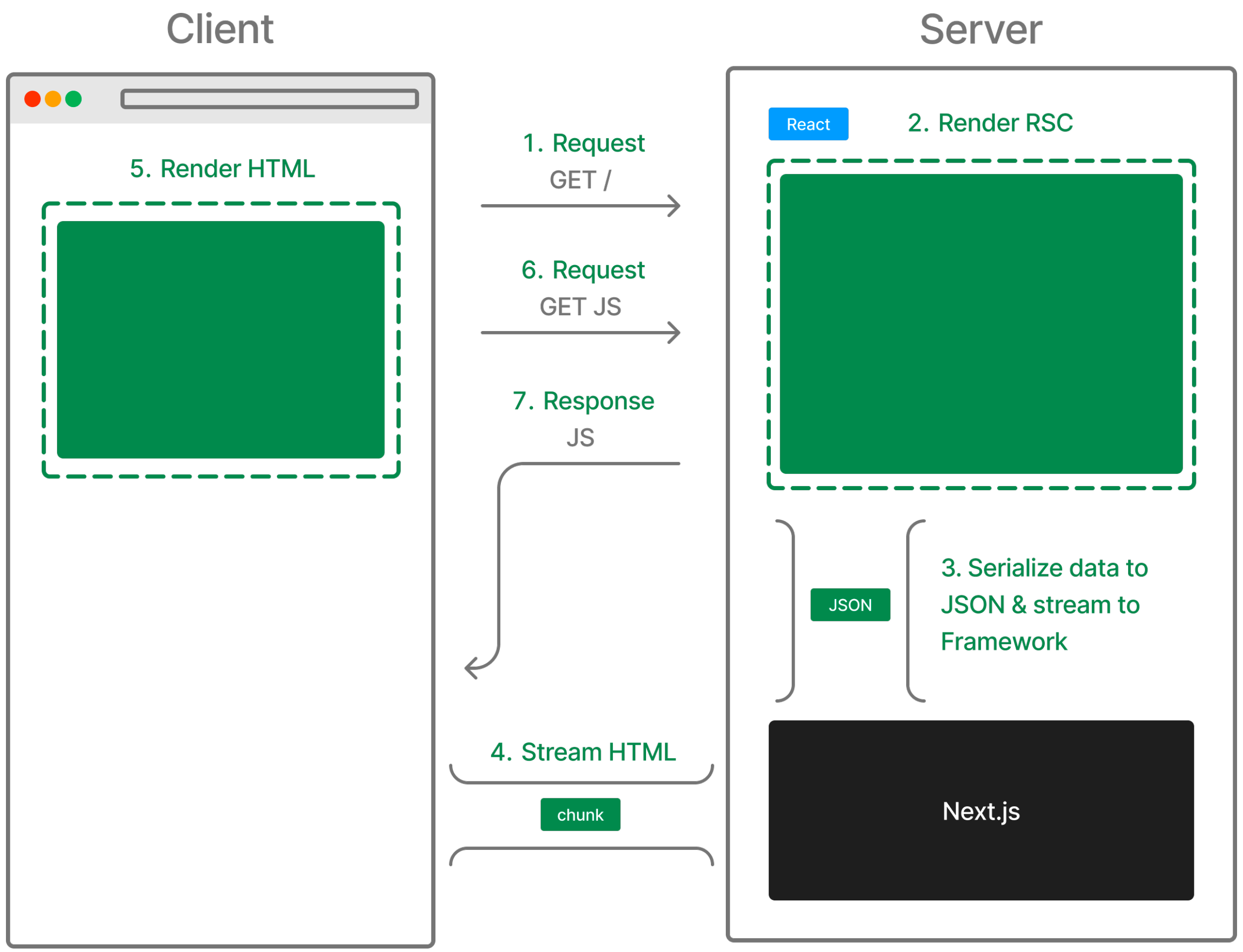


St.Galler Kantonalbank

How are Server Components rendered?

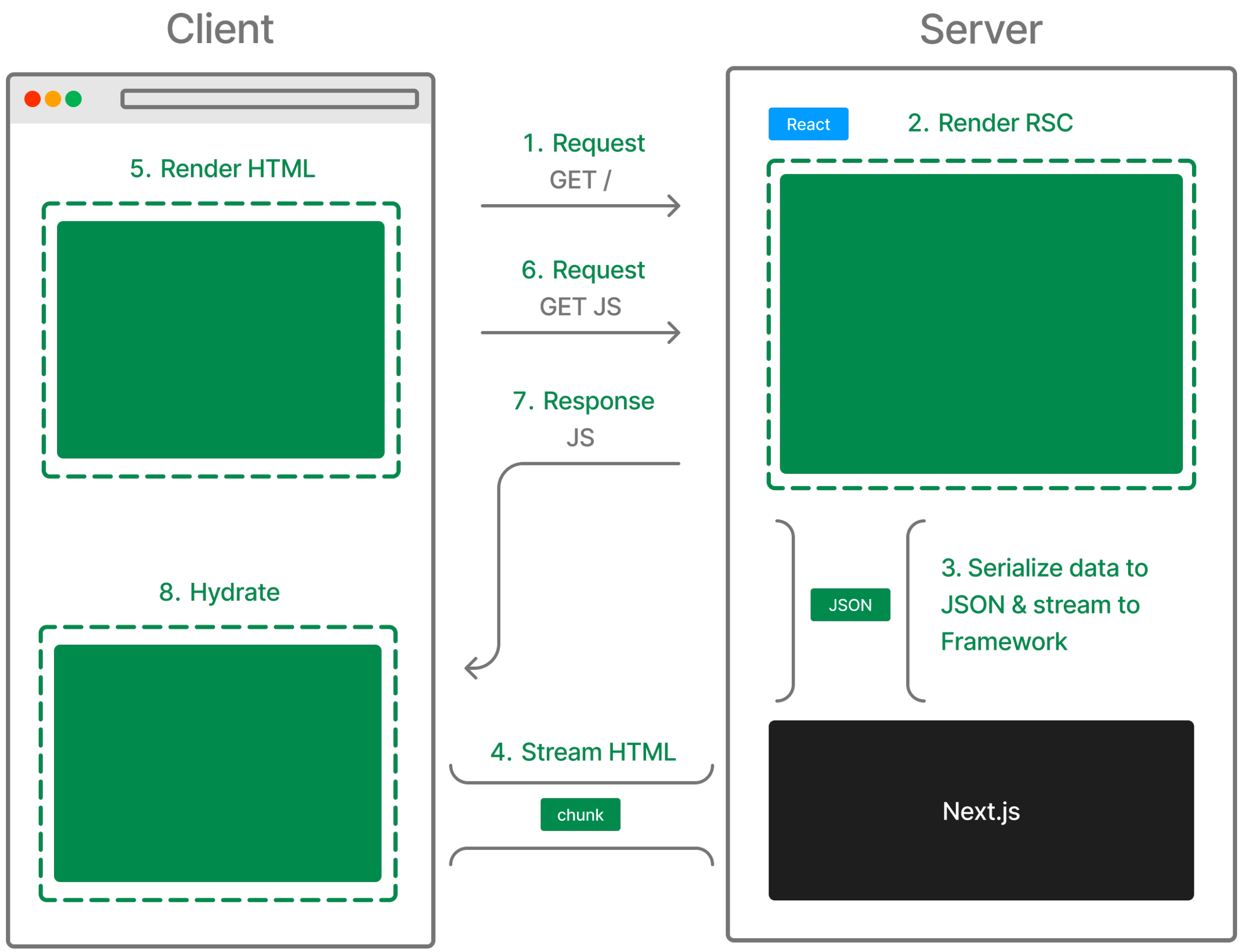


How are Server Components rendered?



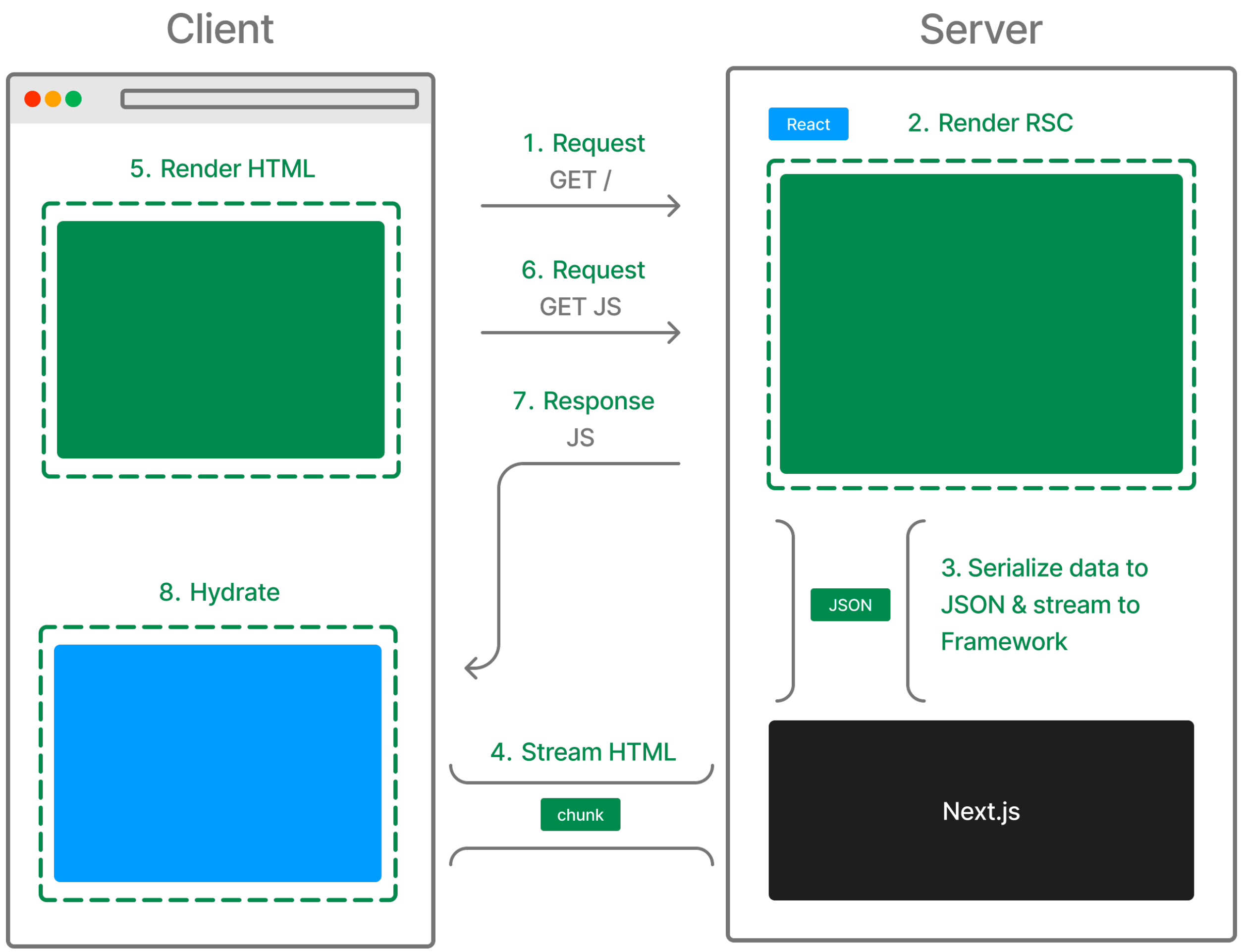
St.Galler Kantonalbank

How are Server Components rendered?



