



COMMUNITY DAY

SEOUL

React.js 개발자를 위한
SSR 앱 개발 및 배포하기

원지혁 / AUSG

발표자 소개



Jee-Hyuk Won

AUSG

- Full-Stack Web Developer
- React.js 개발을 주로 해요 
- GraphQL Korea Organizer
 - facebook [GraphQL Korea](#)
- AWSKRUG GraphQL Group Organizer
 - slack [#graphql](#)

“ Vue.js로 Front-end 개발을 처음 시작해,
현재는 TypeScript와 React.js, GraphQL을 주로 사용합니다.
높은 수준의 퀄리티를 보장할 수 있는
소프트웨어 설계에 대해 끊임없이 고민하고 있습니다.

목차

- 배경
- SSR(Server-side Rendering)이 뭘까요?
- Next.js 소개
- Store Hydration을 통한 데이터 주입
- Elastic Beanstalk과 ElastiCache를 활용해 어플리케이션 배포하기
- CloudFront를 이용한 정적 자원 제공 가속화
- Demo

배경



배경

외부의 API 서버



API 서비스는 클라이언트와 완전히 분리되어 있음

API 서버의 인증 방식



- API 요청 할 때,
Authorization 헤더에 Access Token을 포함
- Access Token이 만료된 경우 Refresh Token를 이용해
Access Token을 재발급
- Access Token의 만료 시간: 30분
- Refresh Token의 만료 시간: 14일

API 명세



- POST /auth/login
 - username, password를 넘겨주면, Access Token, Refresh Token을 반환함
- POST /auth/refresh
 - Refresh Token을 넘겨주면, 새로운 Access Token과 Refresh Token을 반환함
- GET /me
 - Access Token을 읽고 DB에서 내 정보를 가져옴

배경

API 명세



API

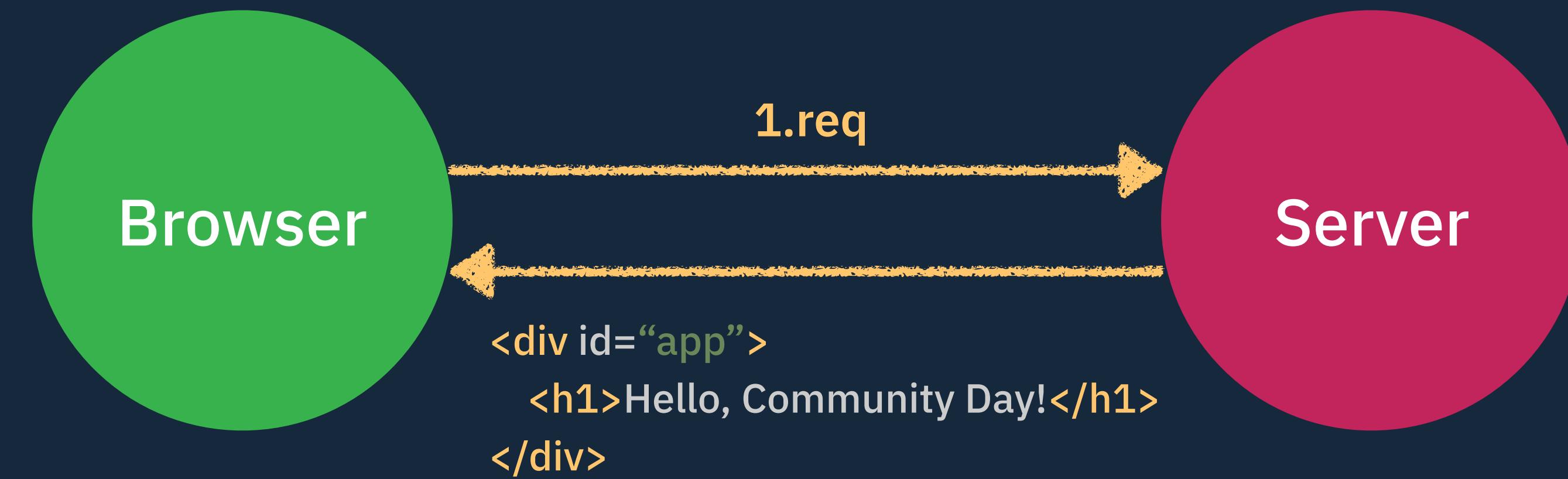
<https://react-ssr-demo-api.now.sh>

SSR이 뭘까요?



SSR이 뭘까요?

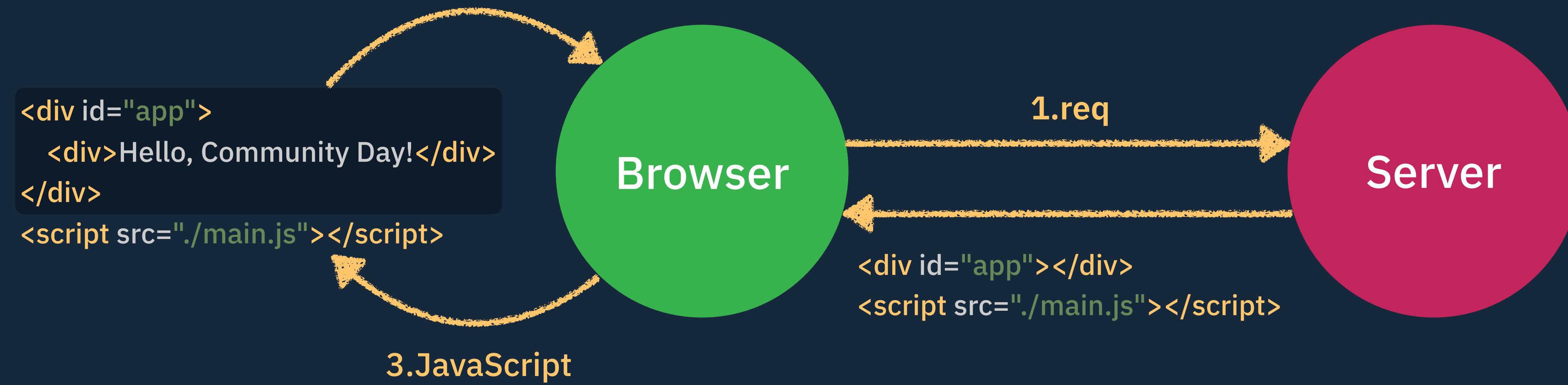
전통적인 Web Framework



기존의 전통적인 Web Framework에서는 서버에서 HTML 문서를 작성함

SSR이 뭘까요?

