Rendering Patterns

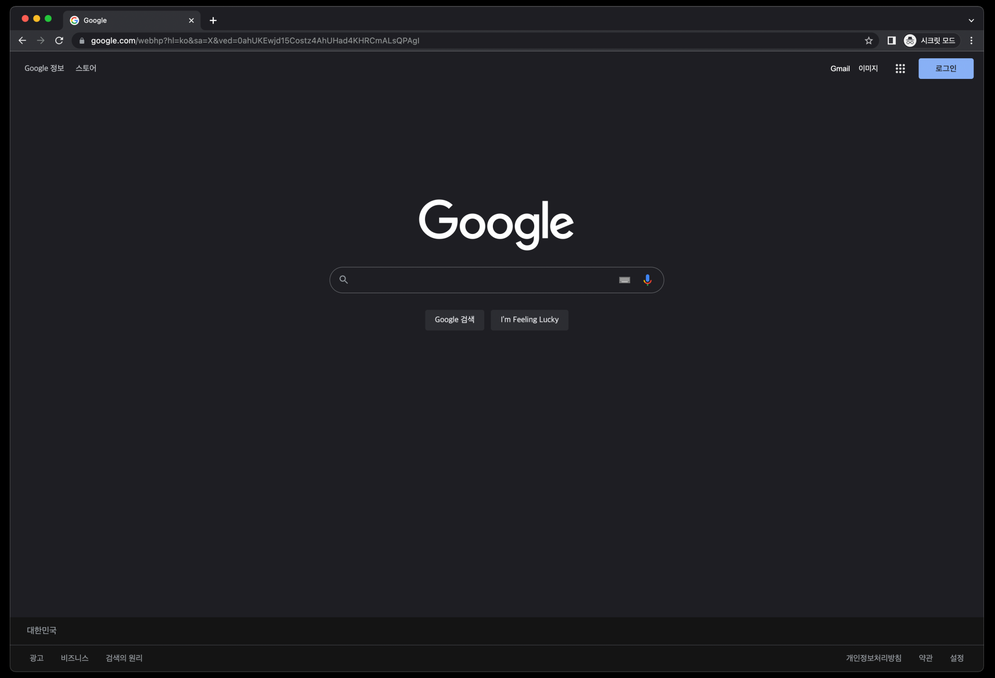
방슐랭 가이드 🏠
소프트웨어 마에스트로

HYUNJIN LEE

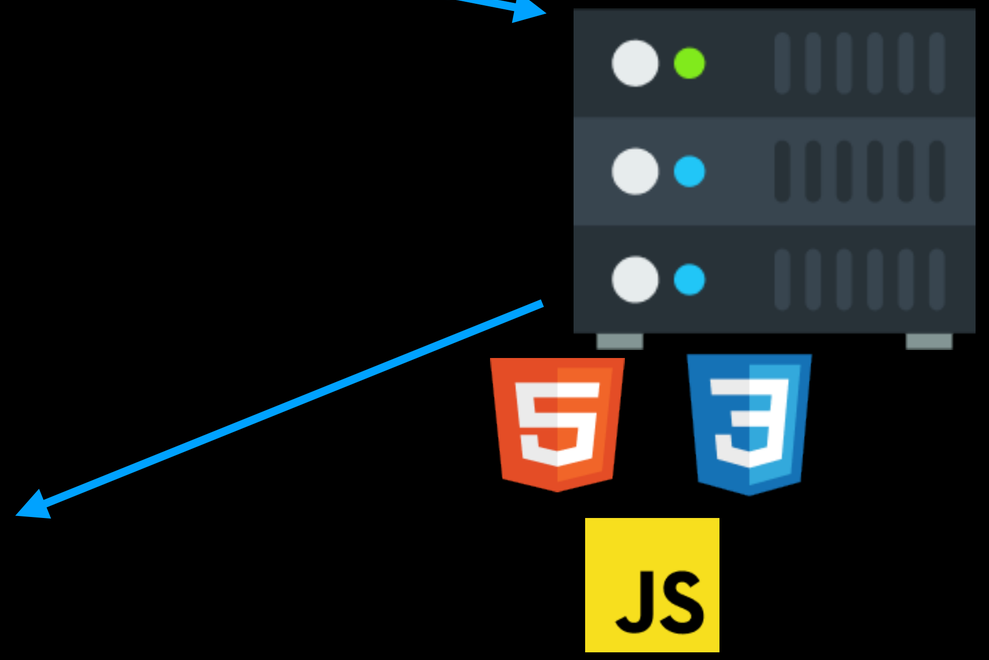
hyunjinee

Rendering

브라우저 화면에 웹 페이지를 그리는 것

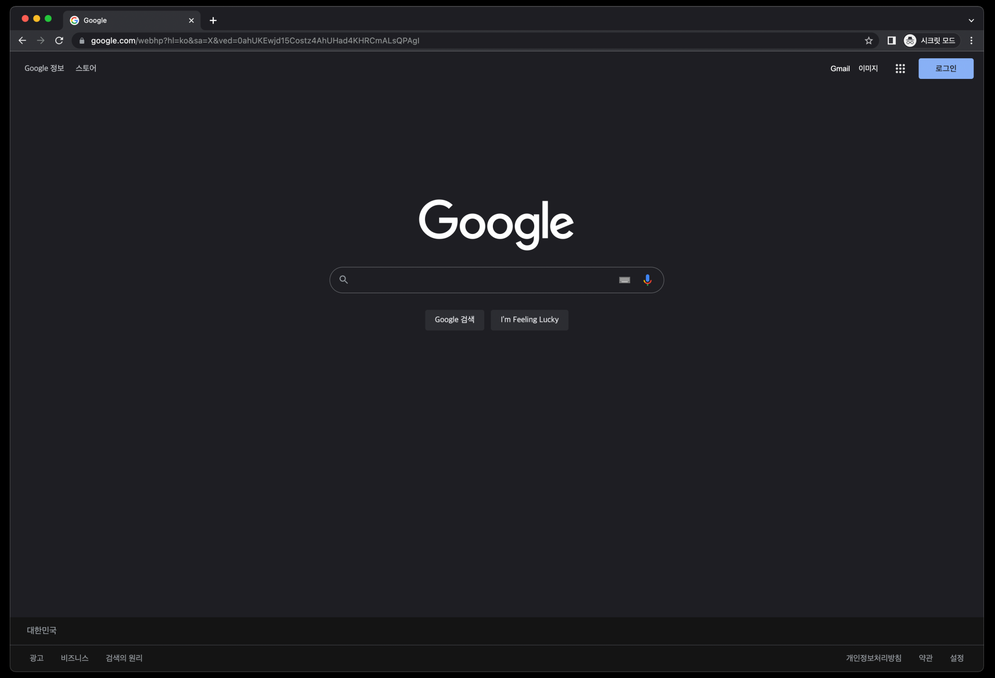