St.Galler Kantonalbank

How are Server Components rendered?



When are Server and when are Client Components used?

What wants to be done?	Server Component	Client Component
Fetch data	✓	✗
Access backend resources (directly)	✓	✗
Keep sensitive information on the server	✓	✗
Keep large dependencies on the server / Reduce client-side JavaScript	✓	✗
Add interactivity and event listeners	✗	✓
Use State and Lifecycle Effects	✗	✓
Use browser-only APIs	✗	✓
Use custom hooks that depend on state, effects, or browser-only APIs	✗	✓
Use React Class components	✗	✓

St.Galler Kantonalbank

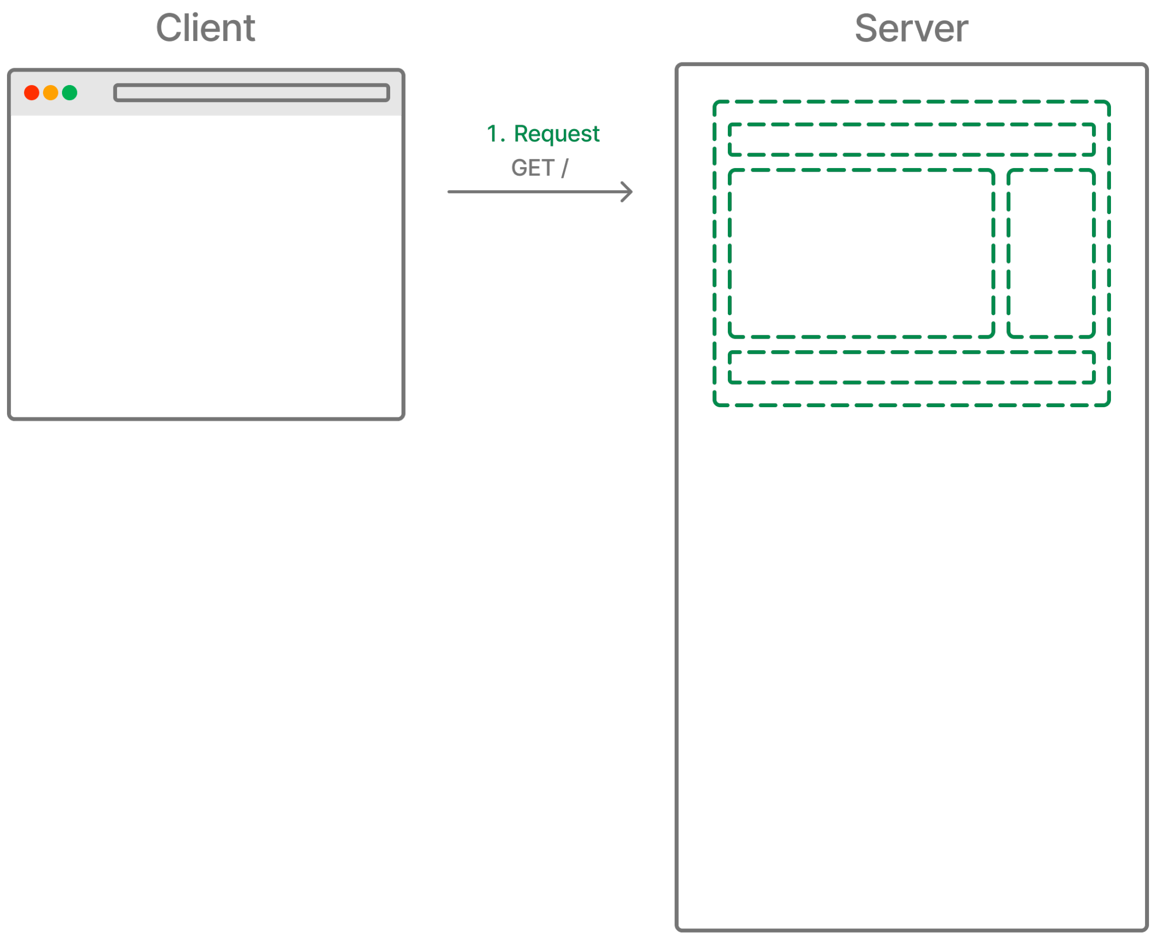
Server-Side Rendering (SSR)

«The process of rendering the complete HTML page on the server in response to a request and returning it to the client.»

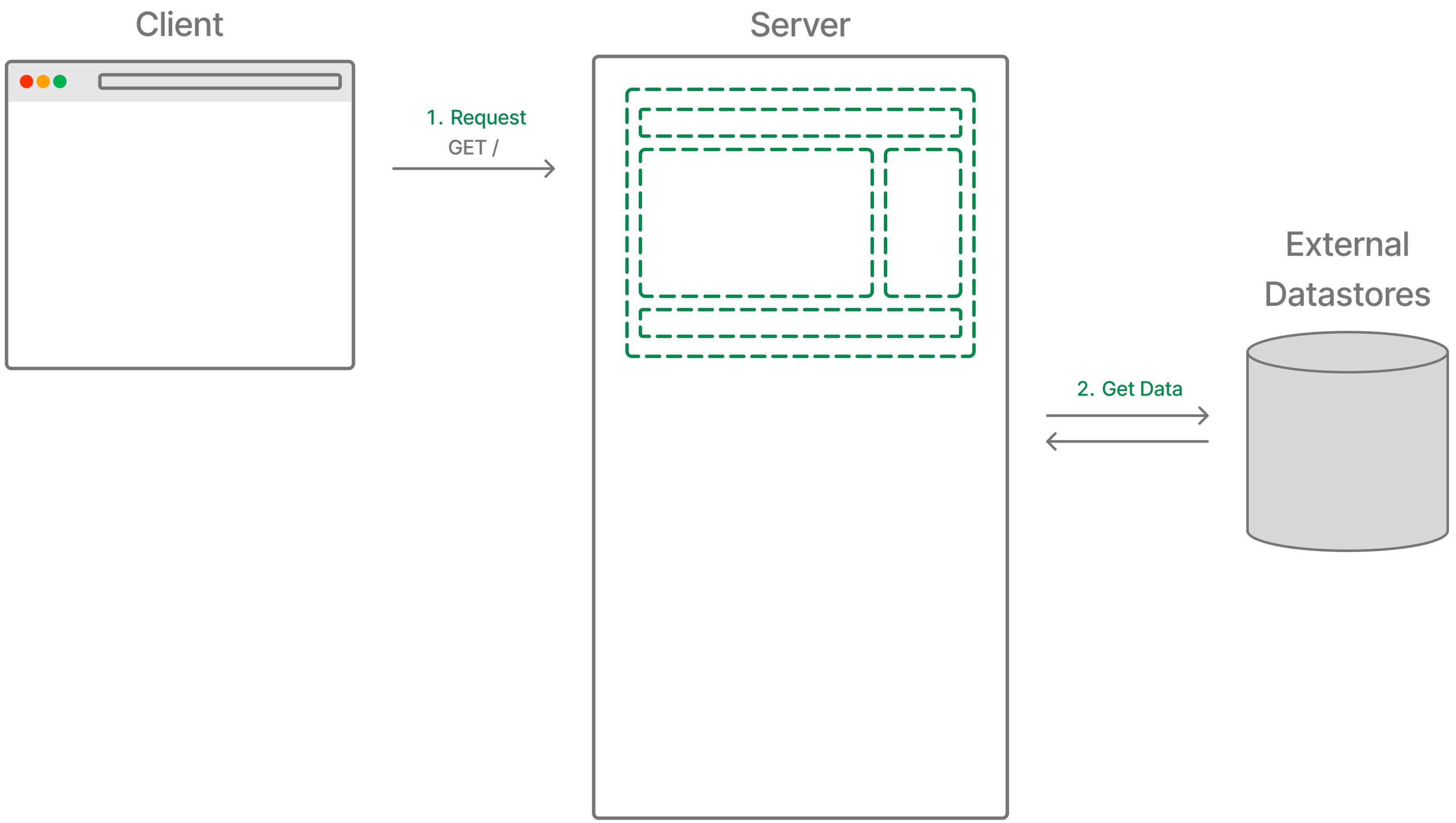
What is Server-Side Rendering (SSR)?

How does Server-Side Rendering (SSR) work?

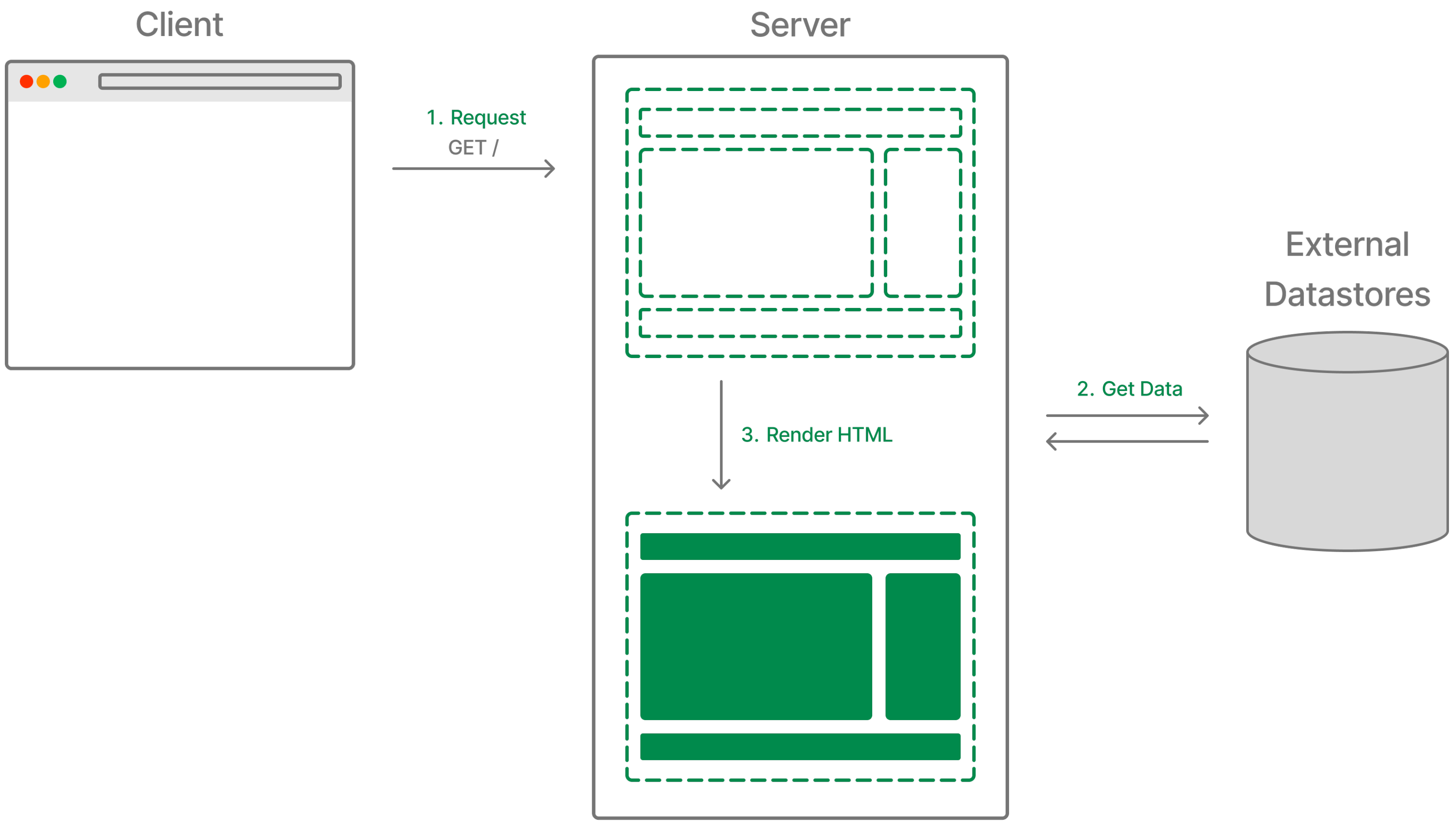
How does Server-Side Rendering (SSR) work?



