

Smart WebDesktop기반 모바일 플랫폼

목차

I. 과제 개요

1. 과제 배경
2. 일반적인 웹데스트탑이란
3. Smart WebDesktop 특징
4. 과제 개발 내역
5. 기대 효과

II. 개발 내역

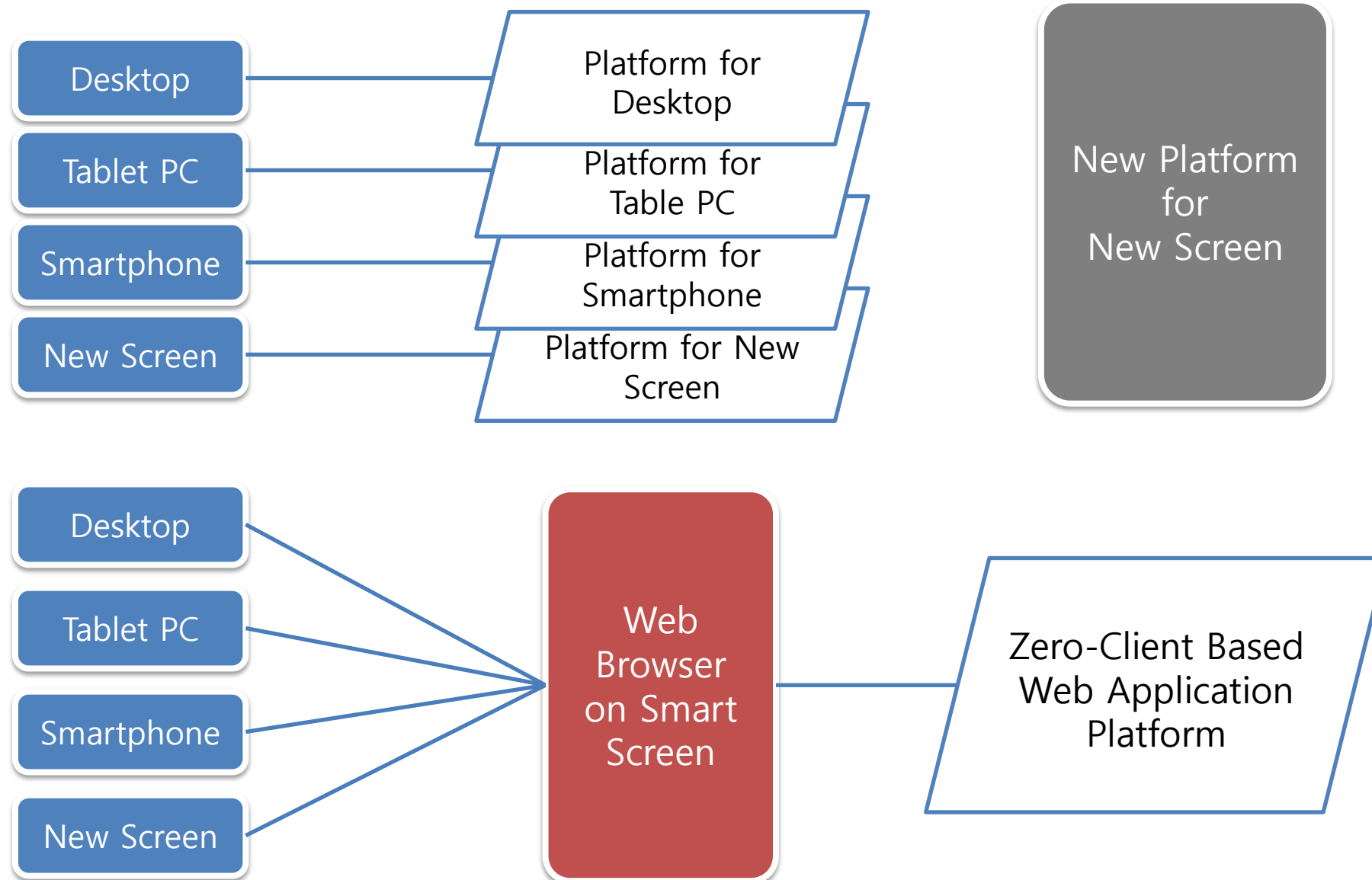
1. 아키텍처 요구 사항
2. Smart WebDesktop 전체 구성도
3. Smart WebDesktop Client 아키텍처
4. Smart WebDesktop Client Framework Component
5. Smart WebDesktop Server 아키텍처
6. Smart WebDesktop Server Component
7. Smart WebDesktop Data 흐름도
8. Smart WebDesktop 데이터 설계
9. 주요 적용 기술
10. Use Case