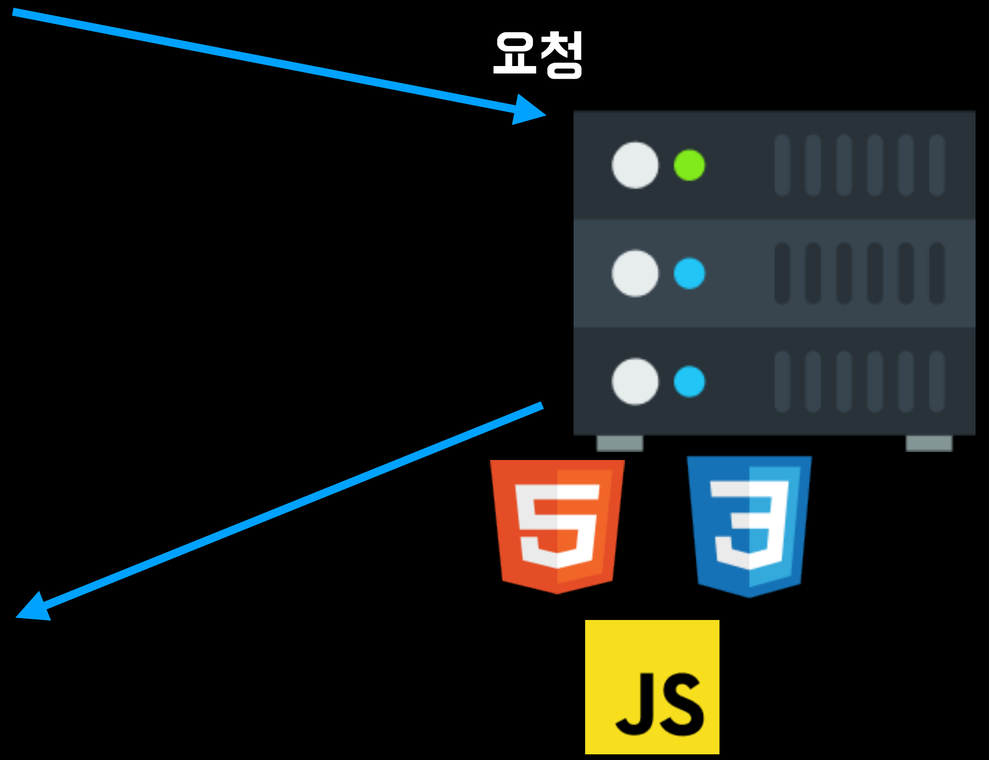
요청



Rendering

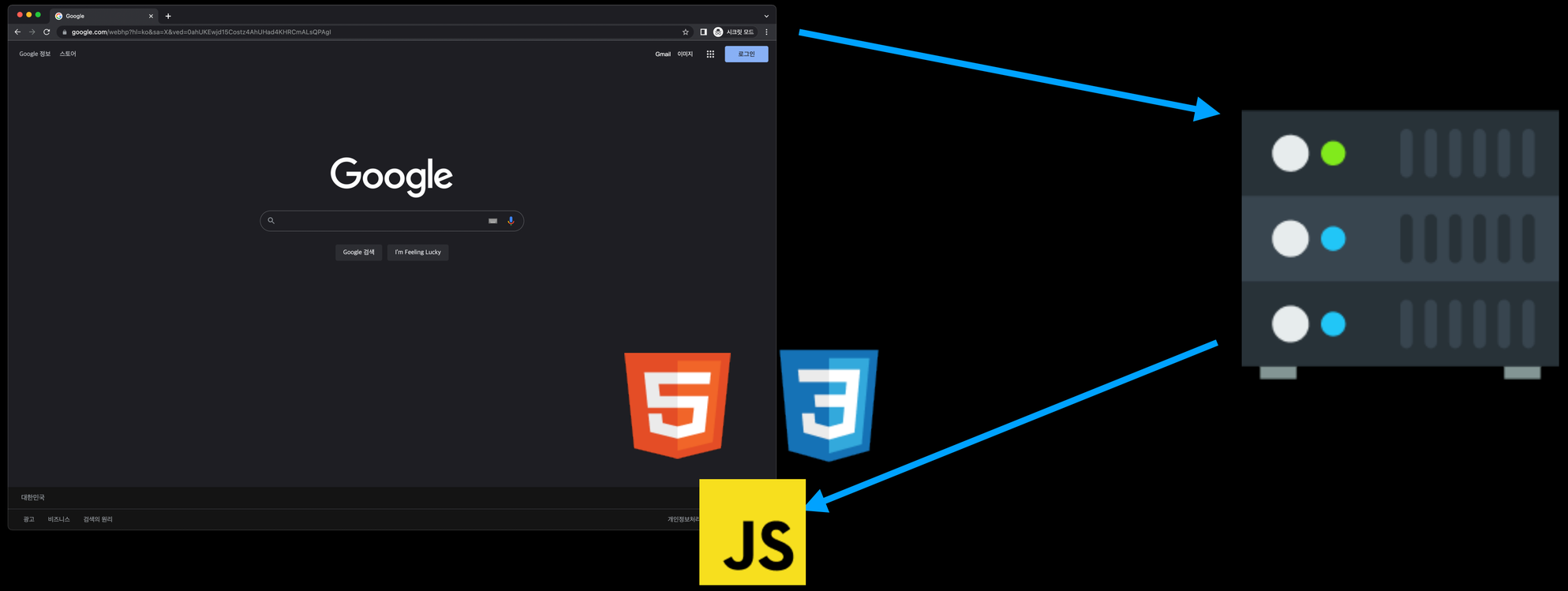
브라우저 화면에 웹 페이지를 그리는 것





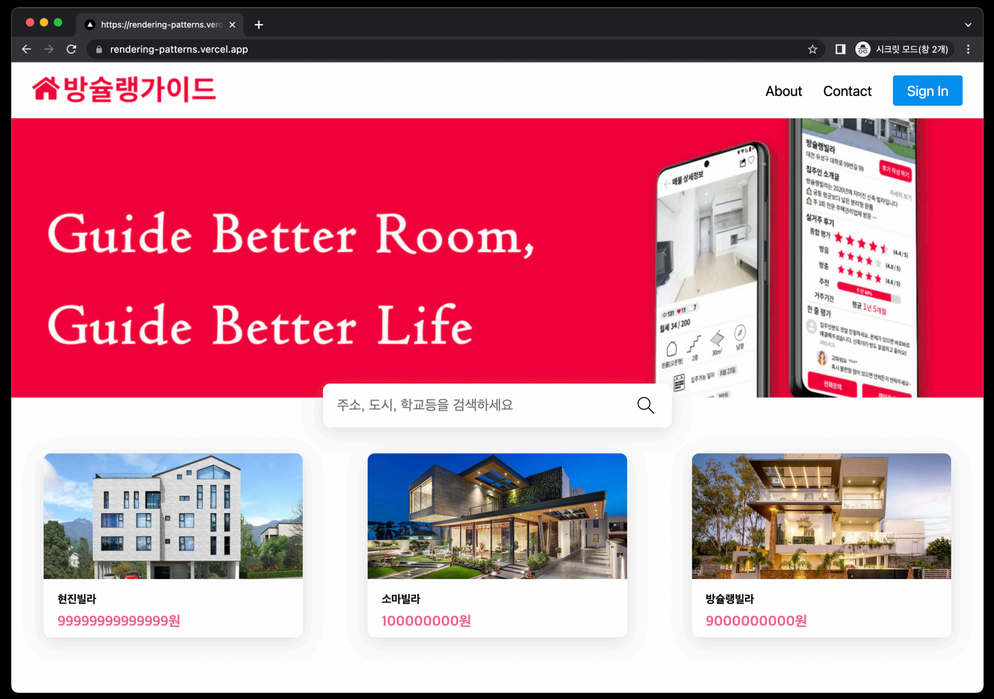
Rendering

브라우저 화면에 웹 페이지를 그리는 것

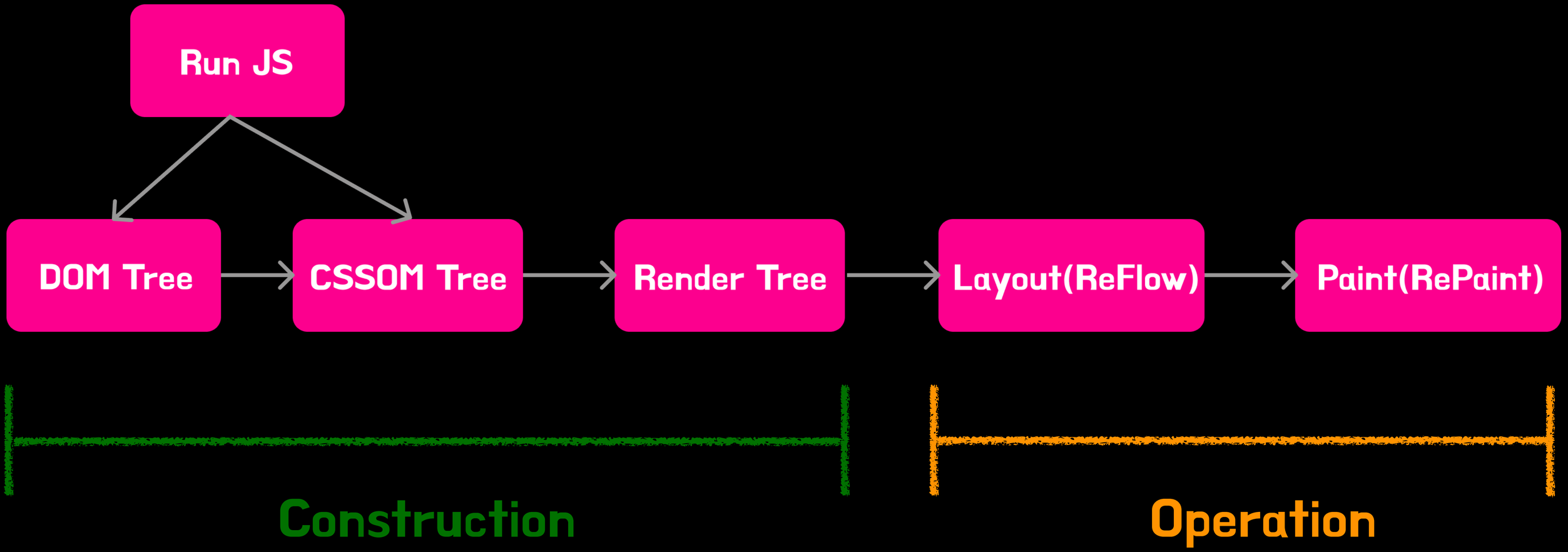


Rendering