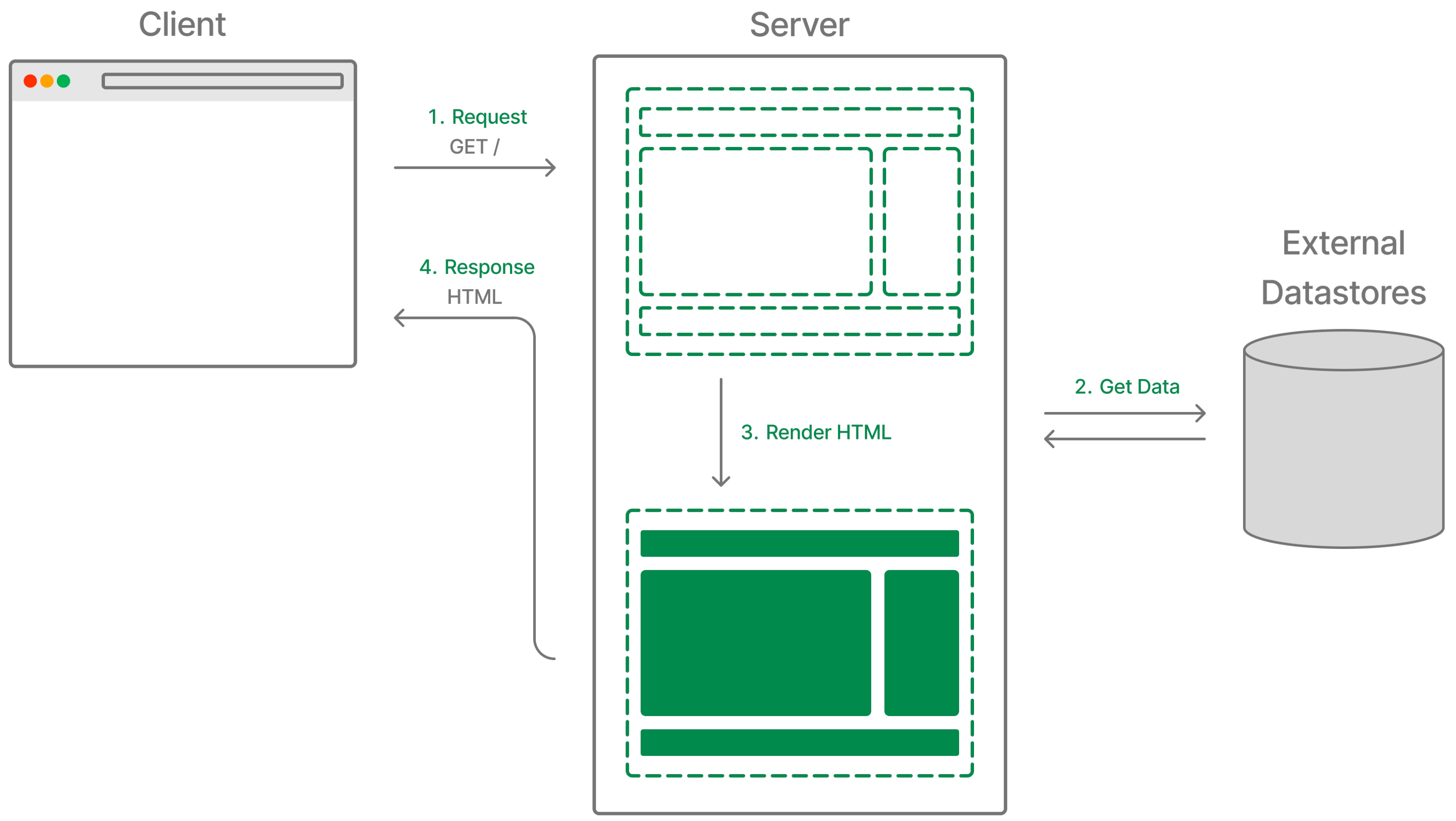
How does Server-Side Rendering (SSR) work?



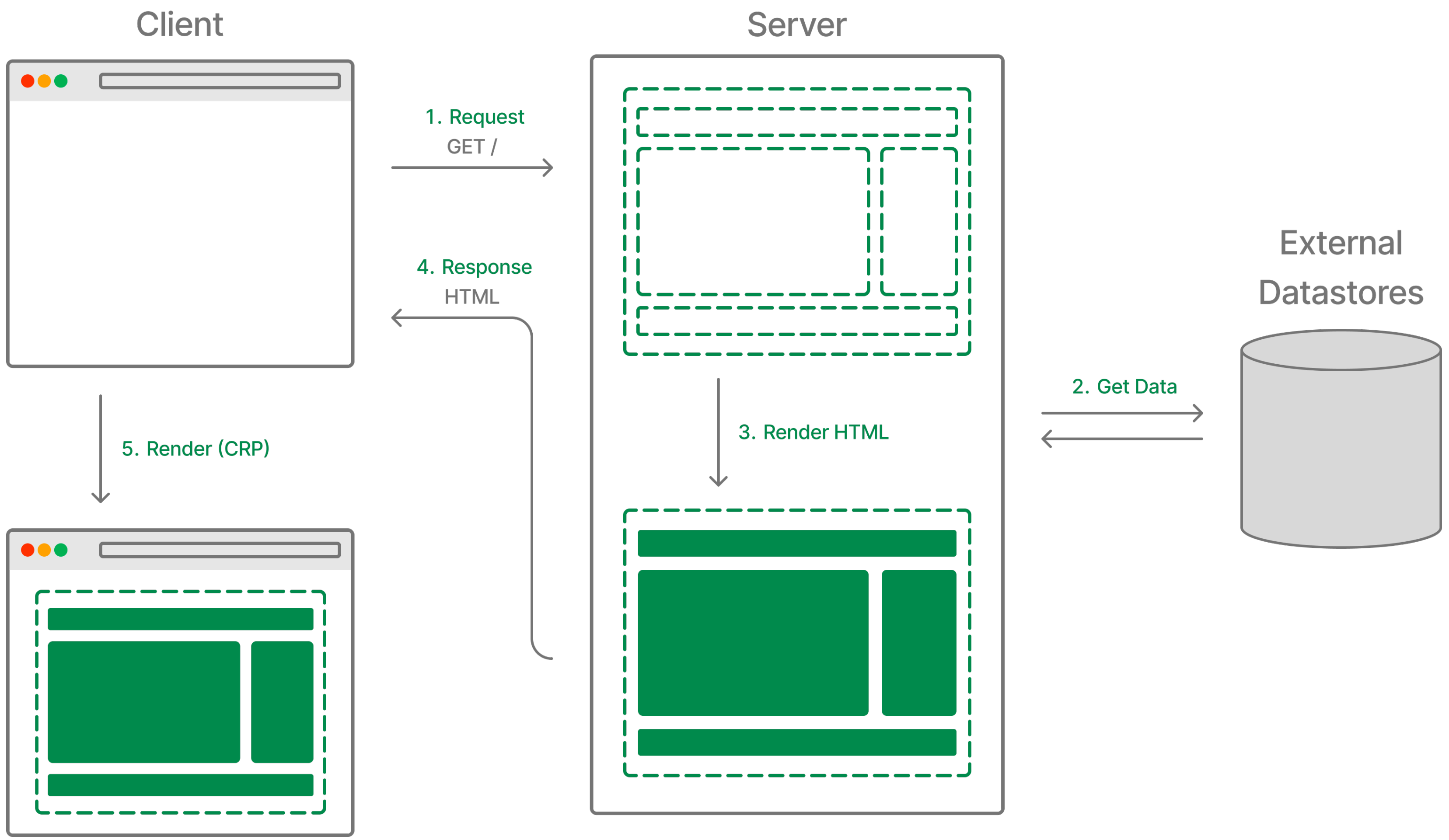
How does Server-Side Rendering (SSR) work?



How does Server-Side Rendering (SSR) work?

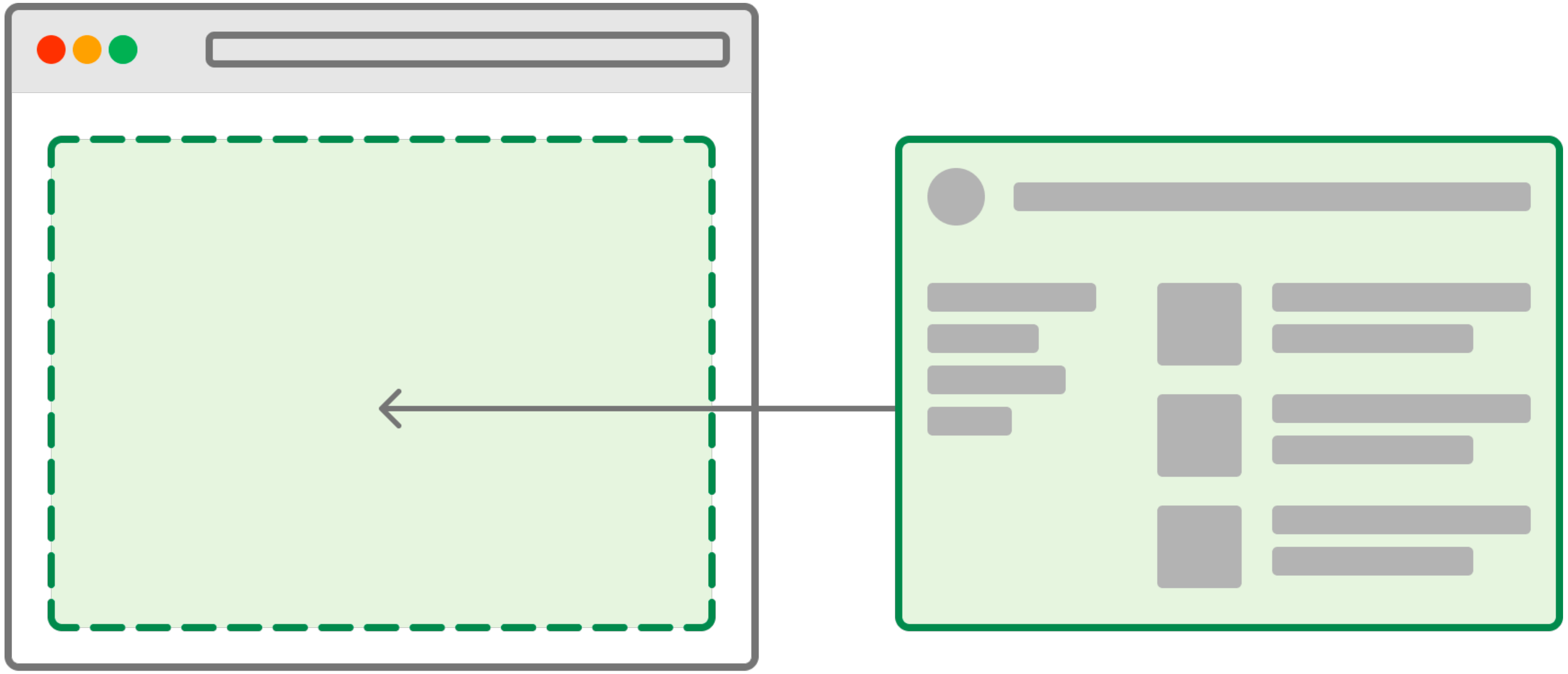


How does Server-Side Rendering (SSR) work?