Client-side Rendering



더 향상된 사용자 경험을 위해 Angular, React.js 등
클라이언트에서 HTML 문서를 조작하는 JavaScript Framework들이 등장

SSR이 뭘까요?

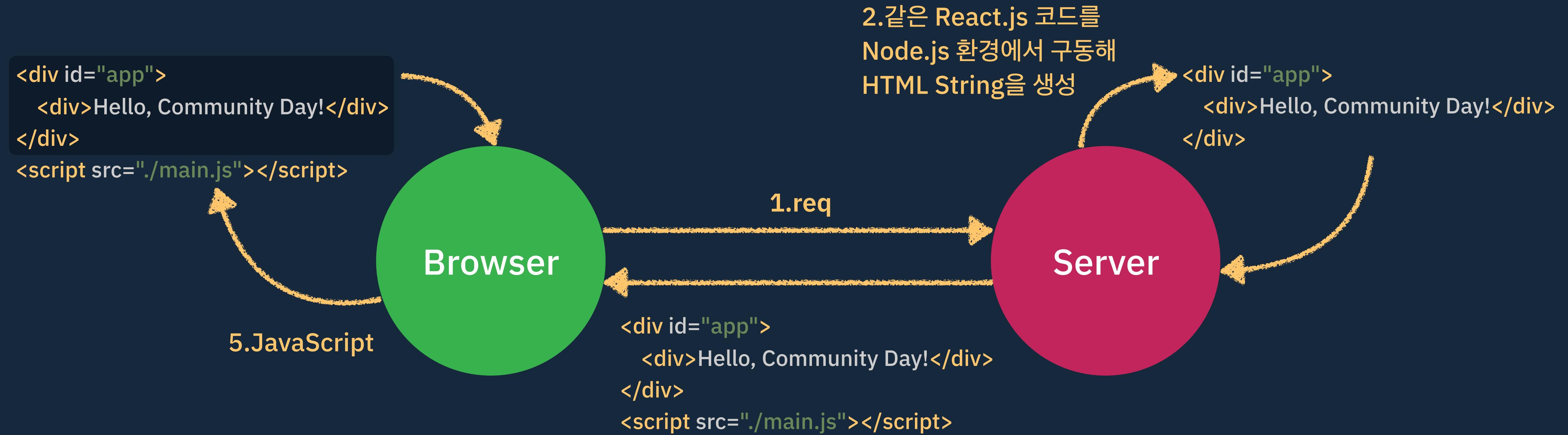
Client-side Rendering의 한계



하지만 SEO(검색엔진 최적화), 초기 렌더링 시간 단축 등의 목적으로
서버에서 HTML 문서를 만들어야 할 필요가 생김

SSR이 뭘까요?

SSR (Universal Rendering)



따라서 해당 문제를 해결하기 위해 서버에서 첫번째 HTML을 그리지만,
다음 렌더링부터는 클라이언트에서 이어나가는 방식을 SSR이라고 부름

SSR이 뭘까요?

간단 용어 소개

Hydration

“ Hydration은 React.js, Vue.js 등의 Client-side Framework가 서버에서 전송한 정적 문서를 데이터 변경에 반응할 수 있는 동적 형태로 변환하는 클라이언트 측 프로세스를 말합니다.

Isomorphic JavaScript

“ SSR 과정 중 한 JavaScript 코드가 Node.js와 Browser 환경에서 모두에서 동작하게 되는데, 이렇게 두 환경에서 모두에서 동작할 수 있는 JavaScript 코드를 Isomorphic JavaScript라고 합니다.

Next.js 소개

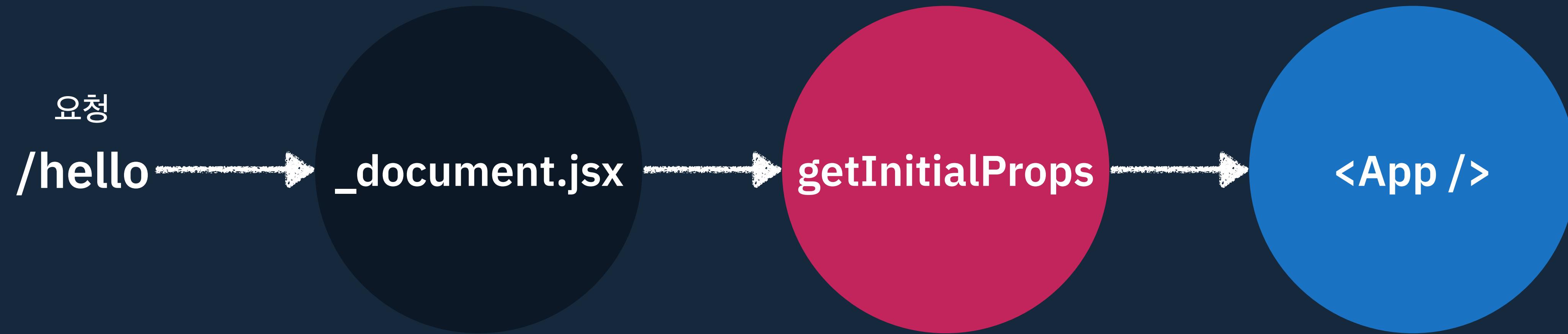


Hello, Next.js



- ZEIT에서 만든 오픈소스 프로젝트
- React.js 기반 SSR 프로젝트를 쉽게 시작할 수 있도록 도와줍니다.
- Twitch, Netflix, Tencent 등이 사용중