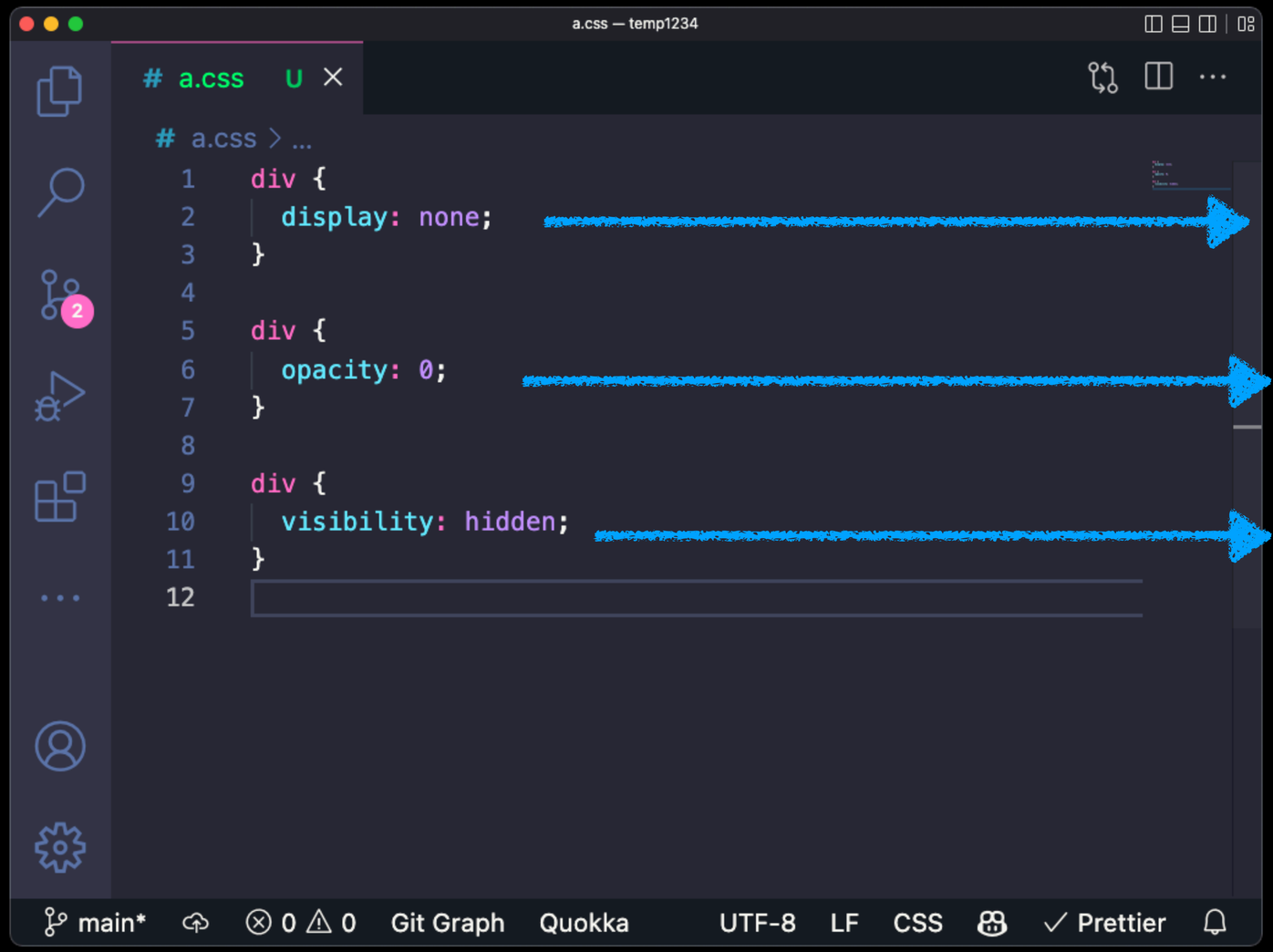
브라우저 화면에 웹 페이지를 그리는 것

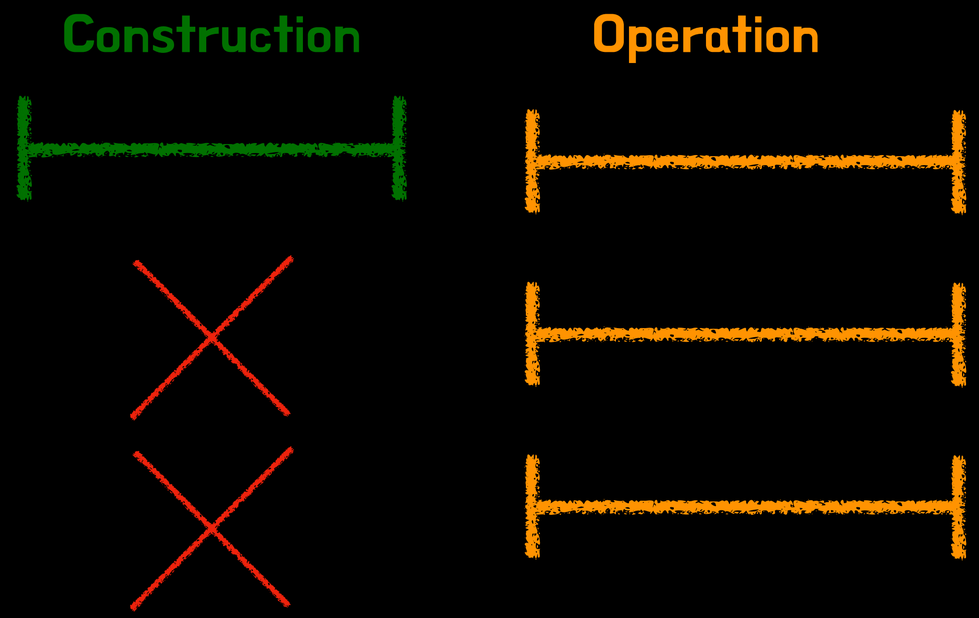


Critical Rendering Path

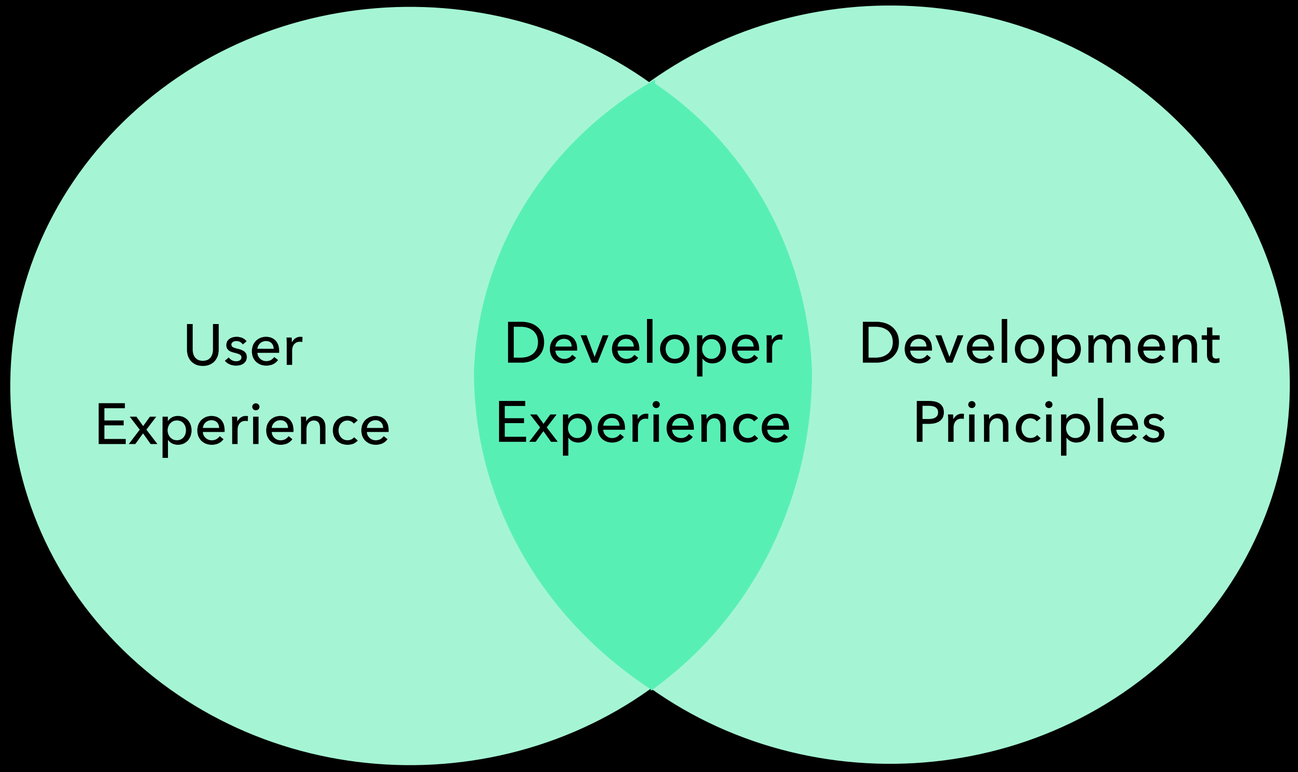


Critical Rendering Path





Rendering -> UX, DX



올바른 Rendering Pattern?