St.Galler Kantonalbank

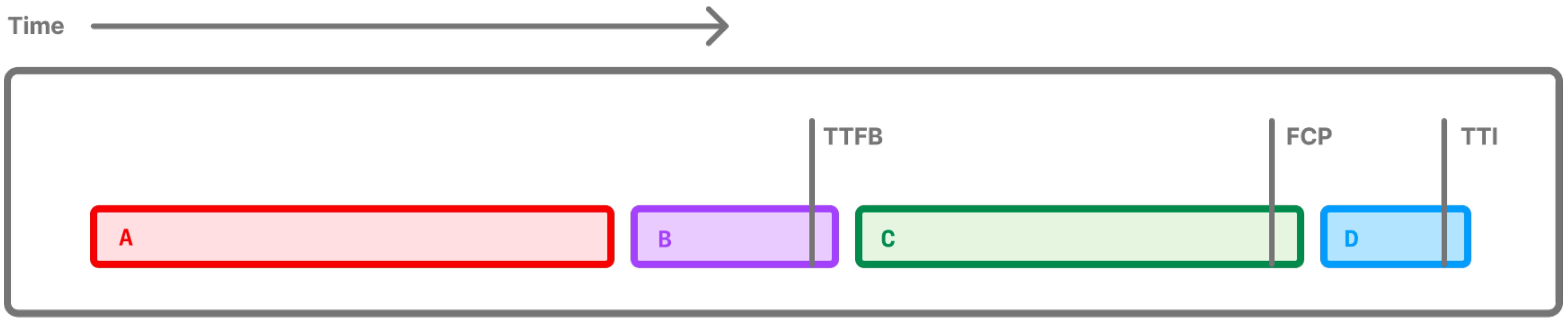
How does Server-Side Rendering (SSR) work?



No content in the browser while content is being rendered on the server

Server-rendered page sent to the client once all components are ready

How does Server-Side Rendering (SSR) work?



A	Fetching data on server
B	Rendering HTML on server
C	Loading Code on the client
D	Hydrating

TTFB	Time To First Byte
FCP	First Contentful Paint
TTI	Time To Interactive

Benefits of Server-Side Rendering (SSR)?

- Faster First Contentful Paint (FCP) & Time to Interactive (TTI)
- Additional budget for client-side JavaScript
- Client computation and bandwidth offloaded to the server

When should Server-Side Rendering (SSR) be used?

- When initial load times are more important than subsequent ones
- When JavaScript and heavy interactivity is not required on the client
- When client-side routing is not required
- When computation and bandwidth are to be offloaded from the client to the server

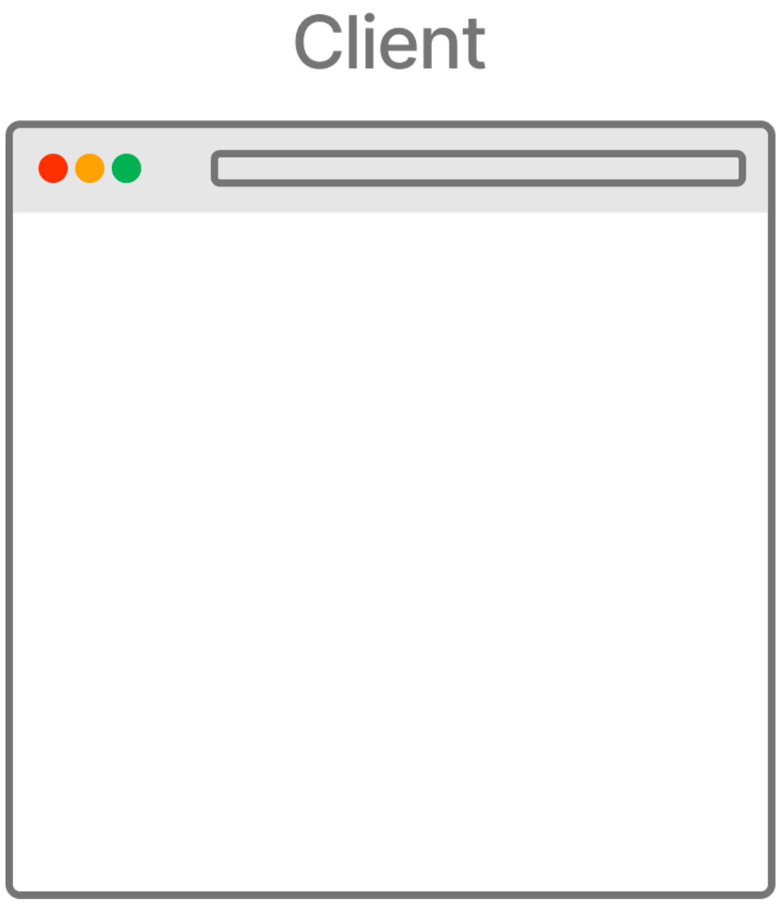
St.Galler Kantonalbank

Streaming with Suspense

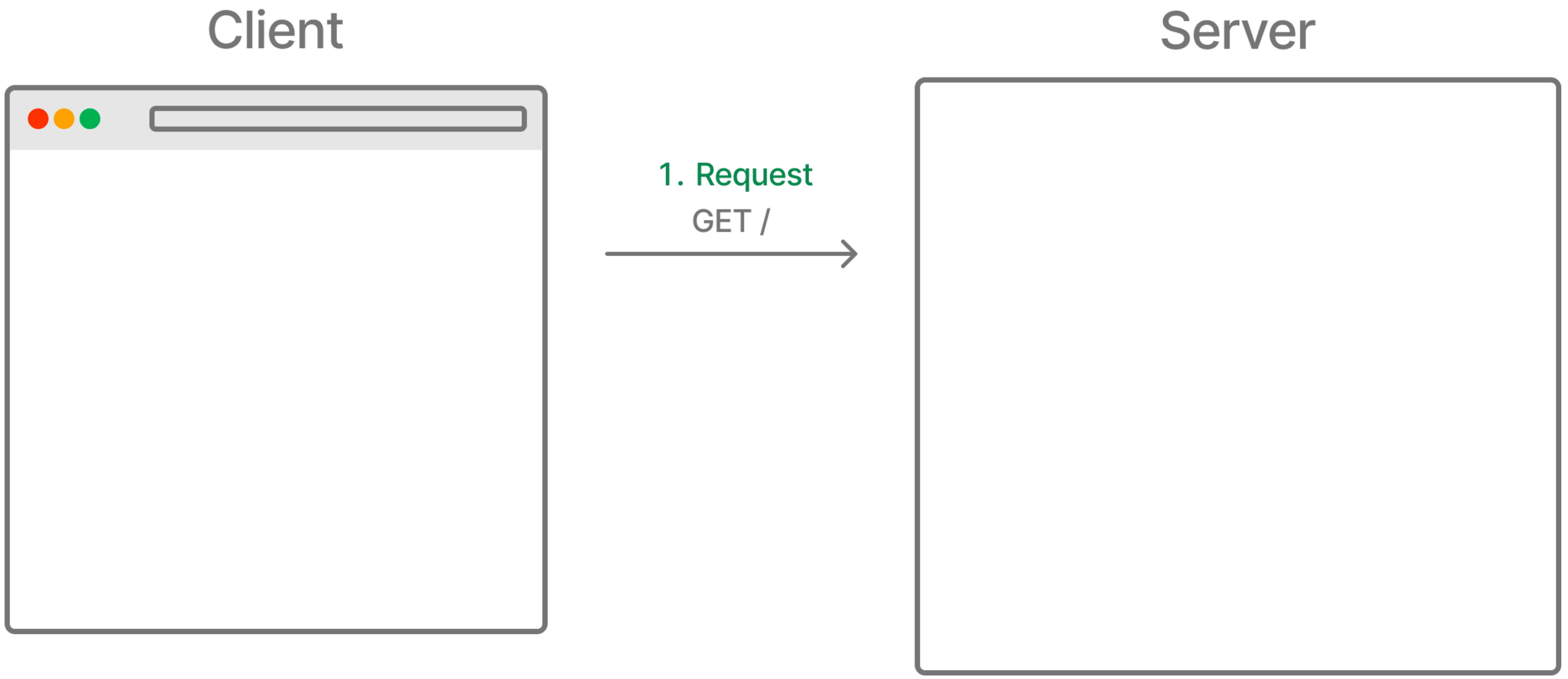
«The process of streaming a complete HTML page one chunk at a time instead of sending it in a single large response.»

What is Streaming with Suspense?