Hello, Next.js



요청이 발생하면 다음과 같은 흐름으로 HTML을 만듭니다

Store Hydration을 통한 데이터 주입



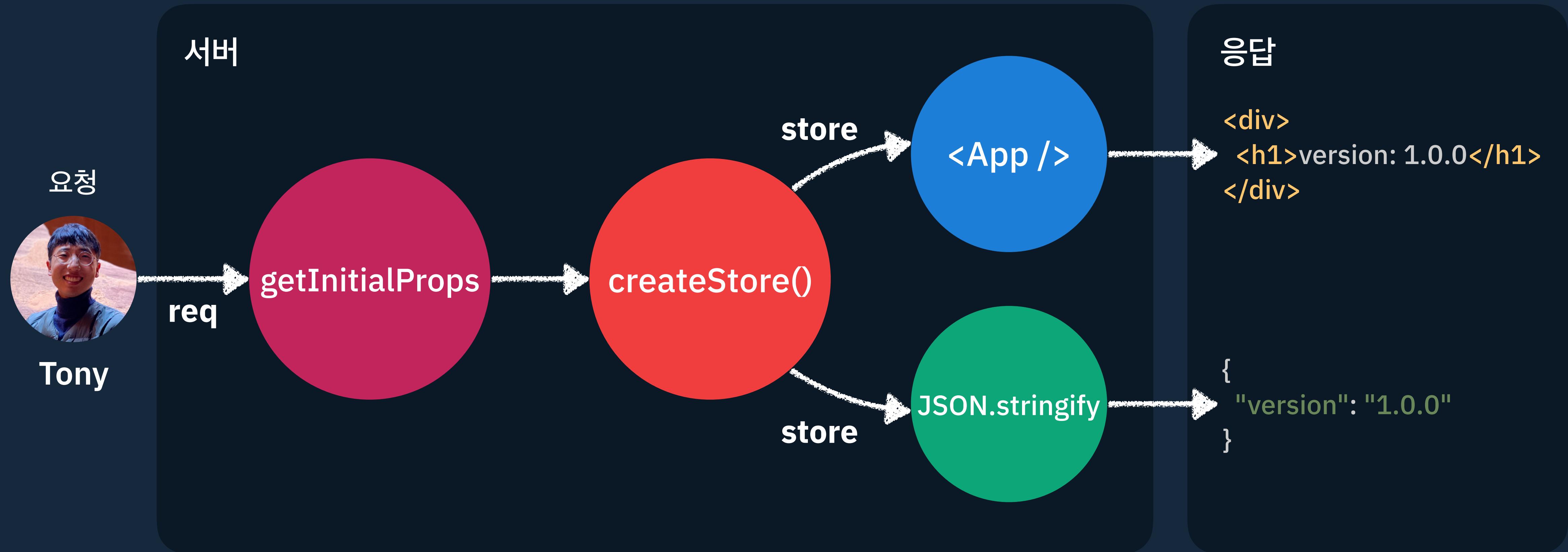
Store Hydration?

Hydration

“ Hydration은 React.js, Vue.js 등의 Client-side Framework가 서버에서 전송한 정적 문서를 데이터 변경에 반응할 수 있는 동적 형태로 변환하는 클라이언트 측 프로세스를 말합니다.