SEO

Web Performance

Core Web Vitals

TTFB

Time To First Byte

FCP

First Contentful Paint

LCP

Largest Contentful Paint

TTI

Time To Interactive

CLS

Cumulative Layout Shift

FID

First Input Delay

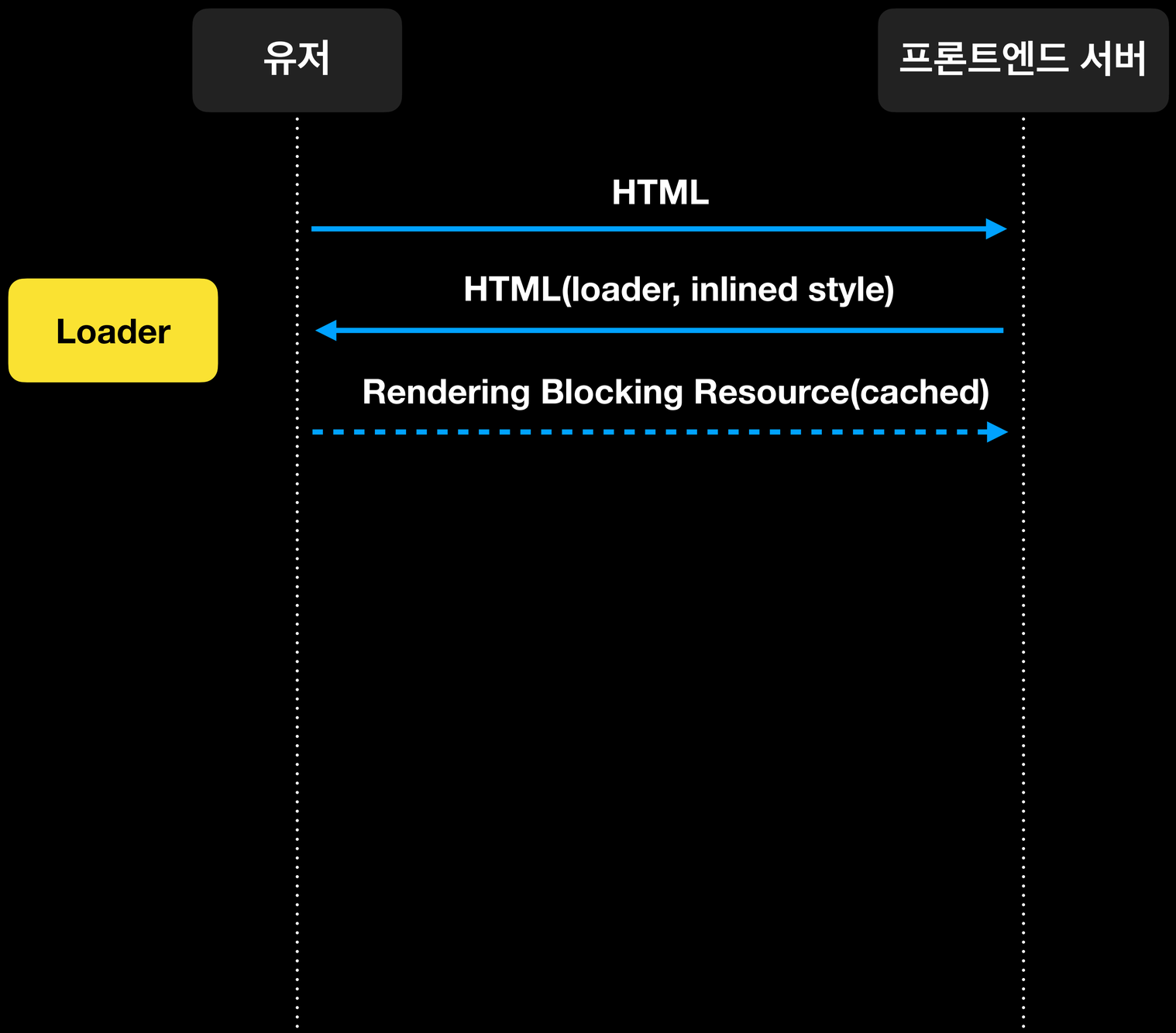
Rendering Patterns

SSG CSR SSR ISR

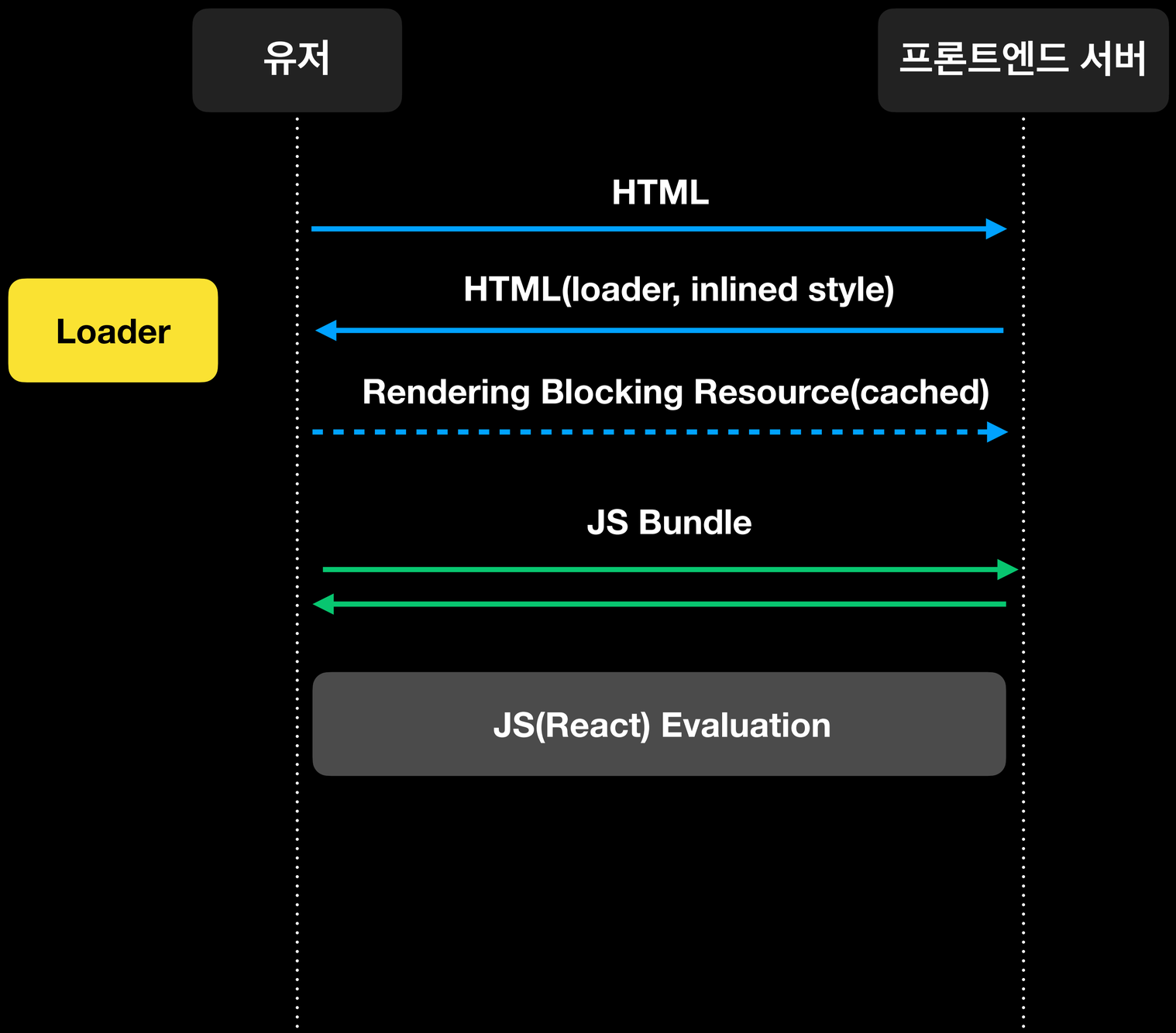
Static Rendering Streaming

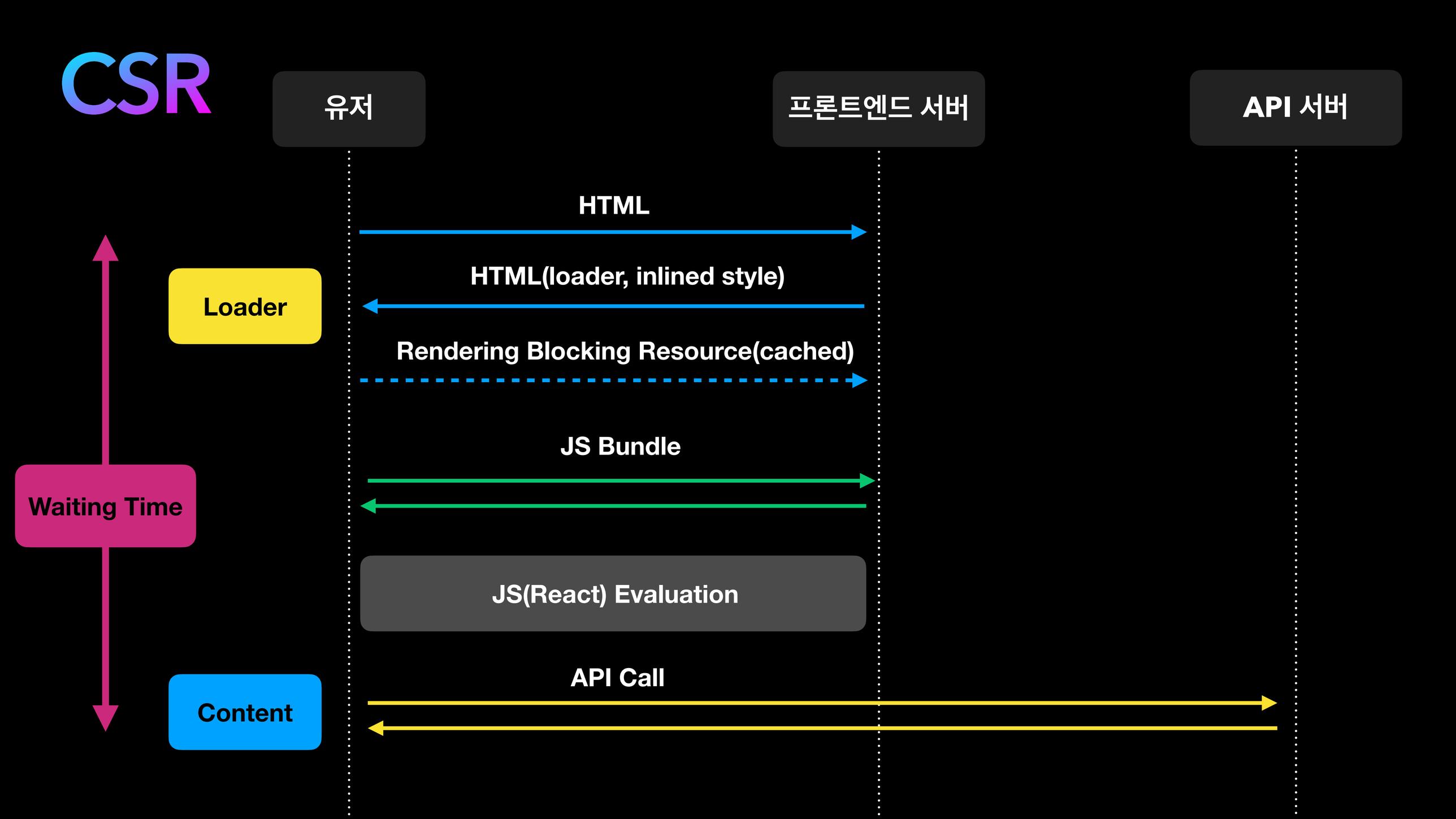
React Server Components

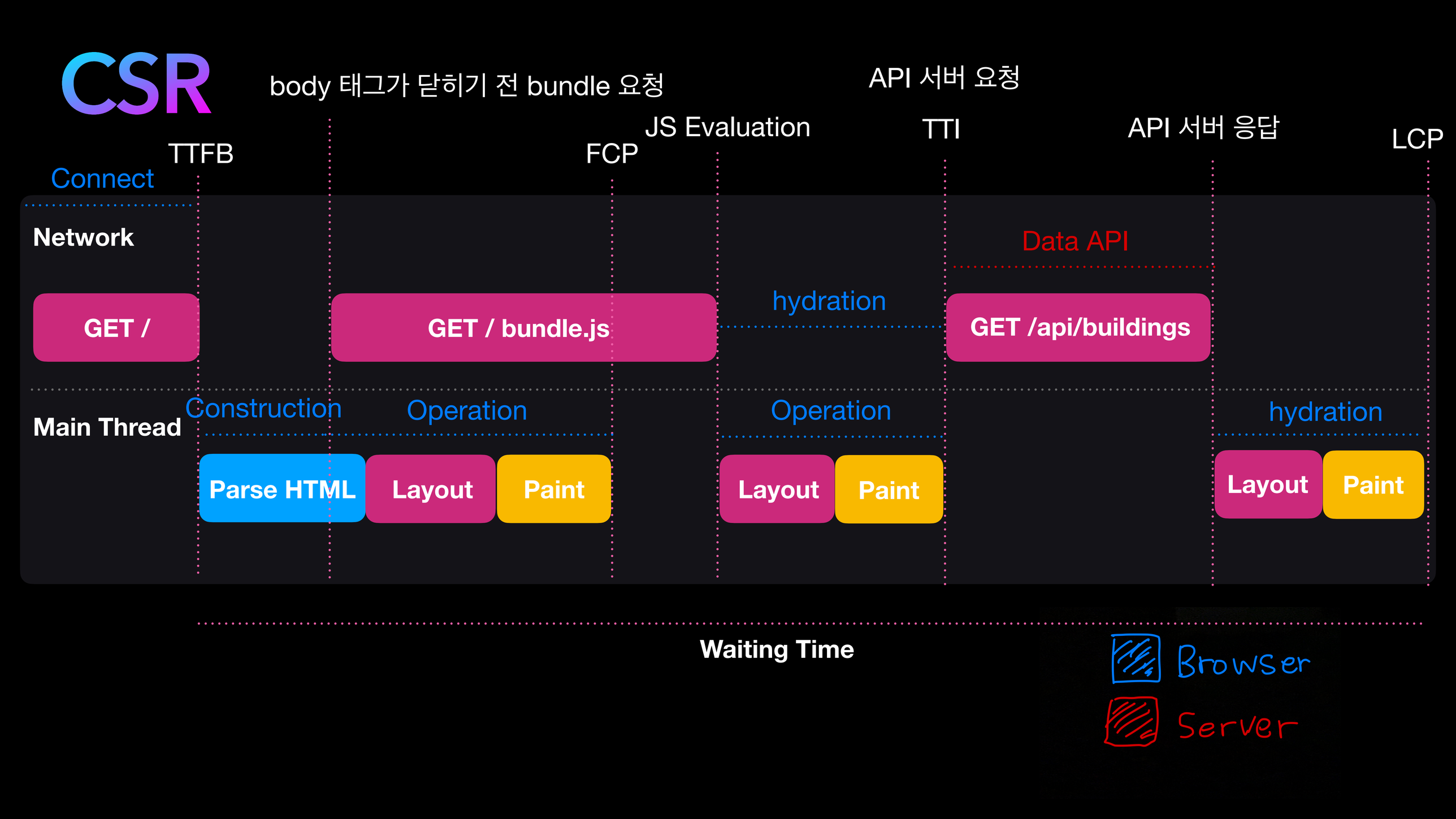
CSR


유저
프론트엔드 서버
HTML
HTML(loader, inlined style)
Loader
Rendering Blocking Resource(cached)

CSR


유저
프론트엔드 서버
HTML
HTML(loader, inlined style)
Loader
Rendering Blocking Resource(cached)
JS Bundle
JS(React) Evaluation


CSR
유저
프론트엔드 서버
API 서버
HTML
HTML(loader, inlined style)
Loader
Rendering Blocking Resource(cached)
JS Bundle
Waiting Time