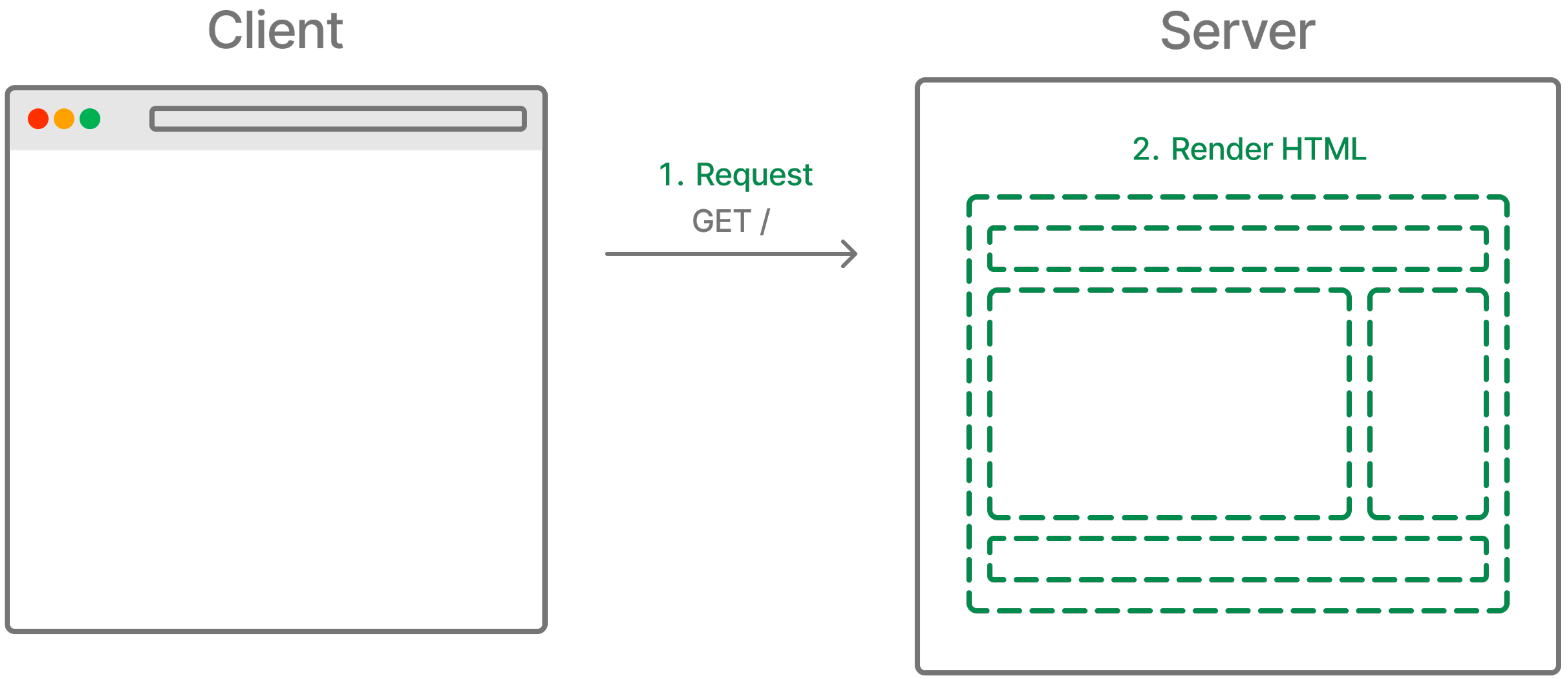
How does Streaming with Suspense work?



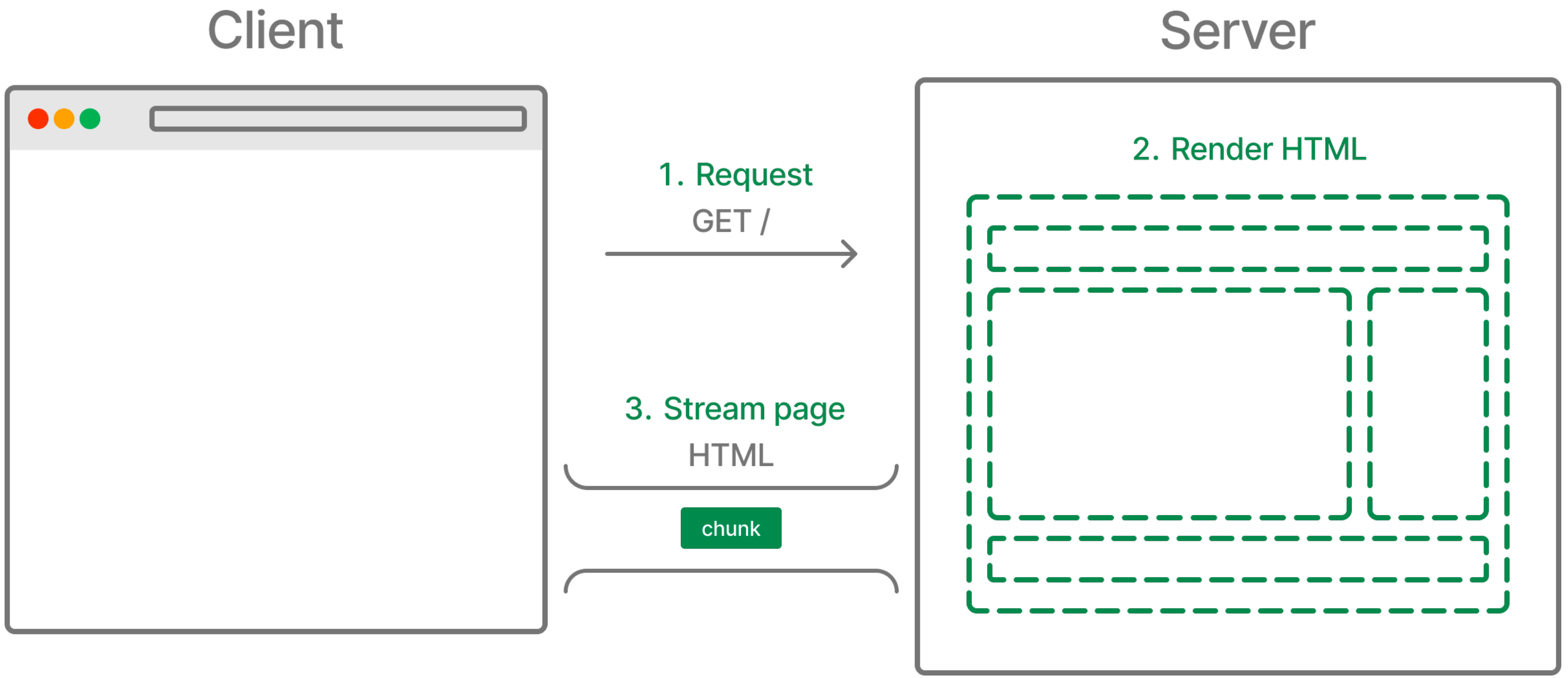
How does Streaming with Suspense work?



How does Streaming with Suspense work?

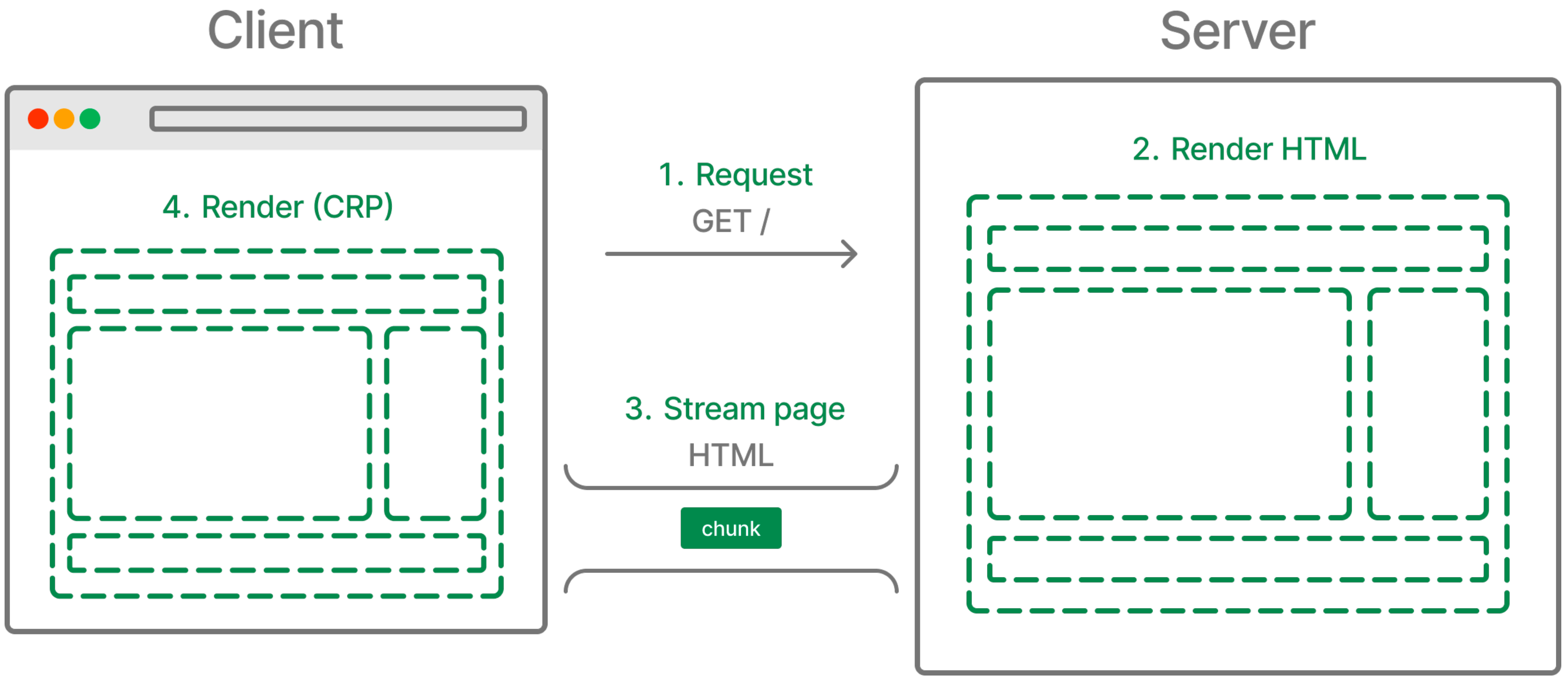


How does Streaming with Suspense work?

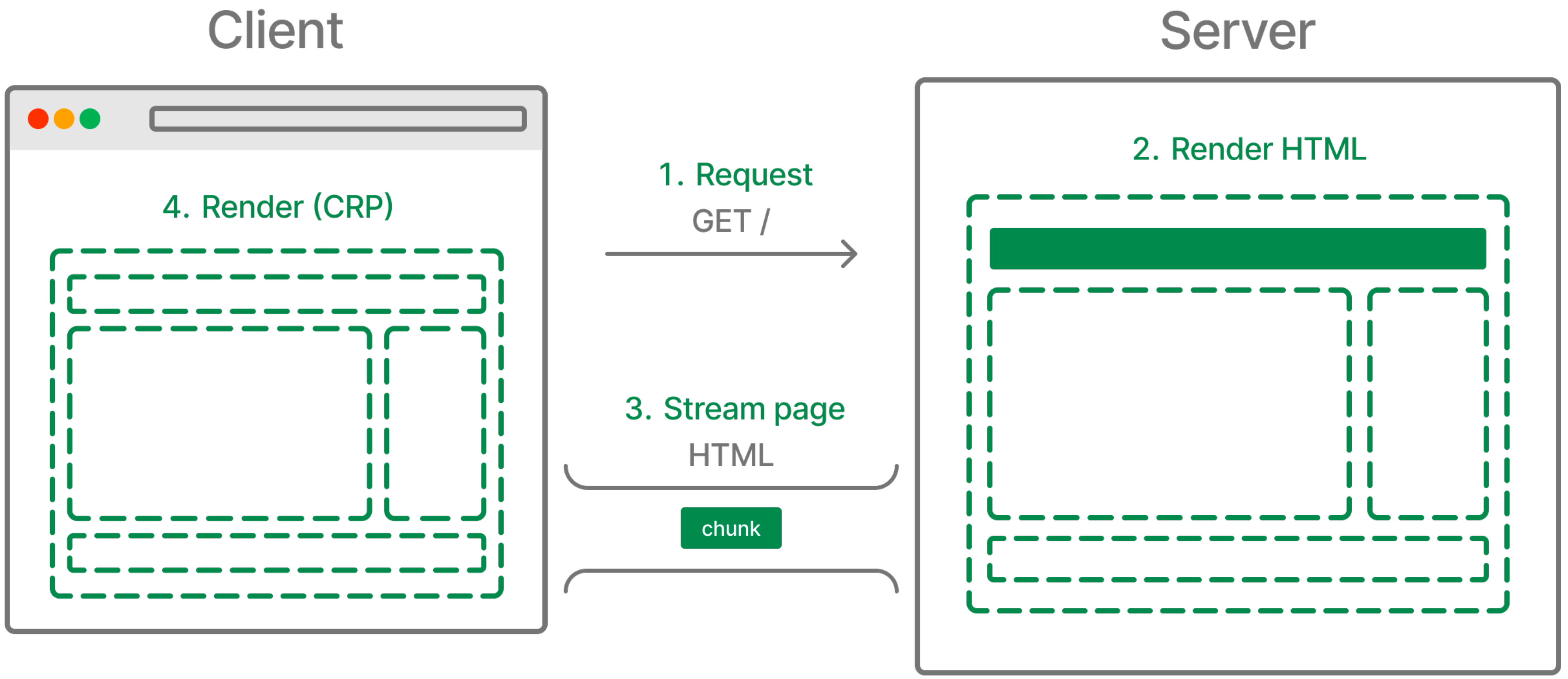


St.Galler Kantonalbank

How does Streaming with Suspense work?