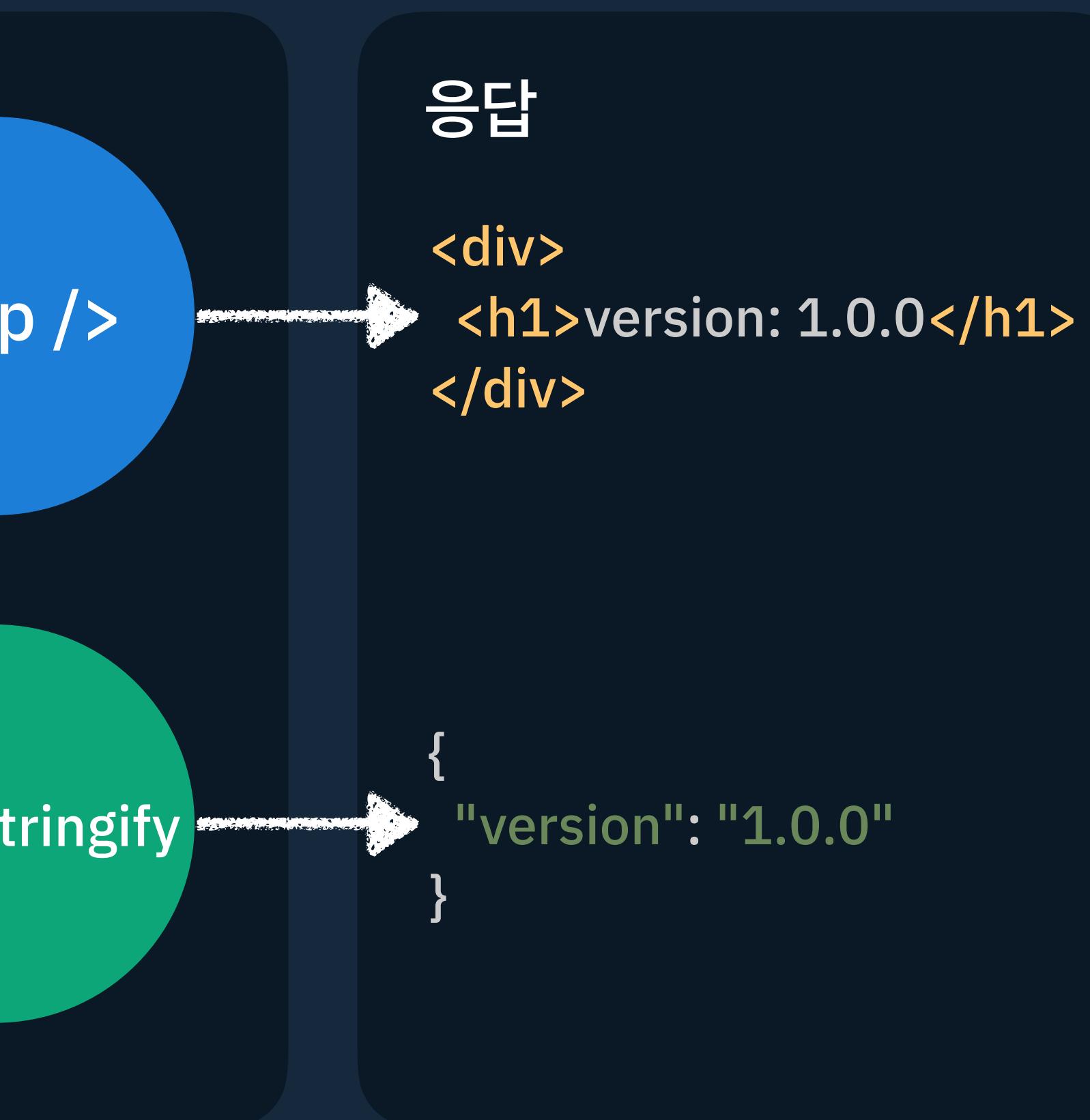
Store Hydration을 통한 데이터 주입

서버에서 요청을 분석해 정적 응답을 만듭니다



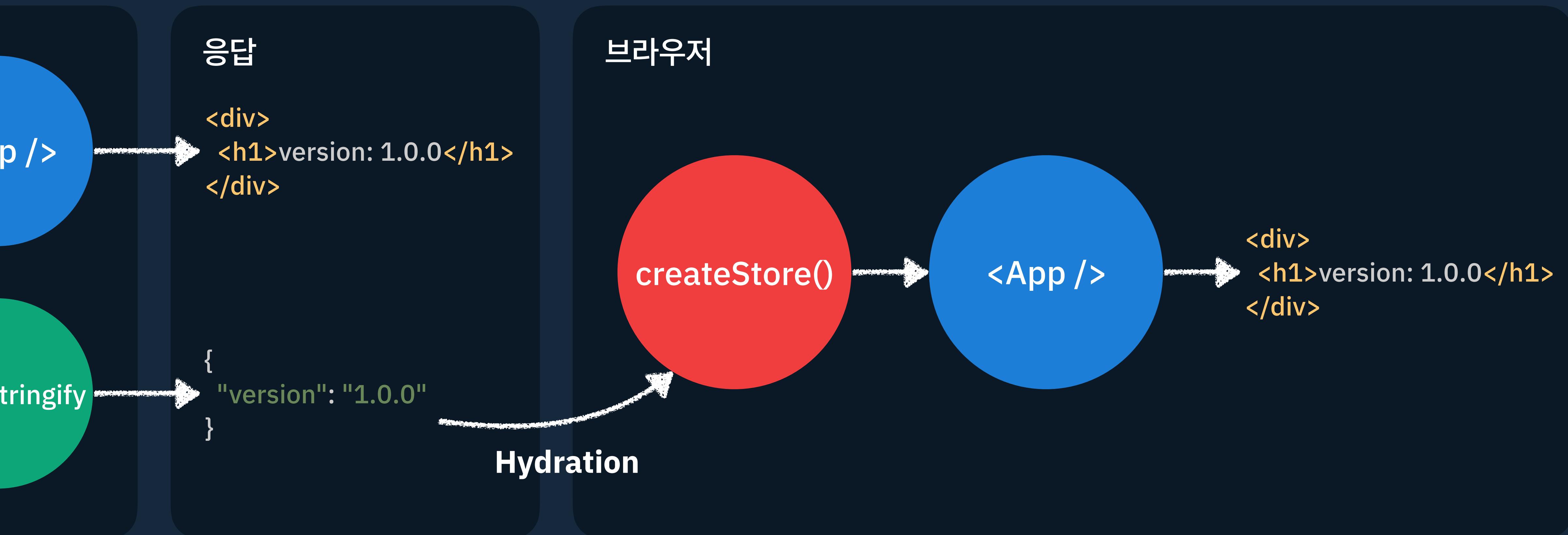
Store Hydration을 통한 데이터 주입

클라이언트에서 정적 응답을 이용해 앱을 Hydrate 합니다



Store Hydration을 통한 데이터 주입

클라이언트에서 정적 응답을 이용해 앱을 Hydrate 합니다



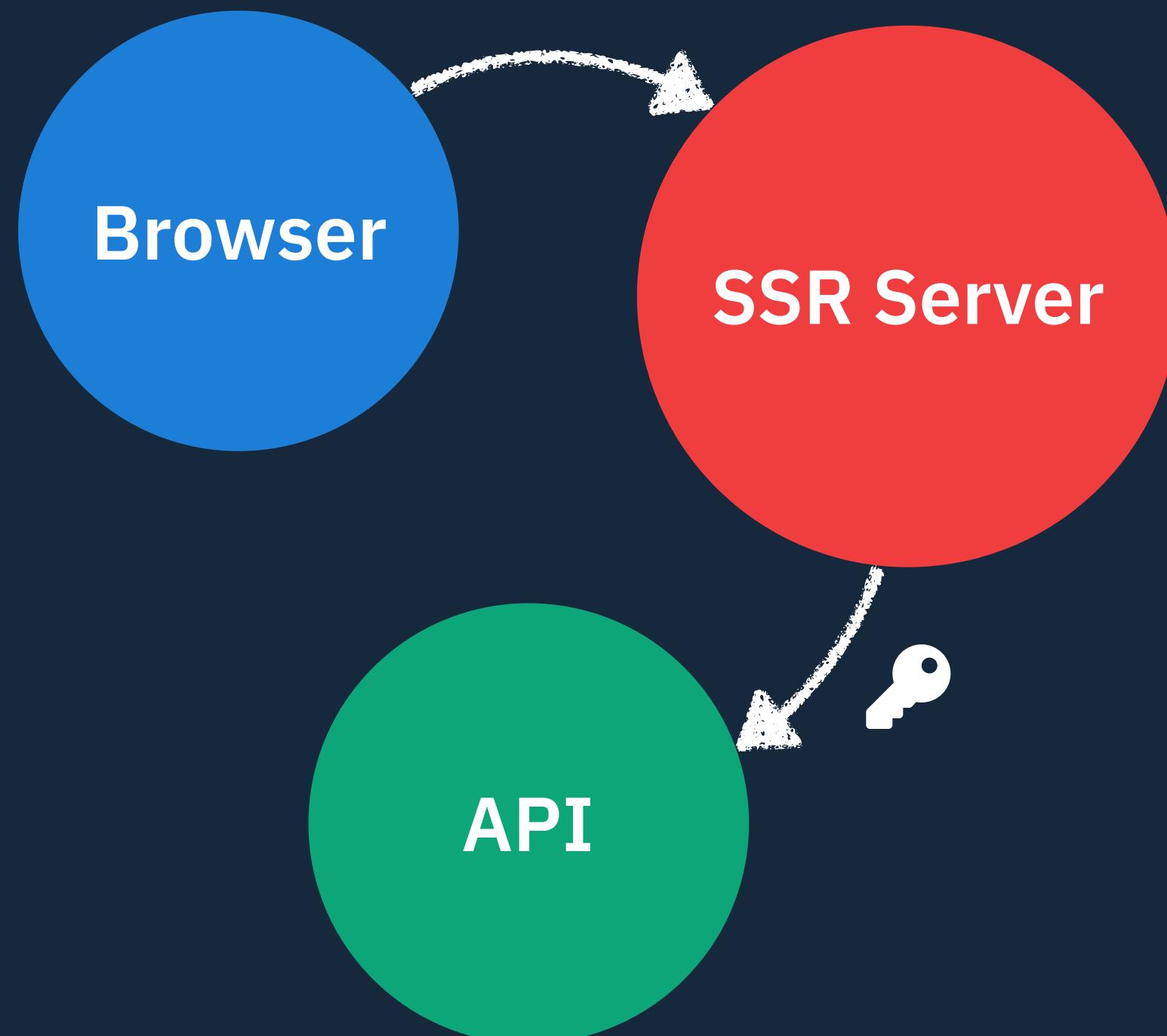
Store Hydration을 통한 데이터 주입

로그인을 어떻게 처리하지?