JS(React) Evaluation
API Call
Content

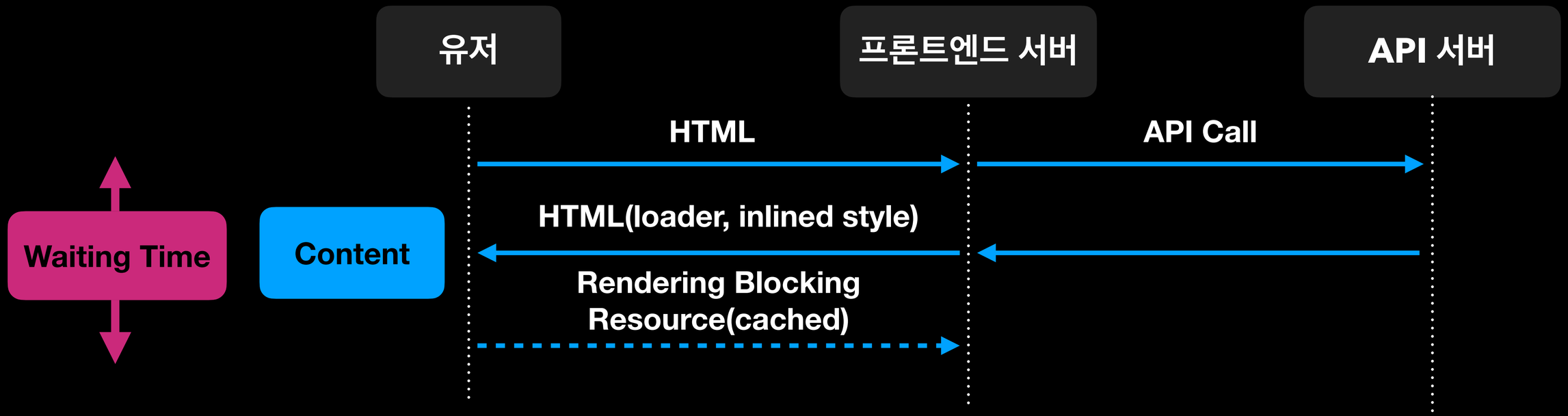

CSR
body 태그가 닫히기 전 bundle 요청
API 서버 요청
JS Evaluation
TTI
API 서버 응답
LCP
TTFB
FCP
Connect
Network
Data API
GET /
GET / bundle.js
hydration
GET /api/buildings
Main Thread
Construction
Operation
Operation
hydration
Parse HTML
Layout
Paint
Layout
Paint
Layout
Paint
Waiting Time
Browser
Server

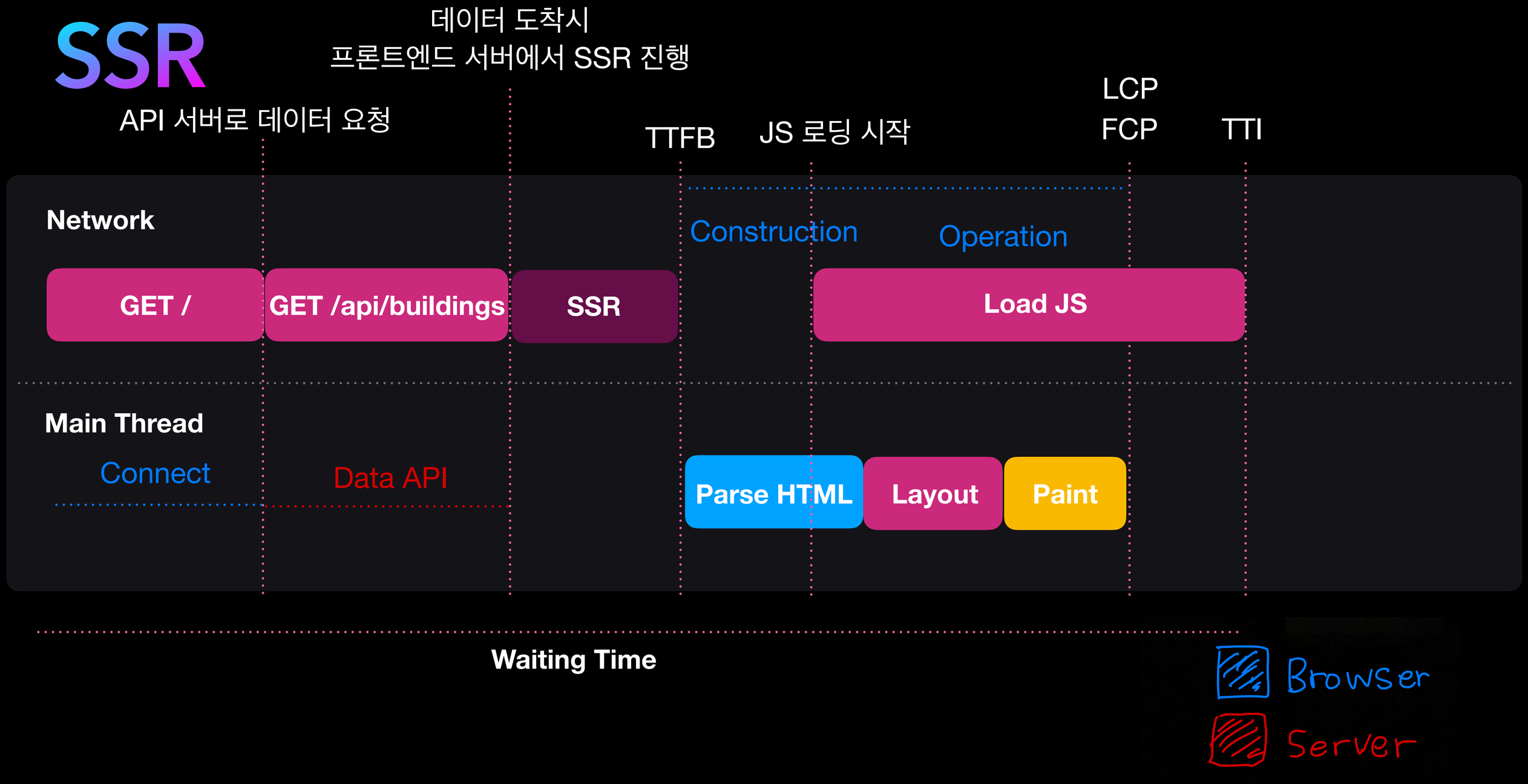
hydration

https://rendering-patterns.vercel.app/

hydrate

SSR


유저
프론트엔드 서버
API 서버
HTML
API Call
HTML(loader, inlined style)
Rendering Blocking
Resource(cached)
Waiting Time
Content


SSR
데이터 도착시
프론트엔드 서버에서 SSR 진행
API 서버로 데이터 요청
TTFB
JS 로딩 시작
LCP
FCP
TTI
Network
Construction
Operation
GET /
GET /api/buildings
SSR
Load JS
Main Thread
Connect
Data API
Parse HTML
Layout
Paint
Waiting Time
Browser
Server