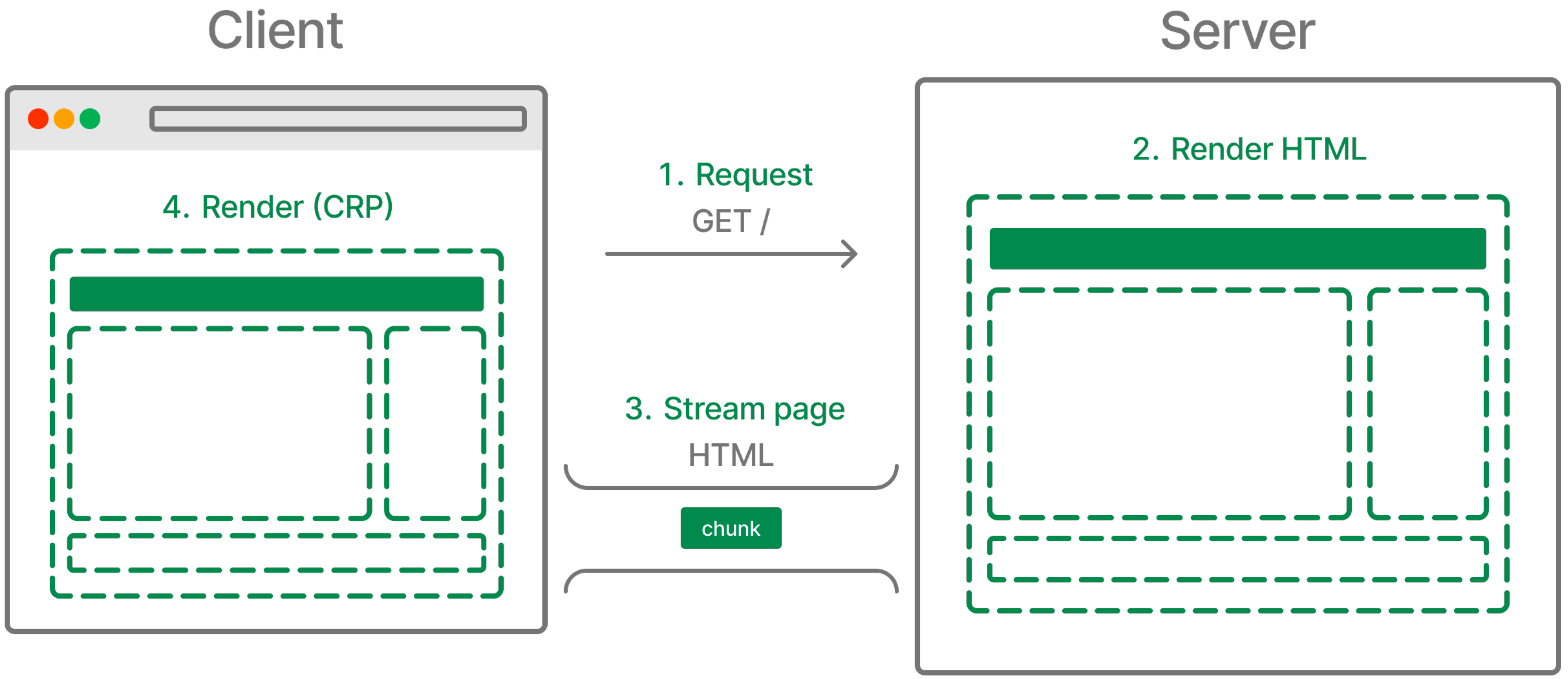


How does Streaming with Suspense work?



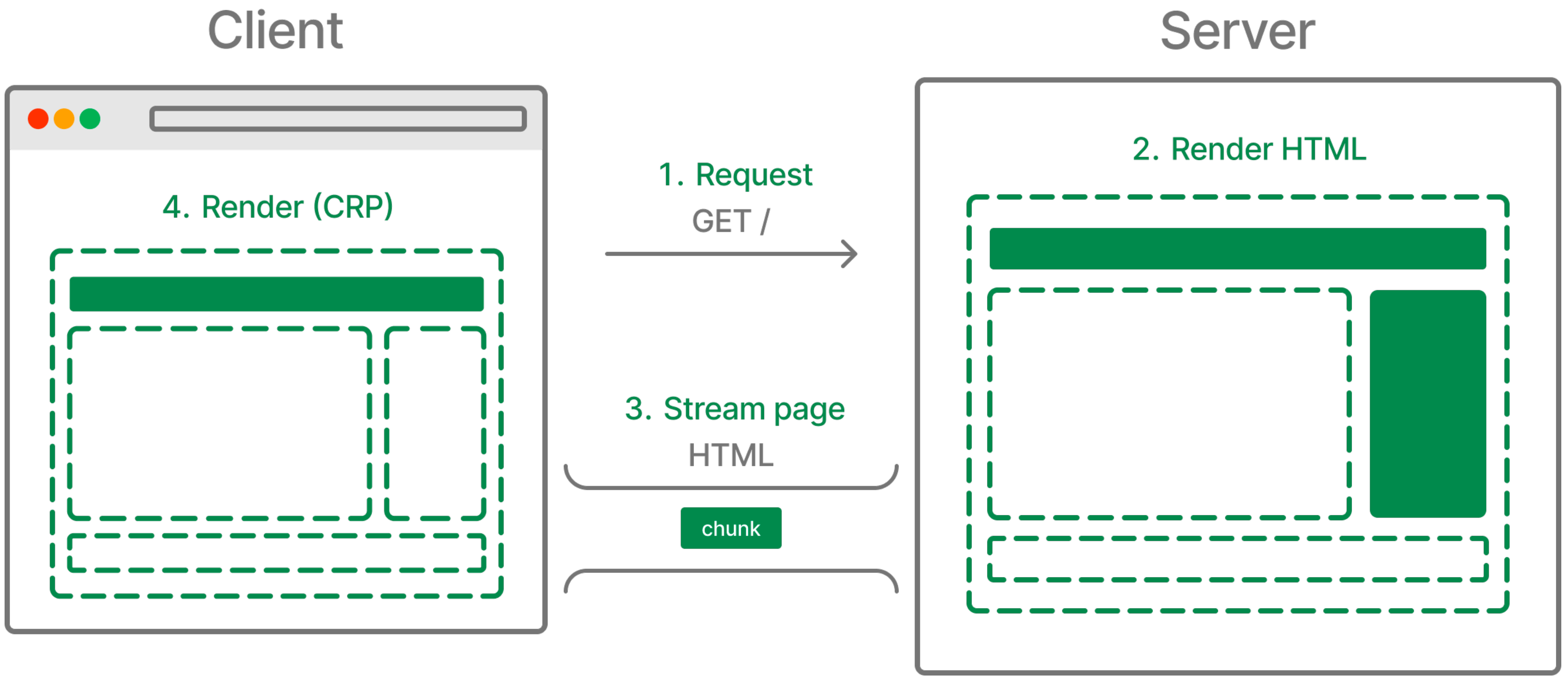
St.Galler Kantonalbank

How does Streaming with Suspense work?

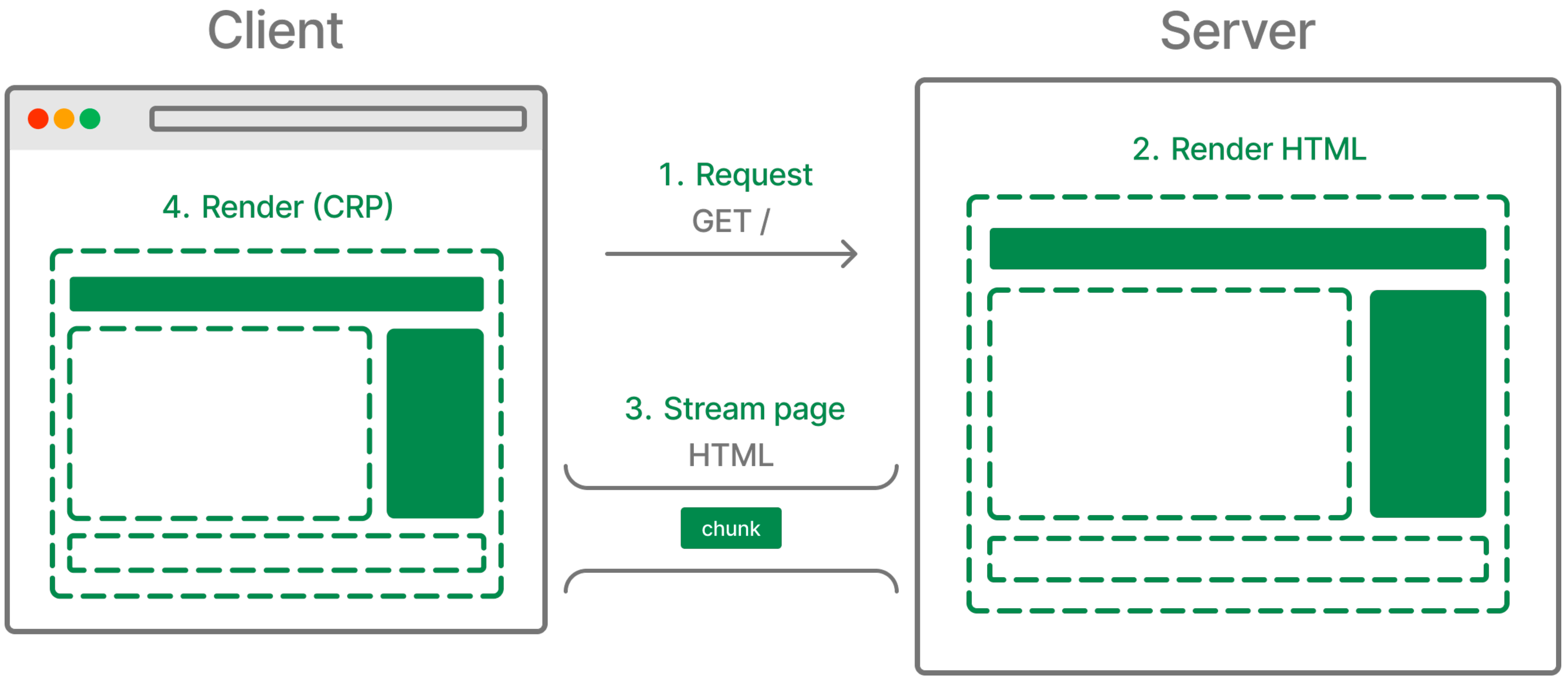


St.Galler Kantonalbank

How does Streaming with Suspense work?