Store Hydration을 통한 데이터 주입

인증 시나리오 - Server to API

첫 요청 시 (서버에서 React가 작동할 때)

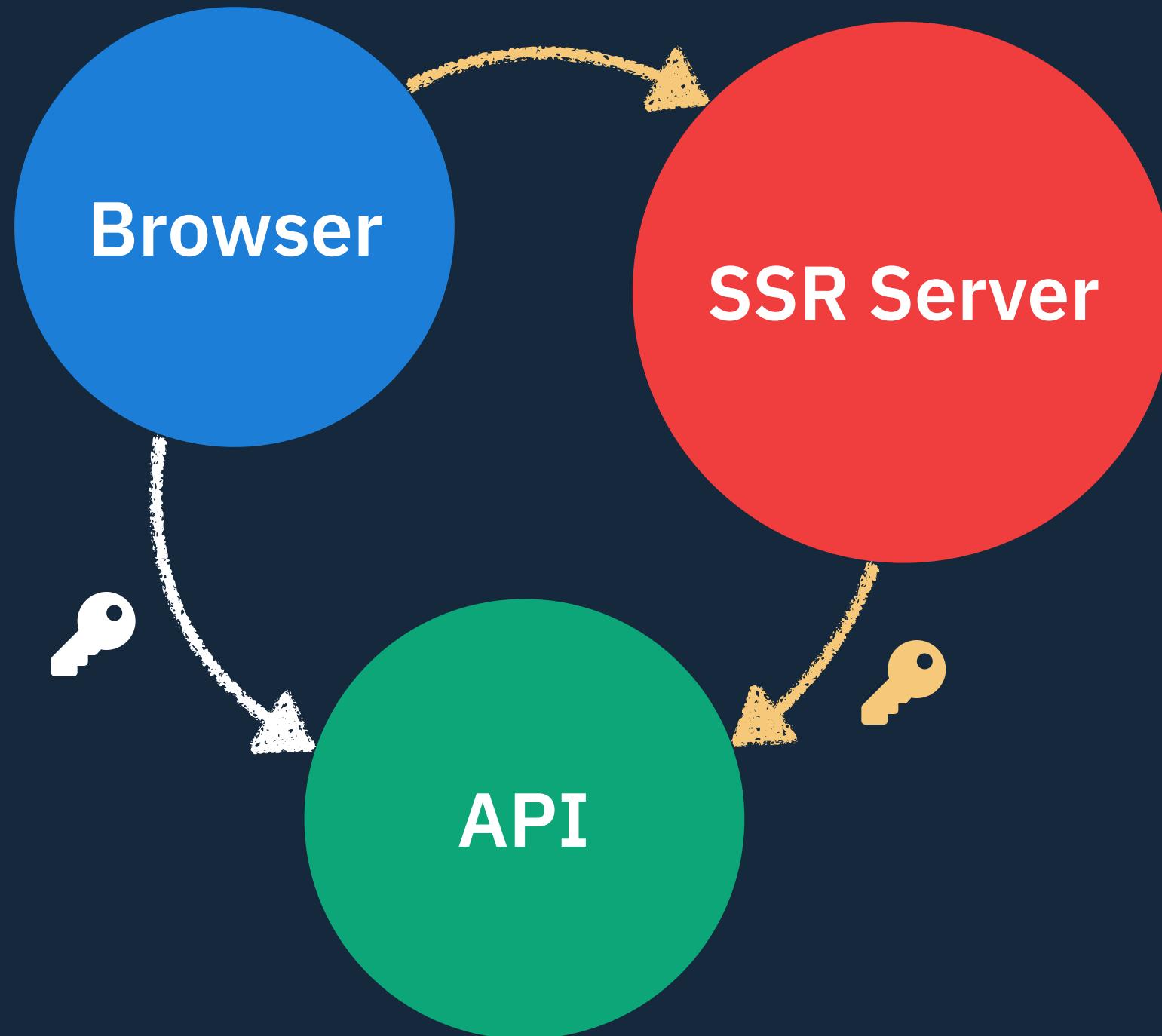


- API로 Data Fetch를 할때 Access Token이 필요
- 이 때 만약 Access Token이 만료되었다면,
토큰 새로고침을 위해 Refresh Token이 필요

Store Hydration을 통한 데이터 주입

인증 시나리오 - Browser to API

브라우저 환경 (브라우저에서 React가 동작할 때)



- API로 Data Fetch를 할때 Access Token이 필요
 - 서버 내에 Proxy를 만들어 사용한다면 서버 내의 Access Token로 인증 가능
 - 단, 매 요청 시 퍼포먼스가 저하되고, 서버에 부하가 생김
- 만약 Access Token이 만료되었다면, 토큰 새로고침을 위해 Refresh Token이 필요
 - 서버 내에 Proxy를 만들어 사용한다면 브라우저 내에 Refresh Token이 없어도 토큰 새로고침 가능
 - 간헐적으로 요청하므로, 서버에 부하 적음

Store Hydration을 통한 데이터 주입

токен을 어디에 저장할까?

단 하나의 원천

LocalStorage

Cookie

Session

- LocalStorage API는 브라우저에서만 이용할 수 있으므로 → 😢 탈락
- Cookie는 가능하지만, Refresh Token이 매 요청마다 Network를 통해 노출되므로 → 😢 탈락

Store Hydration을 통한 데이터 주입

токен을 어디에 저장할까?