과제개요

1. 과제 배경

❖ 과제 배경

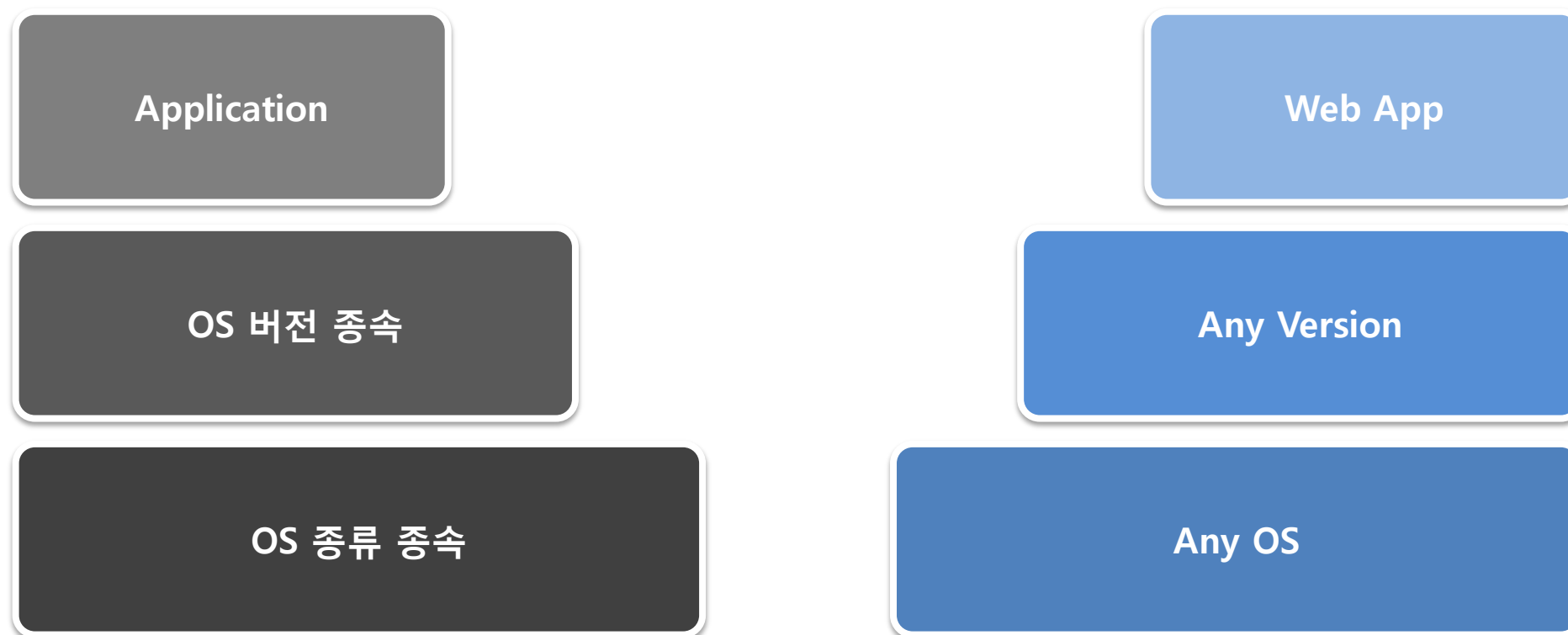
- n-Screen을 지원하는 모바일 플랫폼 필요



1. 과제 배경

❖ 과제 배경

- 앱에서 웹 앱으로의 패러다임 진화 중
 - 앱은 호스트 운영체제 및 버전에 따라 앱의 추가개발 필요
 - 웹 애플리케이션은 **zero client 기반**으로 단말에서의 버전 관리 불필요