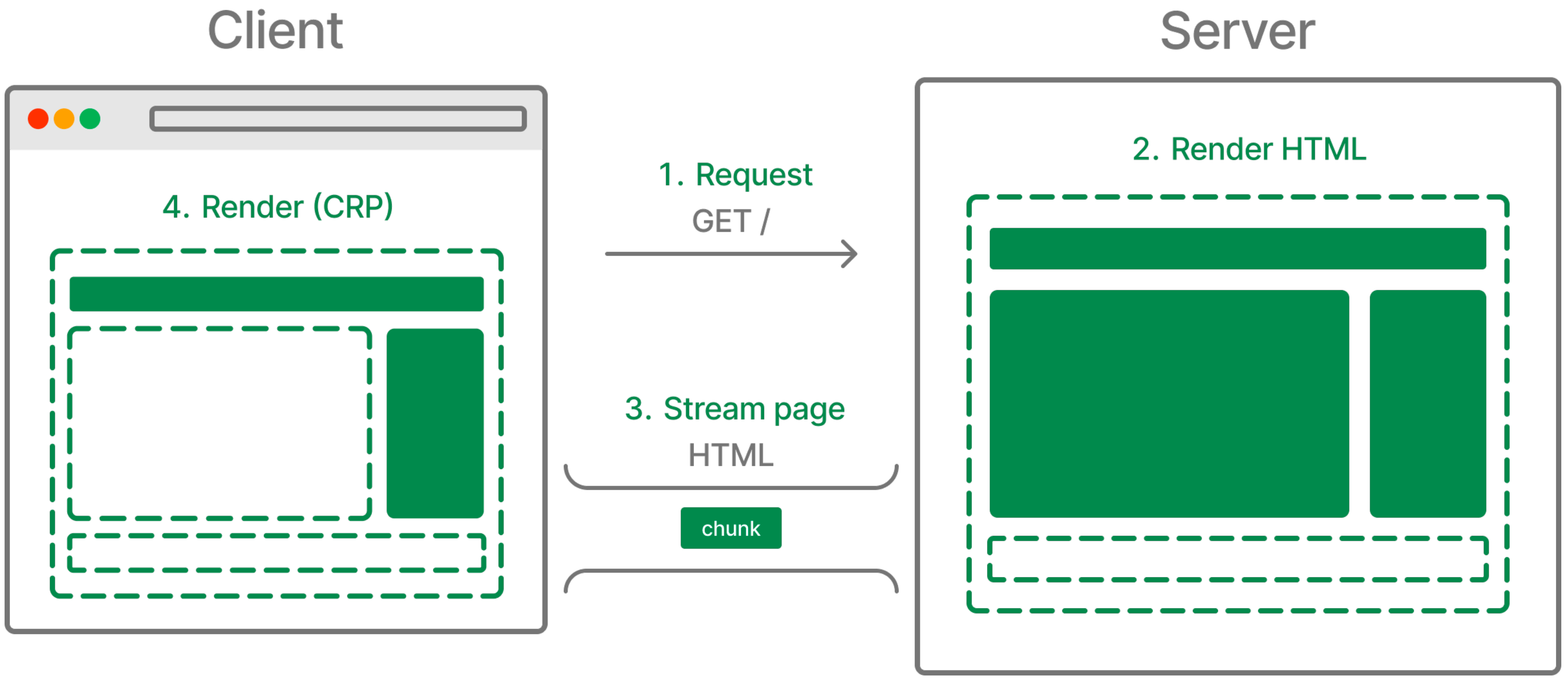


How does Streaming with Suspense work?

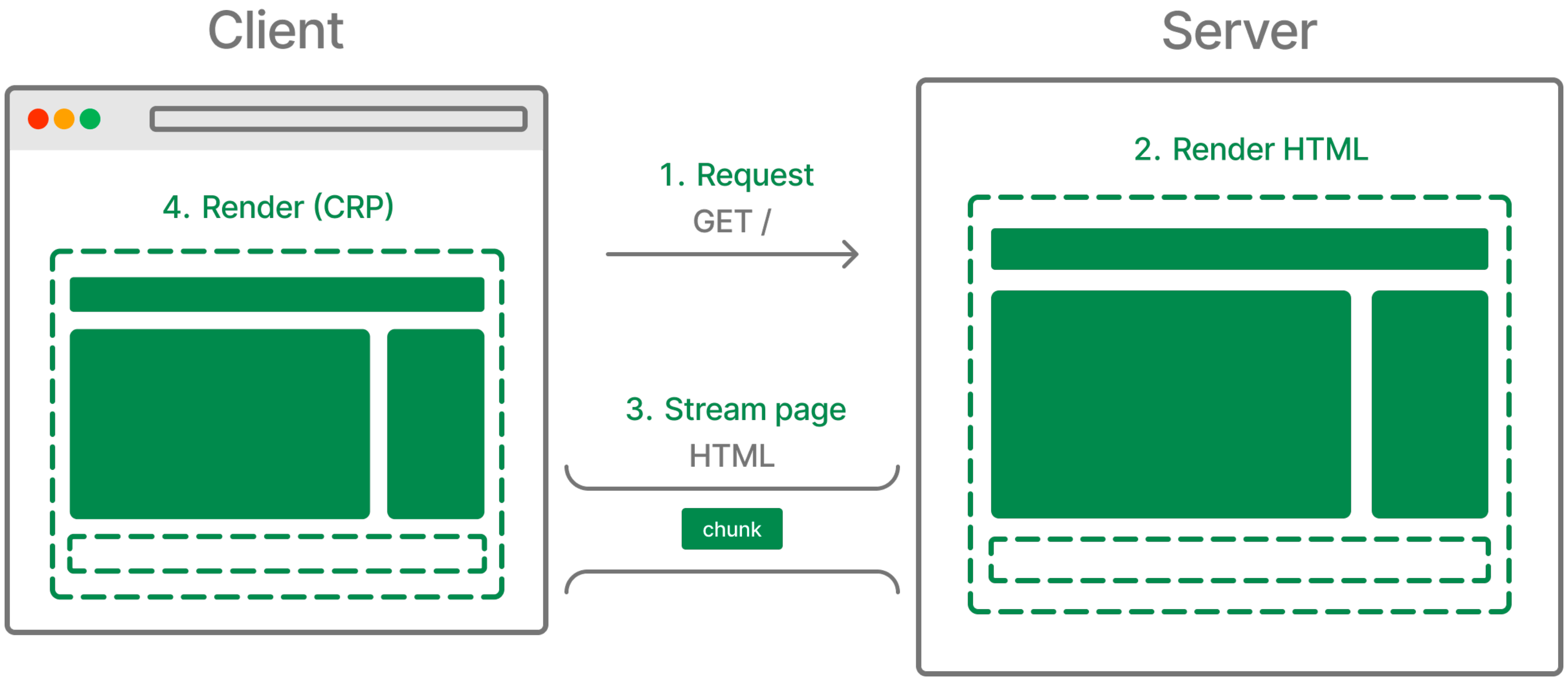


St.Galler Kantonalbank

How does Streaming with Suspense work?

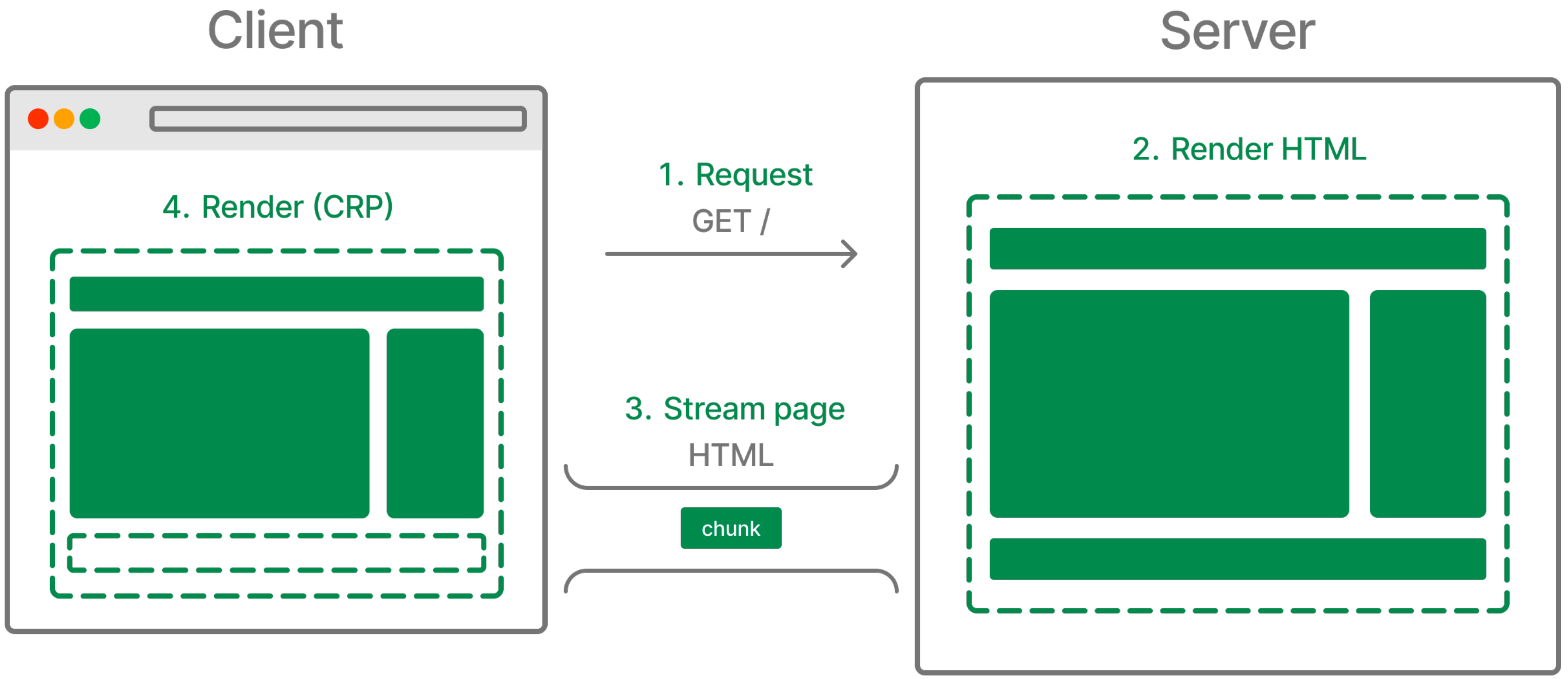


How does Streaming with Suspense work?