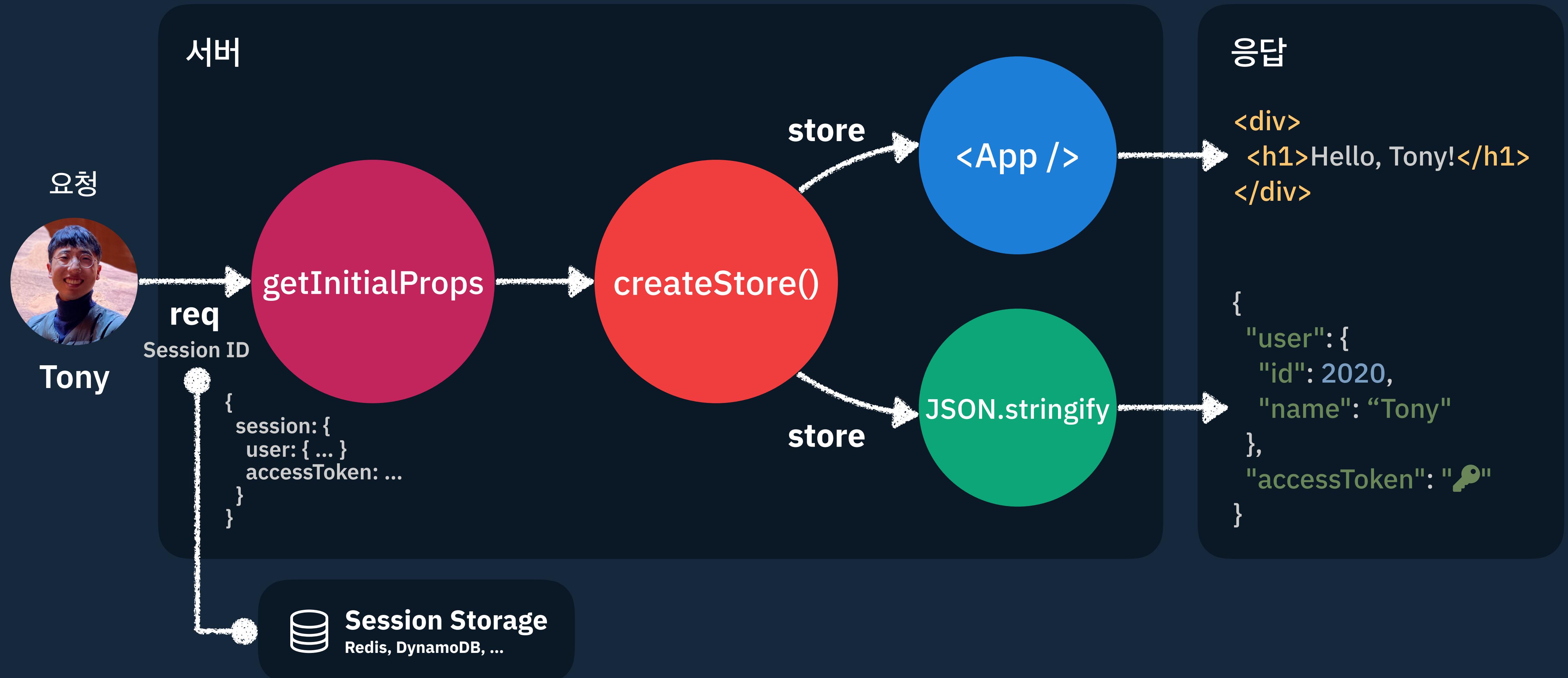


Session

- Session 방식 →  채택
 - Session에 저장된 Access Token을 Store Hydration을 통해 브라우저 환경에 주입
 - Refresh Token은 서버 내 Proxy로 구현