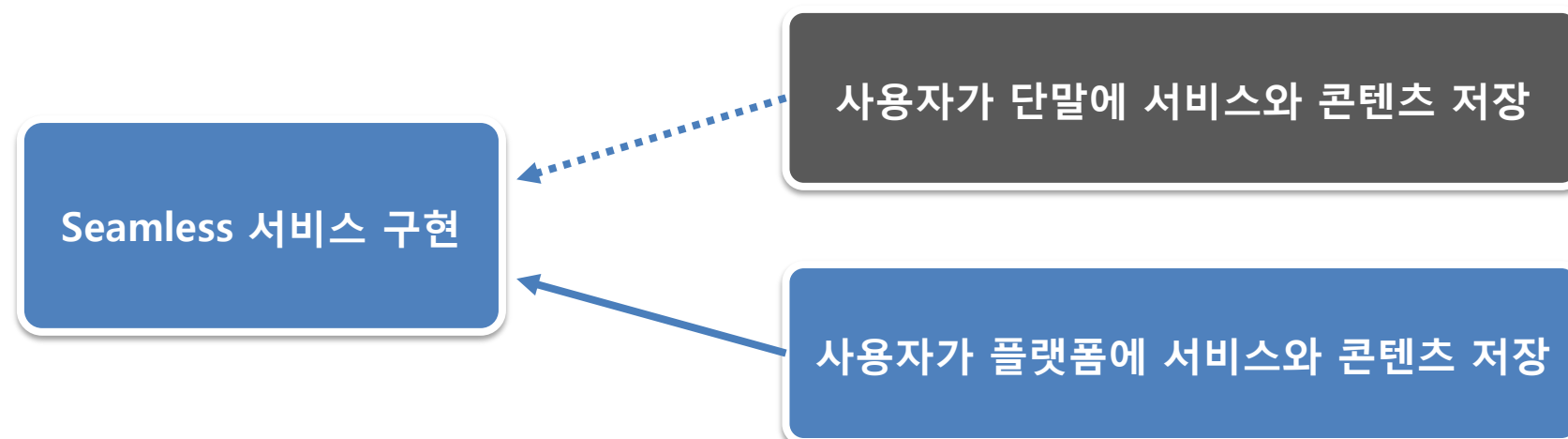
1. 과제 배경

❖ 과제 배경

- 스크린 확대와 모바일 단말 증가로 보안 취약성 강화 필요



- 모바일 환경에서의 seamless 필요성 증가



2. 일반적 웹데스크탑이란?

❖ 디바이스 및 OS에 독립적인 웹 어플리케이션 사용 환경

- 운영체제(OS)와 응용프로그램(Application), 데이터 모두 웹으로부터 제공되는 웹 기반의 데스크 탑 환경
- 가상화된 스토리지에 사용자 정보를 저장
- 사용자 환경(UX/UI)은 웹 브라우저가 기반
- 사용자는 웹 브라우저 사용 가능한 단말 어디에서나 동일한 사용자 환경을 유지