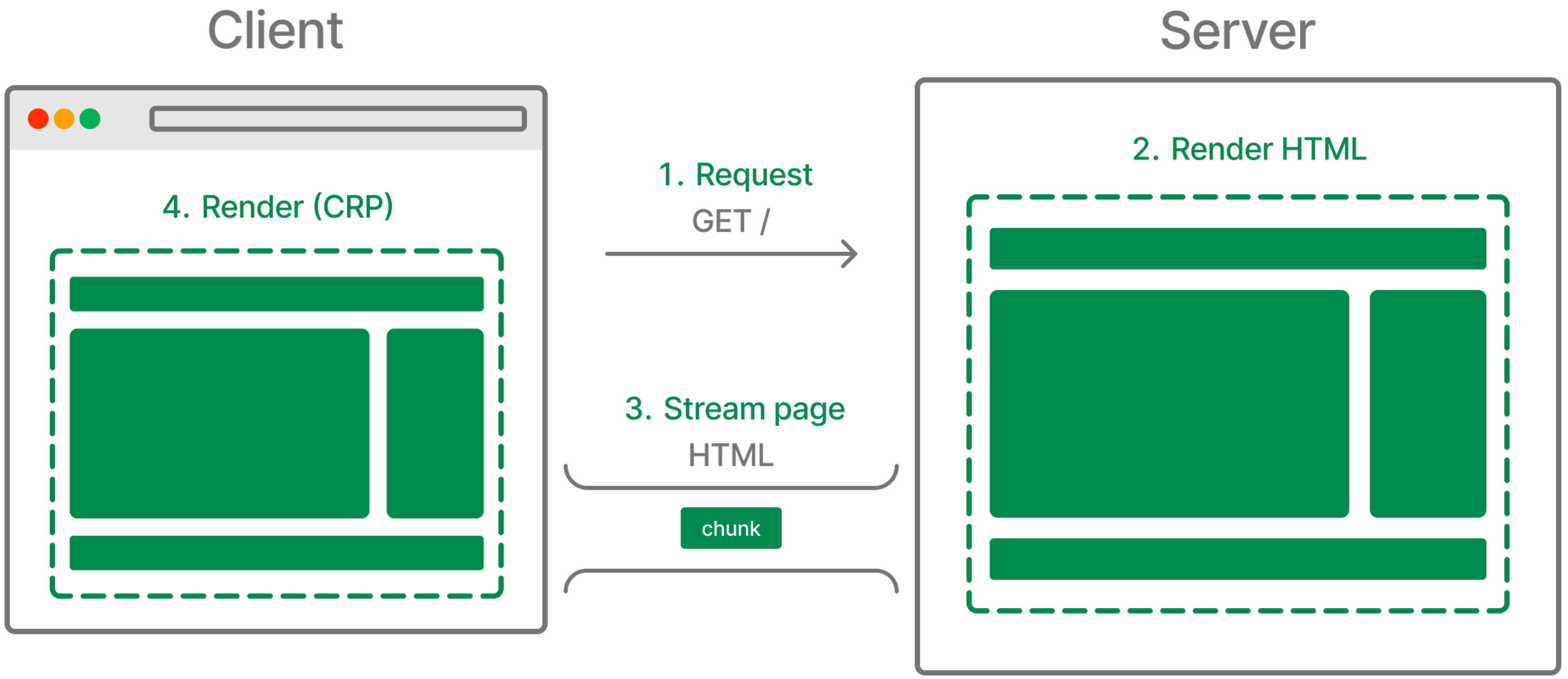


St.Galler Kantonalbank

How does Streaming with Suspense work?

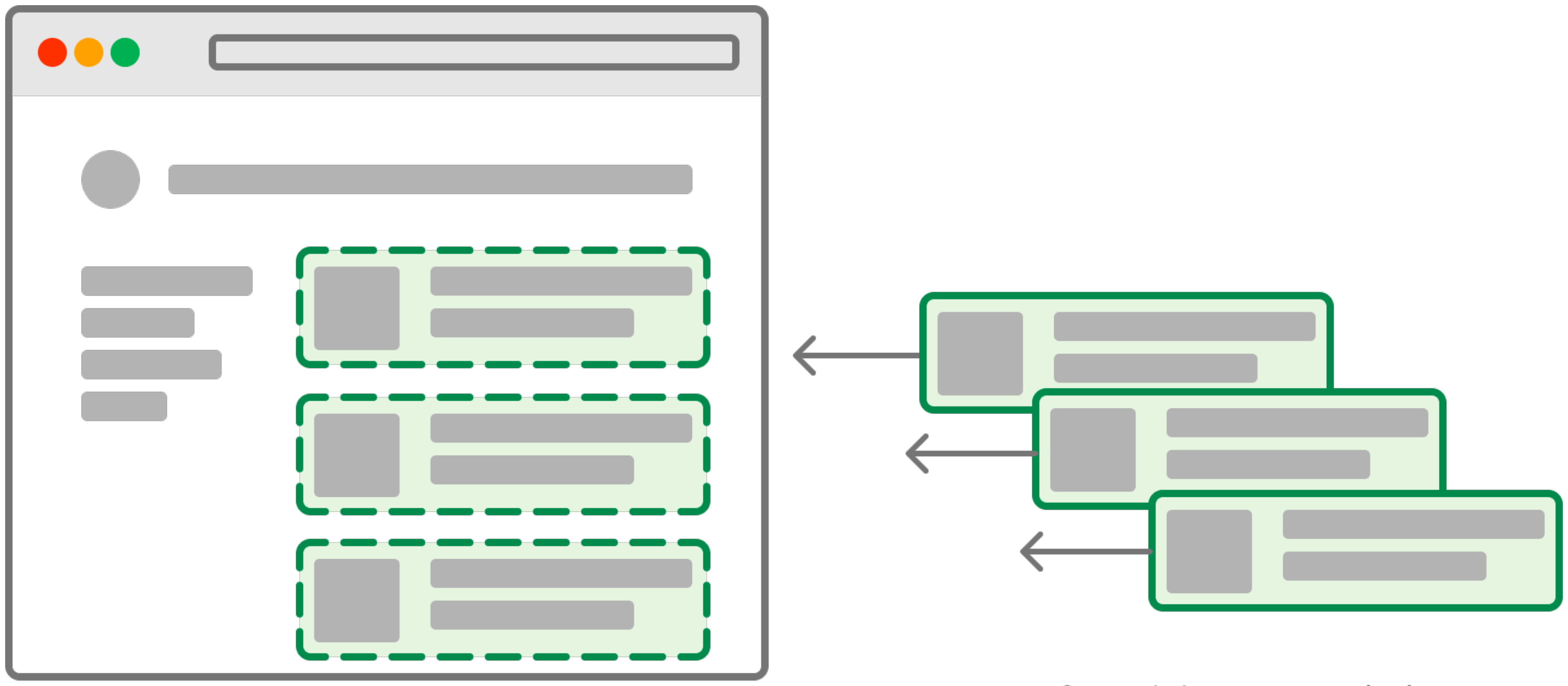


How does Streaming with Suspense work?



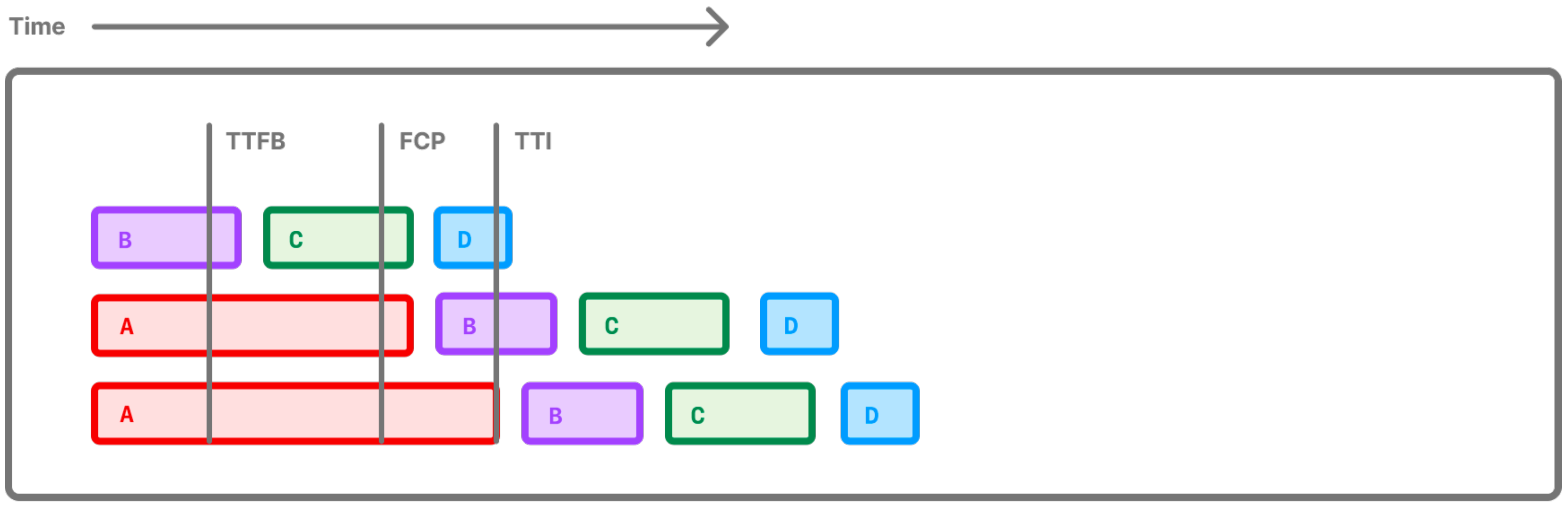
St.Galler Kantonalbank

How does Streaming with Suspense work?



Partial content with loading state

How does Streaming with Suspense work?



A	Fetching data on server
B	Rendering HTML on server
C	Loading Code on the client
D	Hydrating

TTFB	Time To First Byte
FCP	First Contentful Paint
TTI	Time To Interactive

St.Galler Kantonalbank

Benefits of Streaming with Suspense

- Performance improvement over SSR
  - Reduced Time To First Byte (TTFB) and First Contentful Paint (FCP)
  - Improves Time To Interactive (TTI), especially on slower devices
- Better backpressure handling
  - Responsive websites even under challenging conditions

When should Streaming with Suspense be used?

- When an easy transition from SSR into Streaming with Suspense is possible
- When a better Time To First Byte (TTFB), First Contentful Paint (FCP) or Time To Interactive (TTI) is required
- When server components are already in use, but there are difficulties in processing all incoming requests

Thank you for your attention.

Jan Kott
Full-Stack Engineer
jan.kott@sgkb.ch
Telefon +41 71 231 43 03

Meine erste Bank. St.Galler Kantonalbank