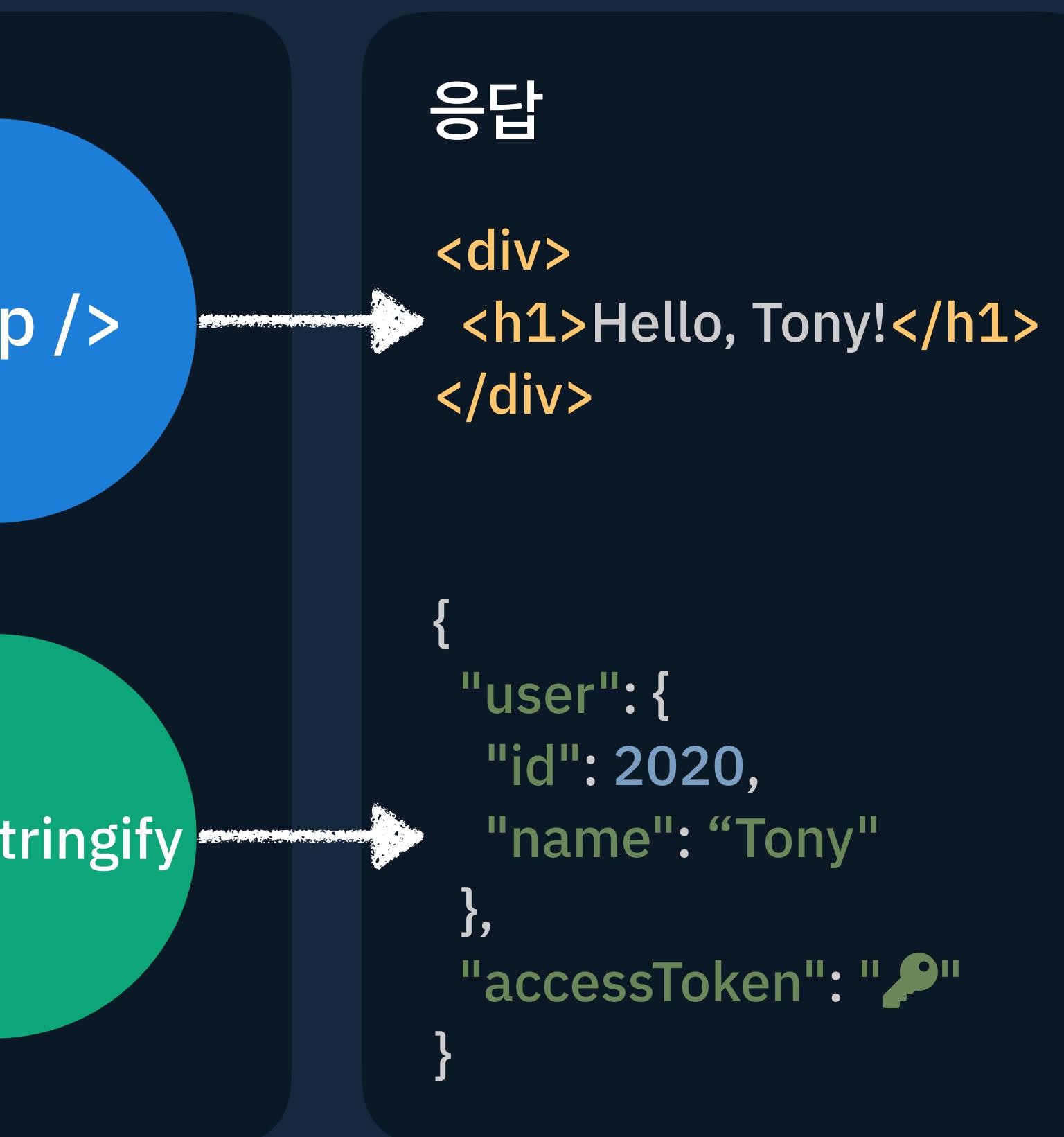
Store Hydration을 통한 데이터 주입

세션에서 데이터를 가져와 Store를 채웁니다



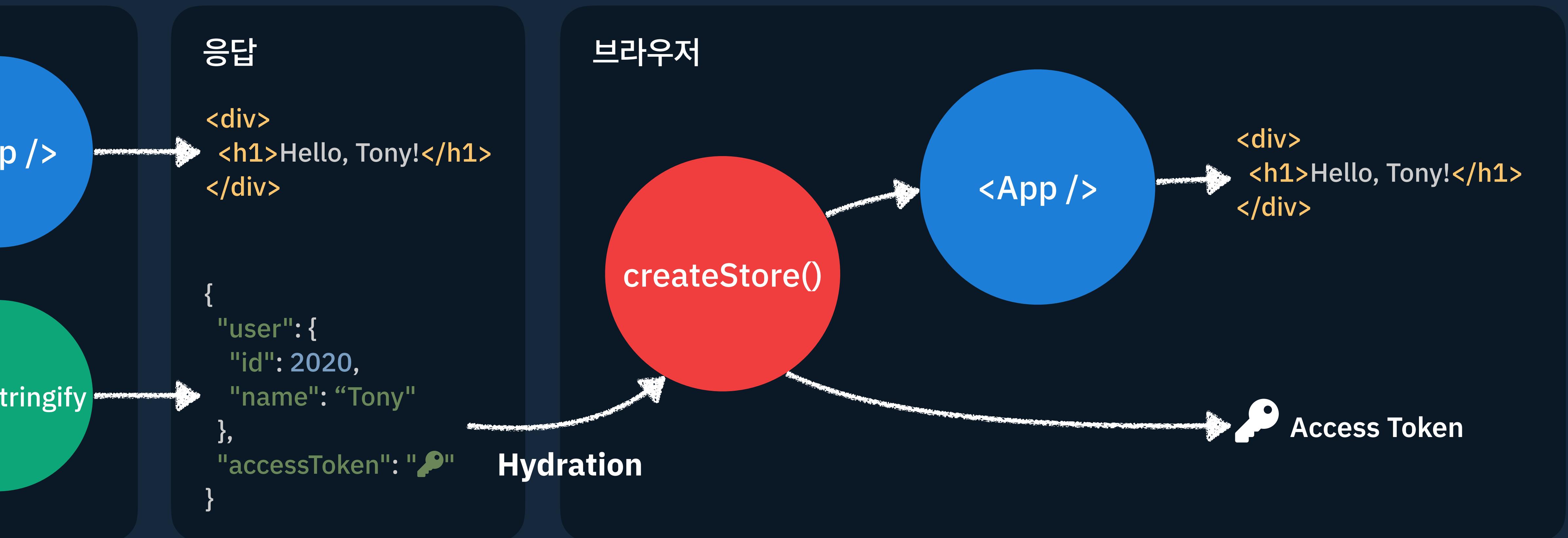
Store Hydration을 통한 데이터 주입

Store Hydration을 통해 브라우저로 데이터를 주입합니다



Store Hydration을 통한 데이터 주입

Store Hydration을 통해 브라우저로 데이터를 주입합니다



Elastic Beanstalk과 ElastiCache를 활용해 어플리케이션 배포하기



Hello, Elastic Beanstalk



AWS Elastic Beanstalk

- 웹 애플리케이션 및 서비스를 간편하게 배포하고 조정할 수 있는 서비스
- 코드를 업로드하기만 하면 Elastic Beanstalk가 용량 프로비저닝, 로드 밸런싱, Auto Scaling부터 시작하여 애플리케이션 상태 모니터링에 이르기까지 배포를 자동으로 처리
- 본 발표에서는 Next.js 서버를 올릴때 사용 (ECS, EKS, Lambda 등 다른 런타임 사용 가능)