CSR + SSR

API 서버로 데이터 요청

LCP
FCP
TTFB
JS 로딩 시작
TTI

Network

Construction
Operation

GET /
GET /api/buildings
SSR
Load JS

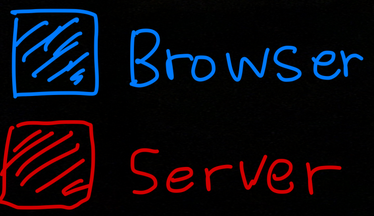
hydration

Main Thread

Connect
Data API

Parse HTML
Layout
Paint

데이터 도착시
프론트엔드 서버에서 SSR 진행




NEXT
.JS

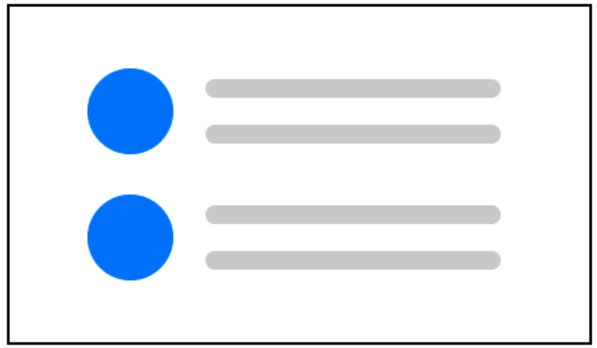
Pre-Rendering

SSG

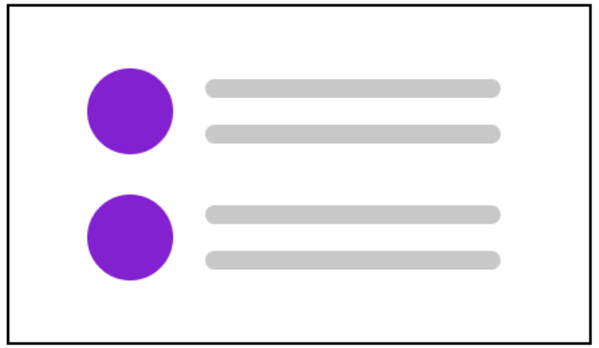
SSR

NEXT.JS

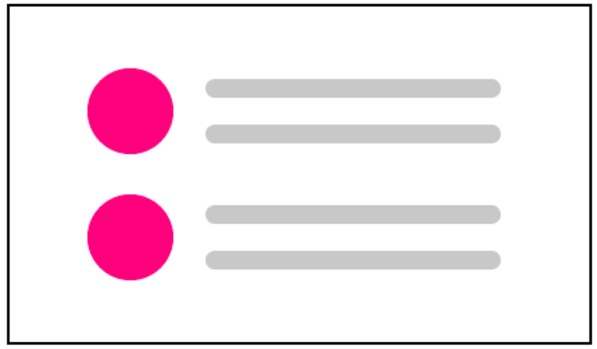
Page A: Static Generation



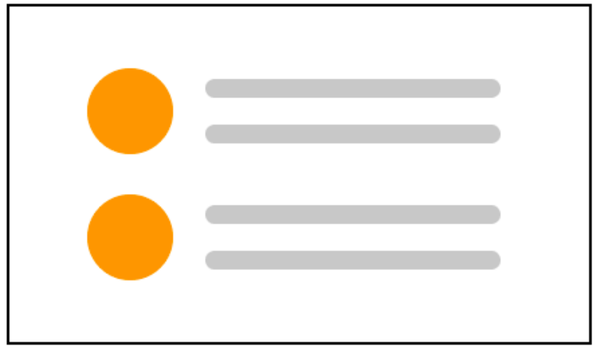
Page B: Server-side Rendering



Page C: Server-side Rendering



Page D: Static Generation



You can choose which pre-rendering form to use for each page.

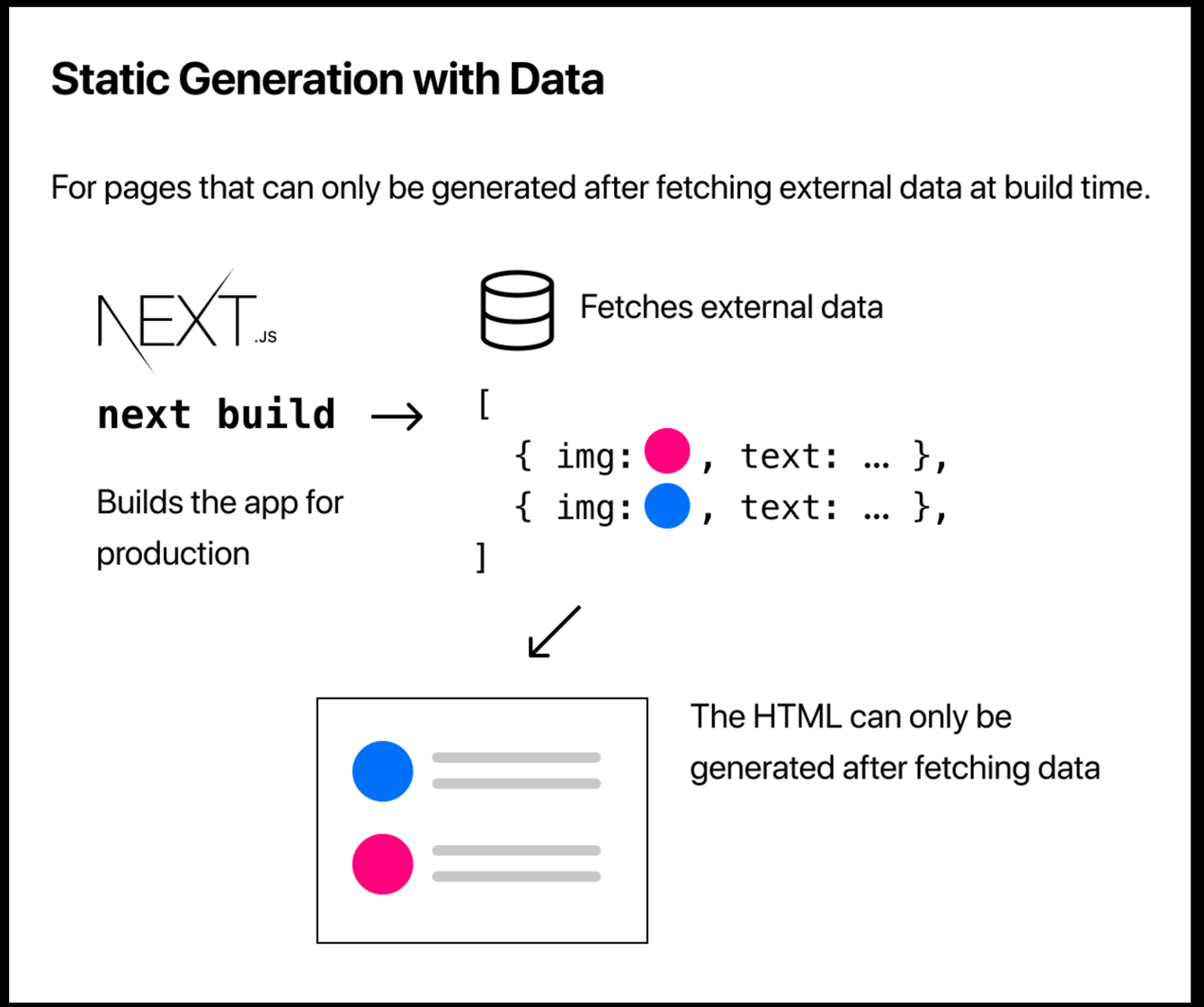
SSG(Static Site Generation)

빌드 타임에 HTML 페이지 생성

빌드 타임에만 API 서버 접근

빌드 타임에 컨텐츠를 만들기 때문에 변하는 데이터에 취약

CDN에 캐시 -> 렌더링 속도 up



SSG(Static Site Generation) + CSR

빌드 타임에 HTML 페이지 생성

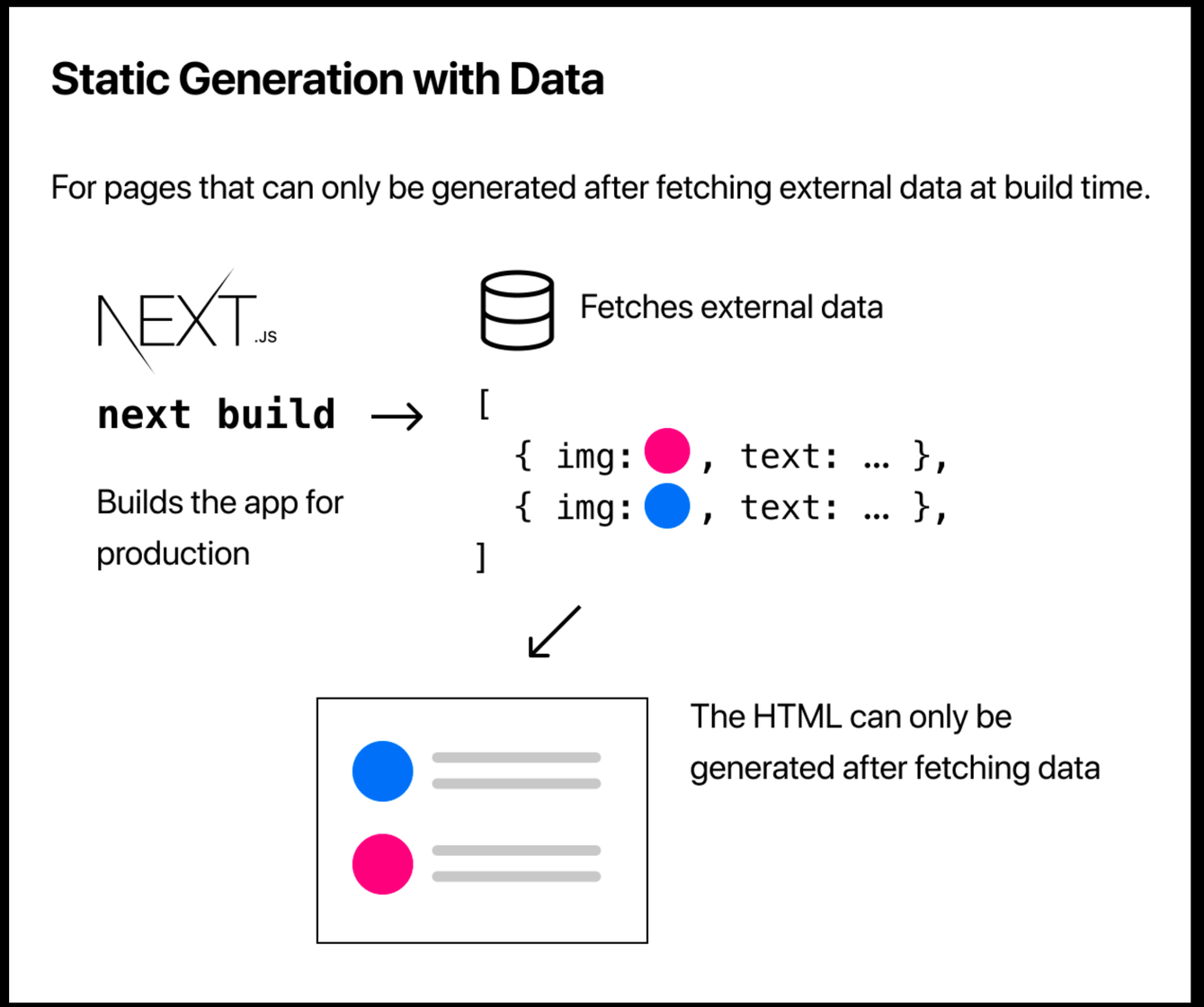
빌드 타임에만 API 서버 접근

빌드 타임에 컨텐츠를 만들기 때문에 변하는 데이터에 취약

CDN에 캐시 -> 렌더링 속도 up

CSR로 구성된 부분은 pre-render 하지 않음

CSR의 장점 + SSG의 장점



ISR (Incremental Static ReGeneration)