3. Smart WebDesktop 특징

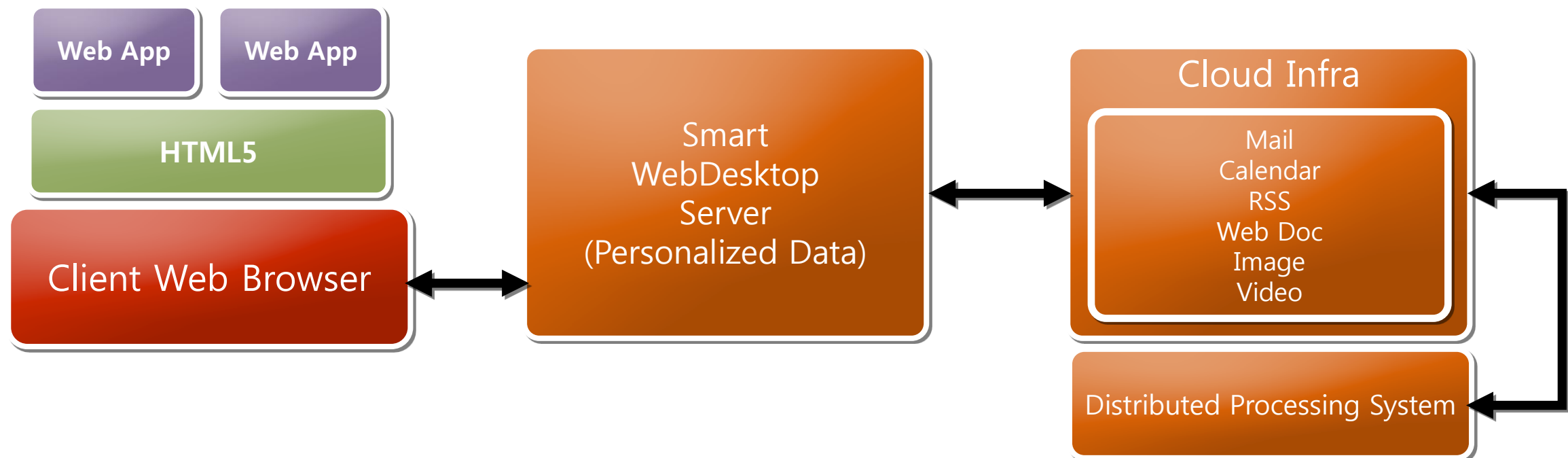
❖ HTML5 기반

- 기존 불가능했던 로컬 스트리지 제어, Rich UI 등
- Device API 지원

❖ N-screen 지원의 모바일 플랫폼

- Seamless 서비스 환경 제공
- Zero Client

❖ 클라우드 서비스로서의 웹 데스크탑으로 기능



4. 과제 개발 내역

❖ Smart WebDesktop 아키텍처 정의 및 개발

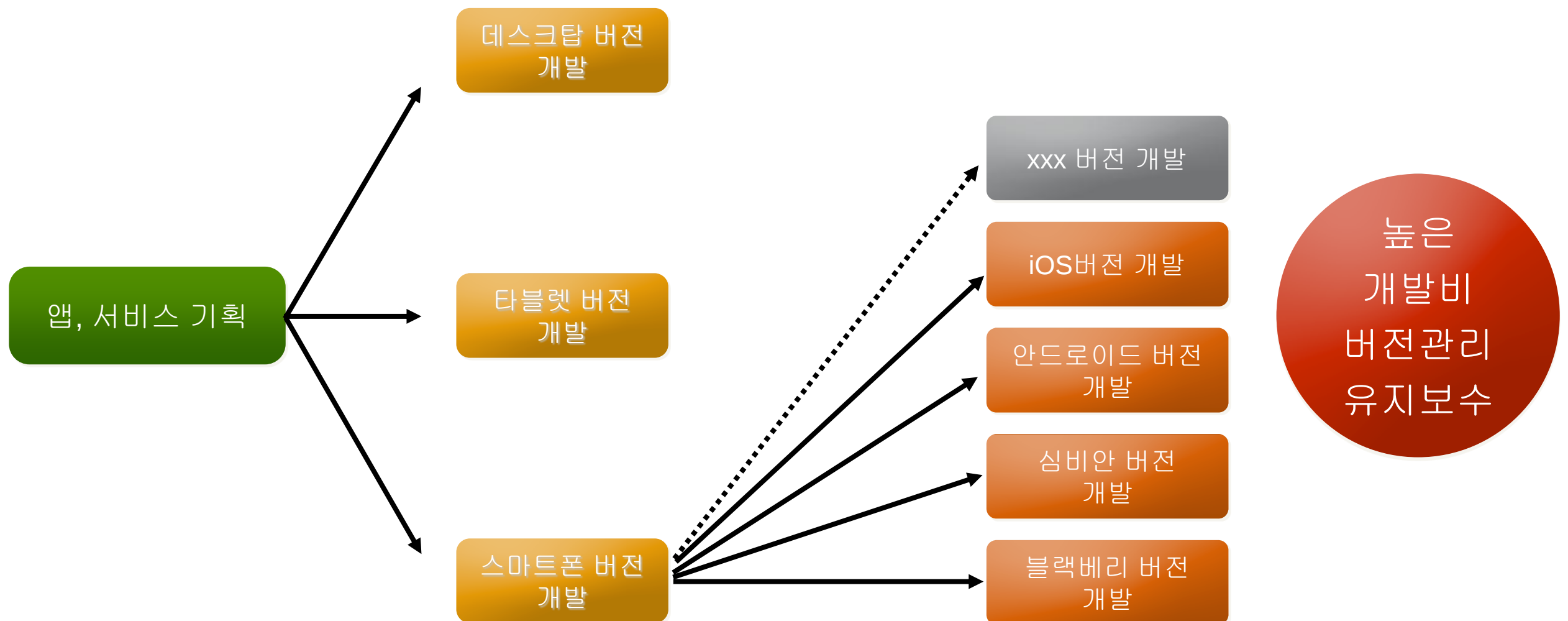
- 안드로이드 기반 스마트 폰을 대상으로 하는 사용자 환경을 개발
- Smart WebDesktop 서버 아키텍처 정의 및 개발
- 오픈 소스 기반의 스택 활용
- 대용량 트래픽 처리와 클라우드 인프라를 위한 오픈 소스 채택