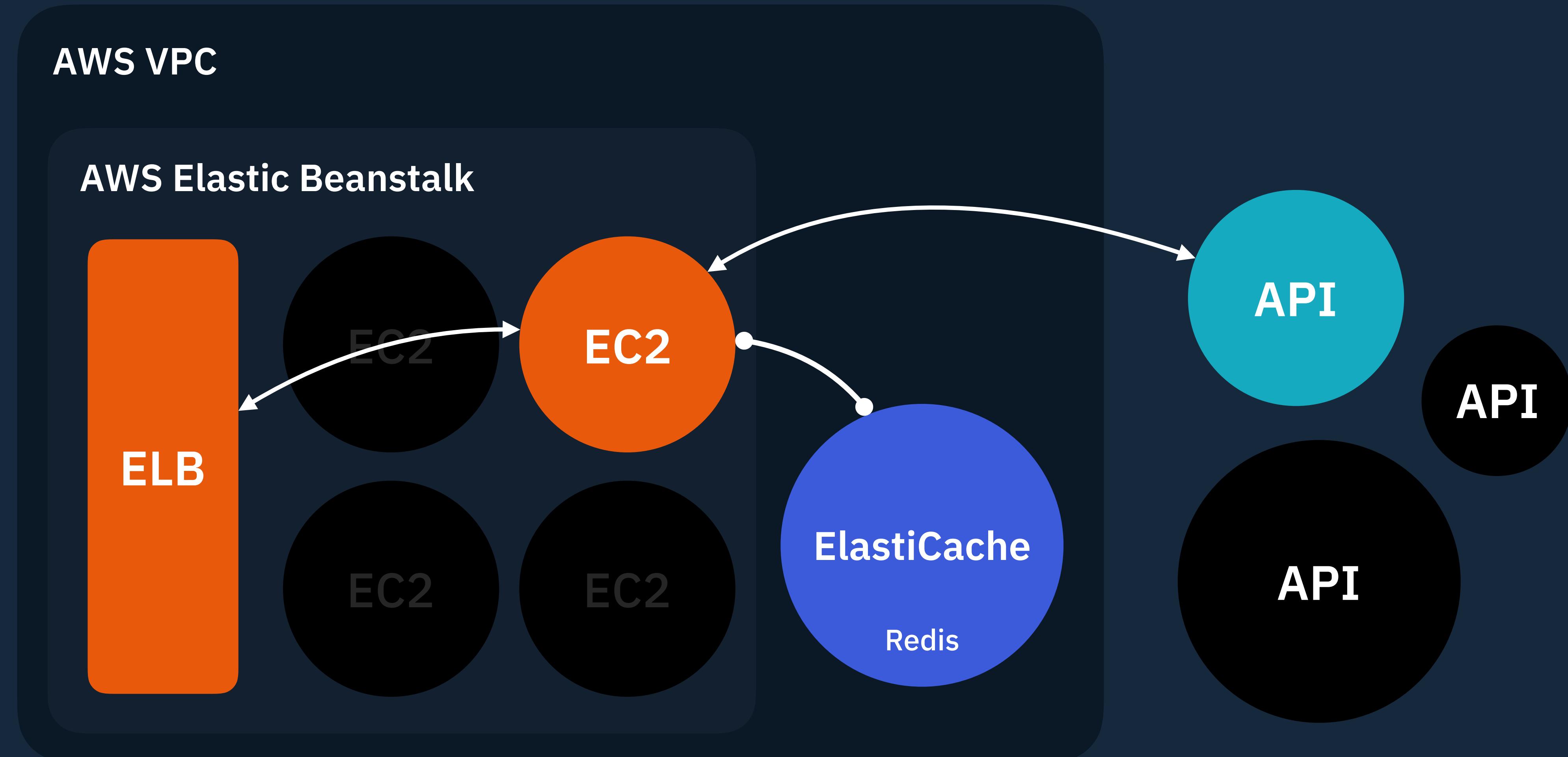
Hello, ElastiCache



- 완전관리형 Redis 및 Memcached
- 본 발표에서는 세션 스토리지로 사용

AWS ElastiCache

Architecture



CloudFront를 이용한 정적 자원 제공 가속화



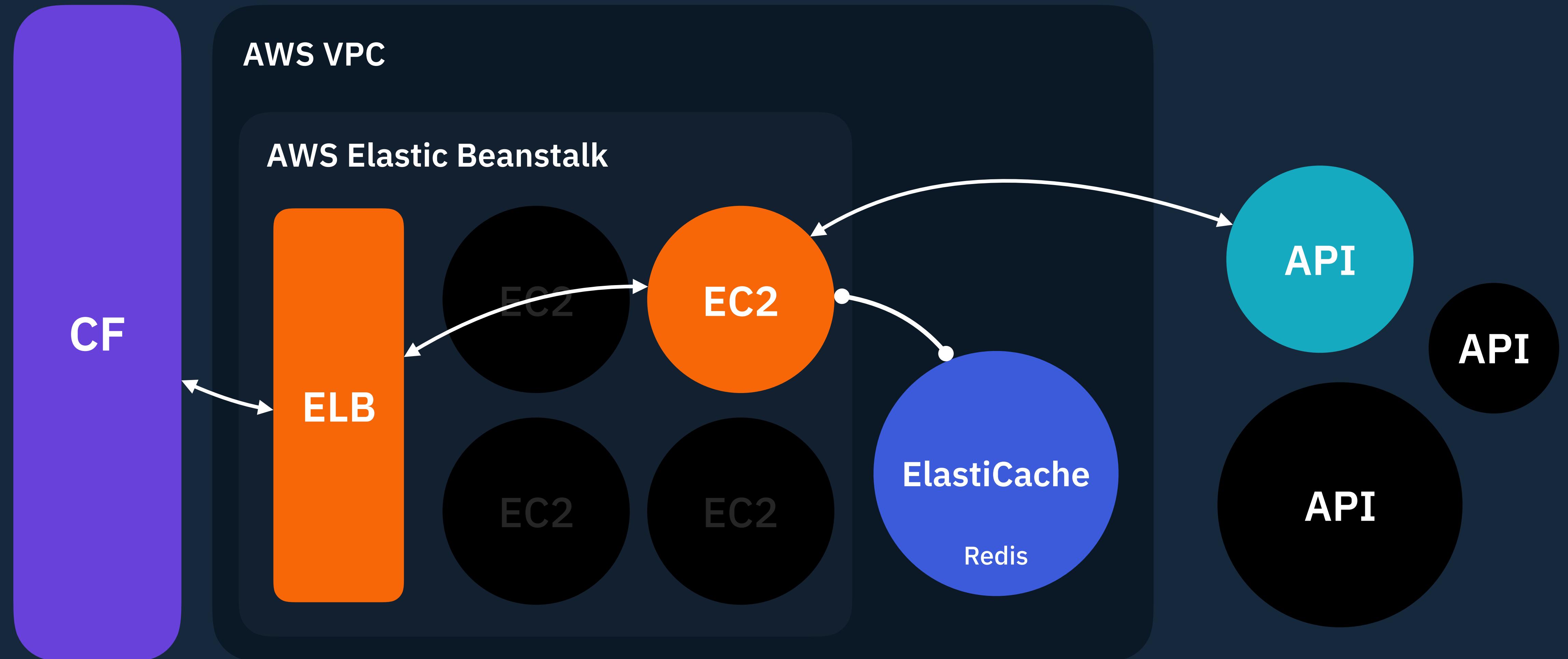
Hello, CloudFront



AWS CloudFront

- 짧은 지연 시간과 빠른 전송 속도로 데이터, 동영상, 애플리케이션 및 API를 전 세계 고객에게 안전하게 전송하는 고속 콘텐츠 전송 네트워크(CDN) 서비스
- 본 데모에서는 `/_next` 밑의 정적 JS, CSS 제공을 가속화하기 위한 목적으로 ELB 앞단에 붙임

Architecture+



Demo



tony-macbookpro:workspace keynote\$ git cl



<https://bit.ly/aws-ssr-demo>

에서 해당 데모를 따라해보실 수 있습니다



감사합니다

질문은 언제나 환영입니다! jeehyukwon@gmail.com