빌드 타임에 HTML 페이지 생성

빌드 타임에만 API 서버 접근

~~빌드 타임에 컨텐츠를 만들기 때문에~~
~~변하는 데이터에 취약~~

CDN에 캐시 -> 렌더링 속도 up

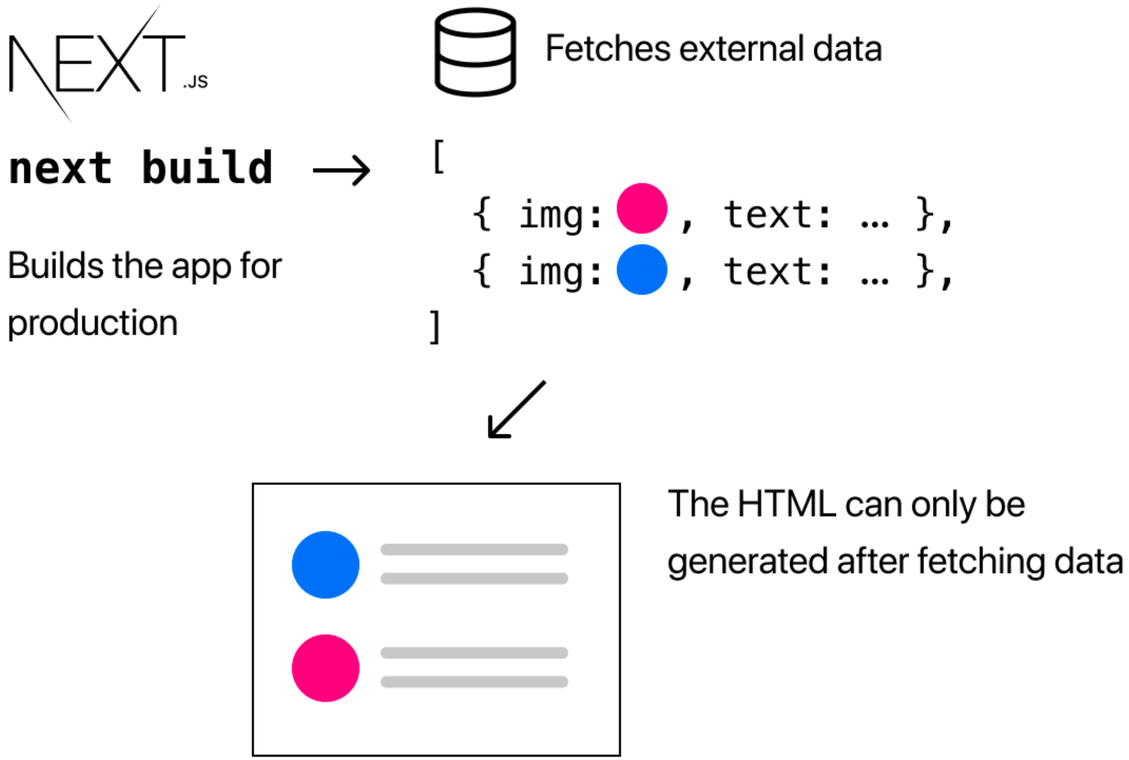
revalidation 옵션 추가

일정 주기마다 데이터의 최신 여부를 검사

업데이트된 데이터로 페이지를 다시 생성

Static Generation with Data

For pages that can only be generated after fetching external data at build time.



ISR + CSR

빌드 타임에 HTML 페이지 생성

빌드 타임에만 API 서버 접근

~~빌드 타임에 컨텐츠를 만들기 때문에~~
~~변하는 데이터에 취약~~

CDN에 캐시 -> 렌더링 속도 up

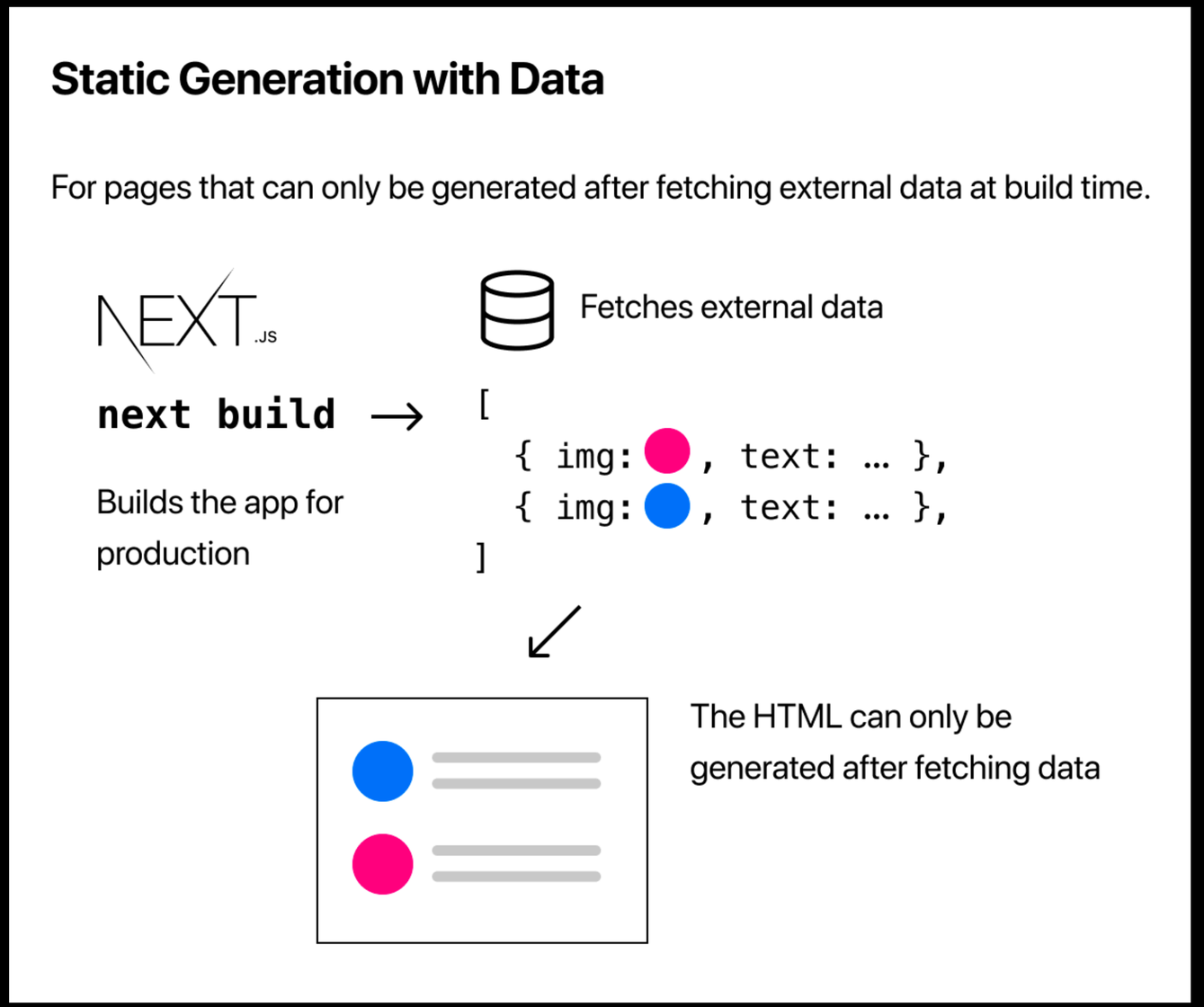
revalidation 옵션 추가

일정 주기마다 데이터의 최신 여부를 검사

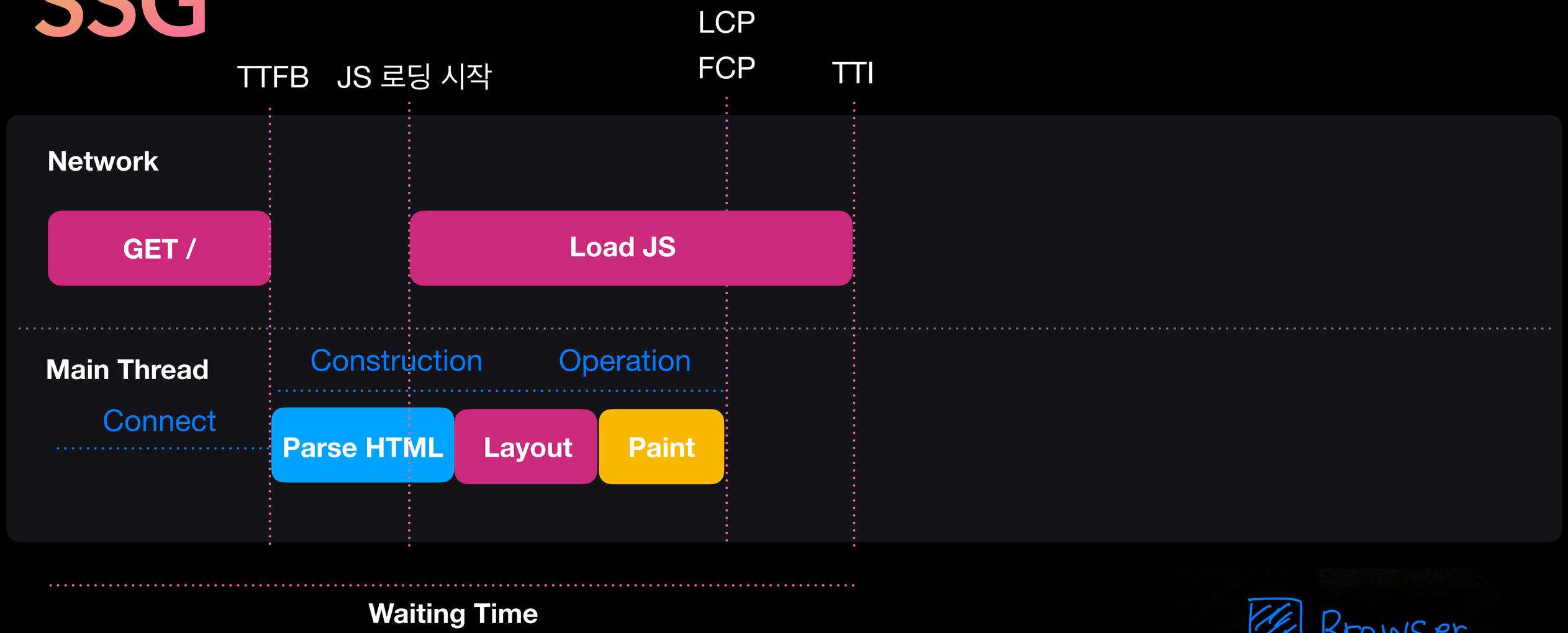
업데이트된 데이터로 페이지를 다시 생성

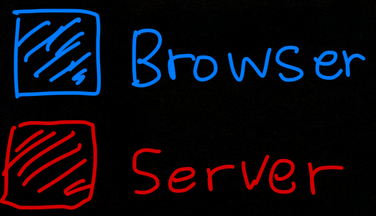
CSR로 구성된 부분은 pre-render 하지 않음

CSR의 장점 + SSG의 장점



SSG


LCP
FCP
TTFB
JS 로딩 시작
TTI
Network
GET /
Load JS
Main Thread
Construction
Operation
Connect
Parse HTML
Layout
Paint
Waiting Time


Browser
Server

SSR

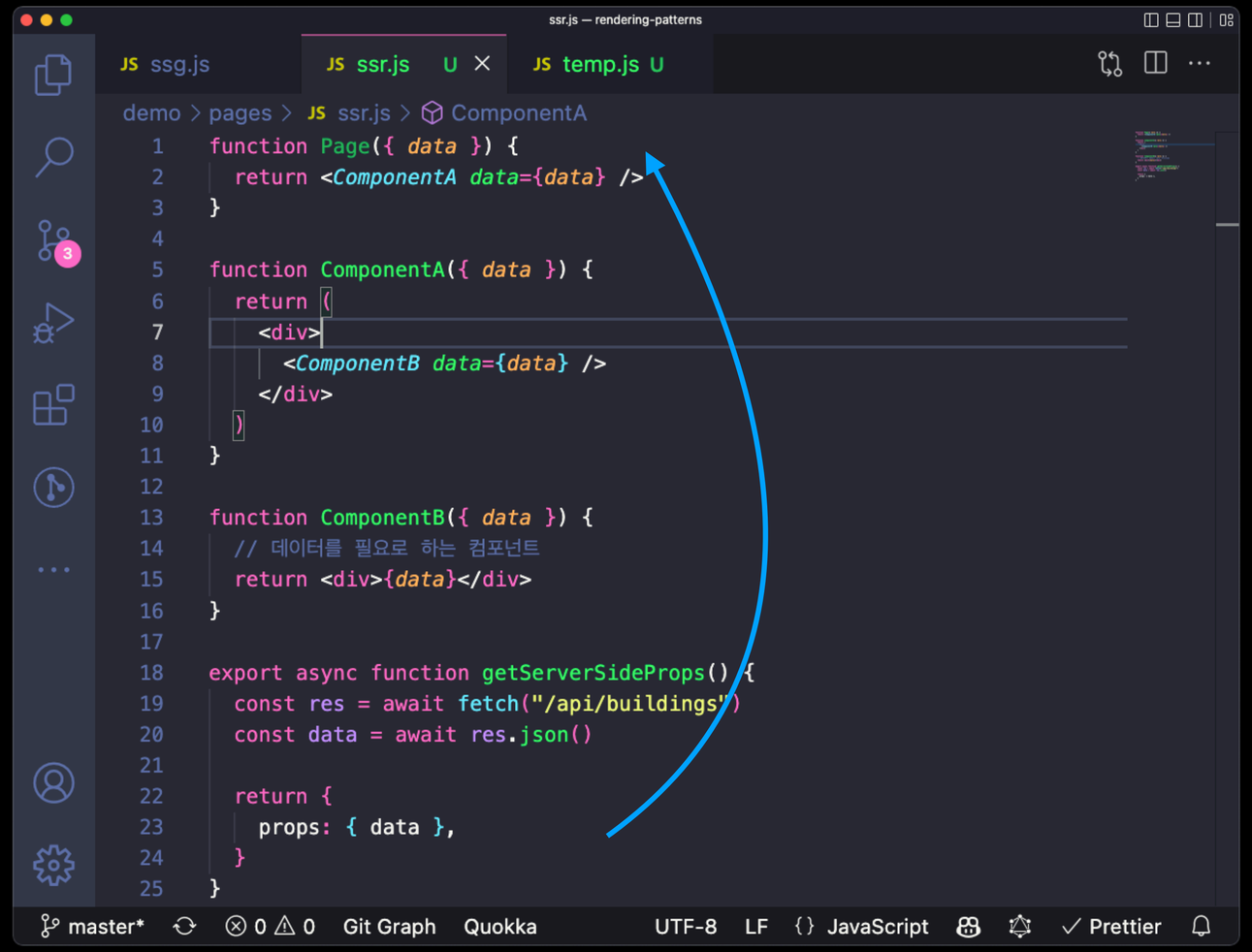
Page 단위 data fetching의 문제

Props Drilling

Data를 필요로하는 컴포넌트는 B

A는 전달하는 매개체로만 사용

관심사의 분리가 되지 않음



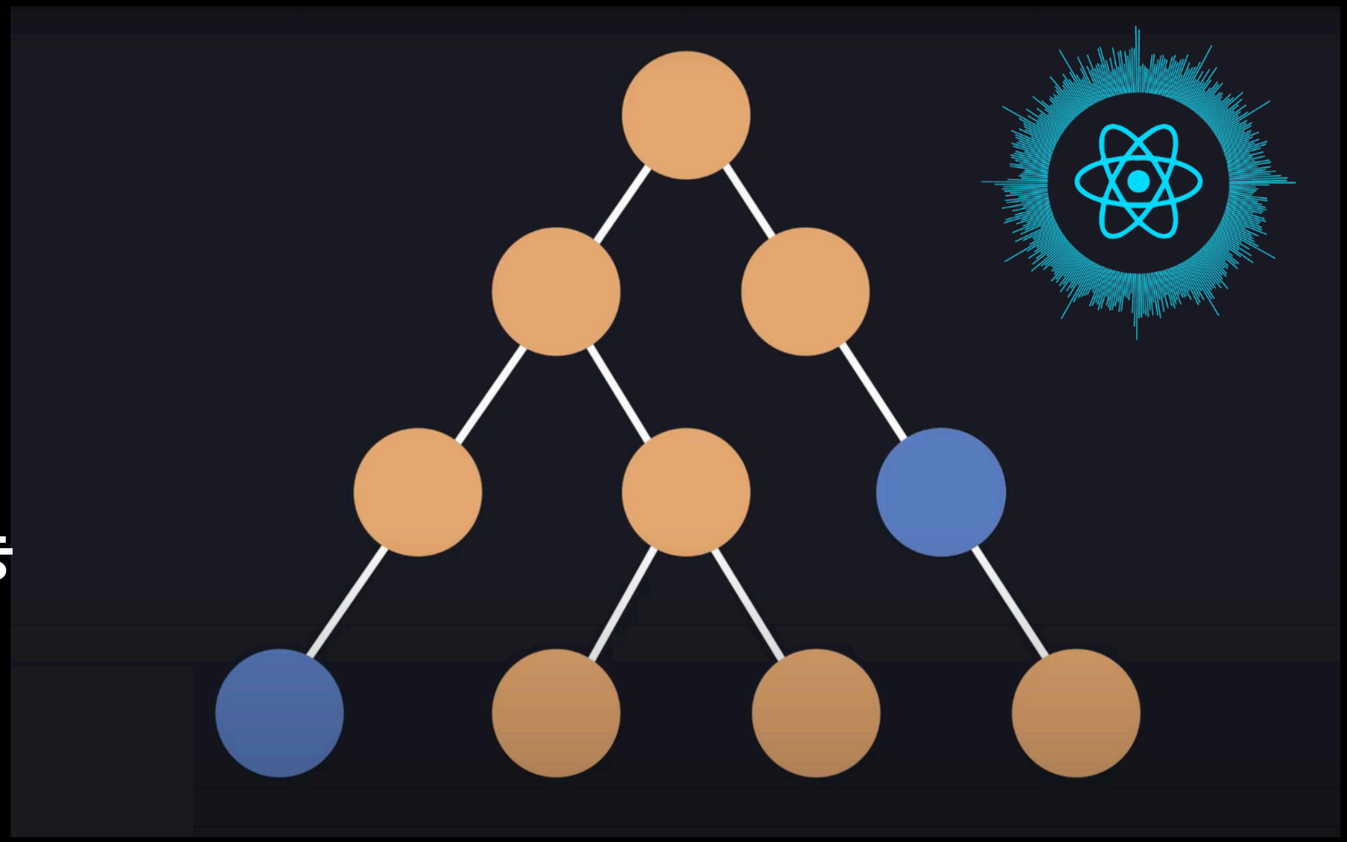
React Server Components

서버에서 data fetching

데이터베이스에 직접 접근

서버컴포넌트는 클라이언트로 전송되는
번들에 포함되지 않음

클라이언트 상태를 유지하며 refetch 가능



UX

TTFB

FCP

LCP

TTI

CLS

FID

DX

빌드 시간 단축

적은 서버 비용

확장 가능한 설계

엄청난 아이디어

Develop

배포(Ship)

Question

렌더링을 언제 고려해야 할까요?

엄청난 아이디어

Develop

배포(Ship)

엄청난 아이디어

Develop

Rendering Patterns

SSG CSR SSR ISR

Static Rendering Streaming

React Server Components

배포(Ship)

Rendering Patterns

꼿

Q & A

참고자료

Advanced Rendering Patterns. Lydia Hallie

JavaScript 프레임워크에서 효율적인 하이드레이션이 어려운 이유. eunbinn

Pre-rendering. 이현진

웹은 어떻게 발전했는가. 이현진

100년을 아껴준 SSR 이야기. toss

What Is DX?. Albert Cavalcante