[HP WebOS 사례]



4. 기대효과

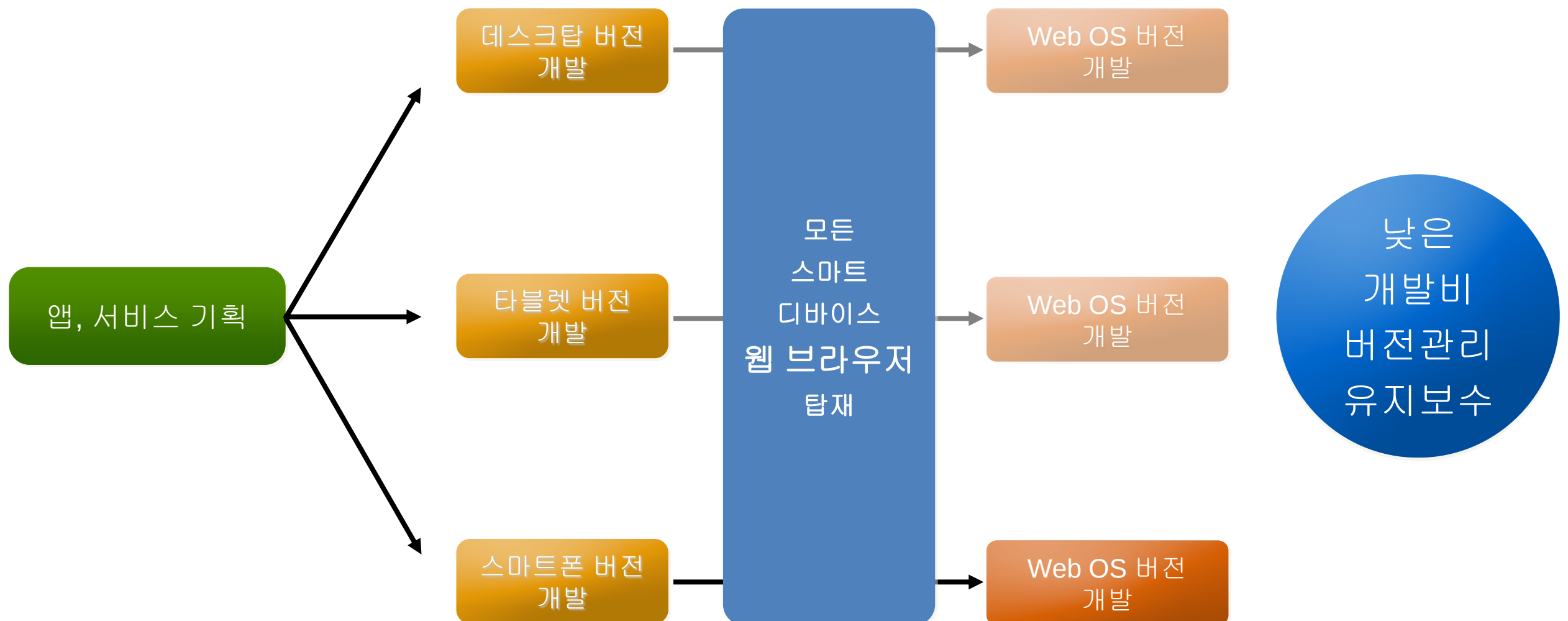
❖ 개발사 관점의 기대효과



4. 기대효과

❖ 개발사 기대효과(계속)

- 사용자 단말 다양화에 따른 복수 개발 비용 절감
- 버전관리와 유지보수 비용 절감
- 클라우드 플랫폼 연계 서비스



4. 기대효과

❖ 이용자 기대효과

- 사용자 단말에 제한 받지 않는 n-Screen 사용자 환경
- 일관성 있는 사용자 환경

Desktop



Smart
WebDesktop



Tablet PC



스마트폰



개 발 내 역

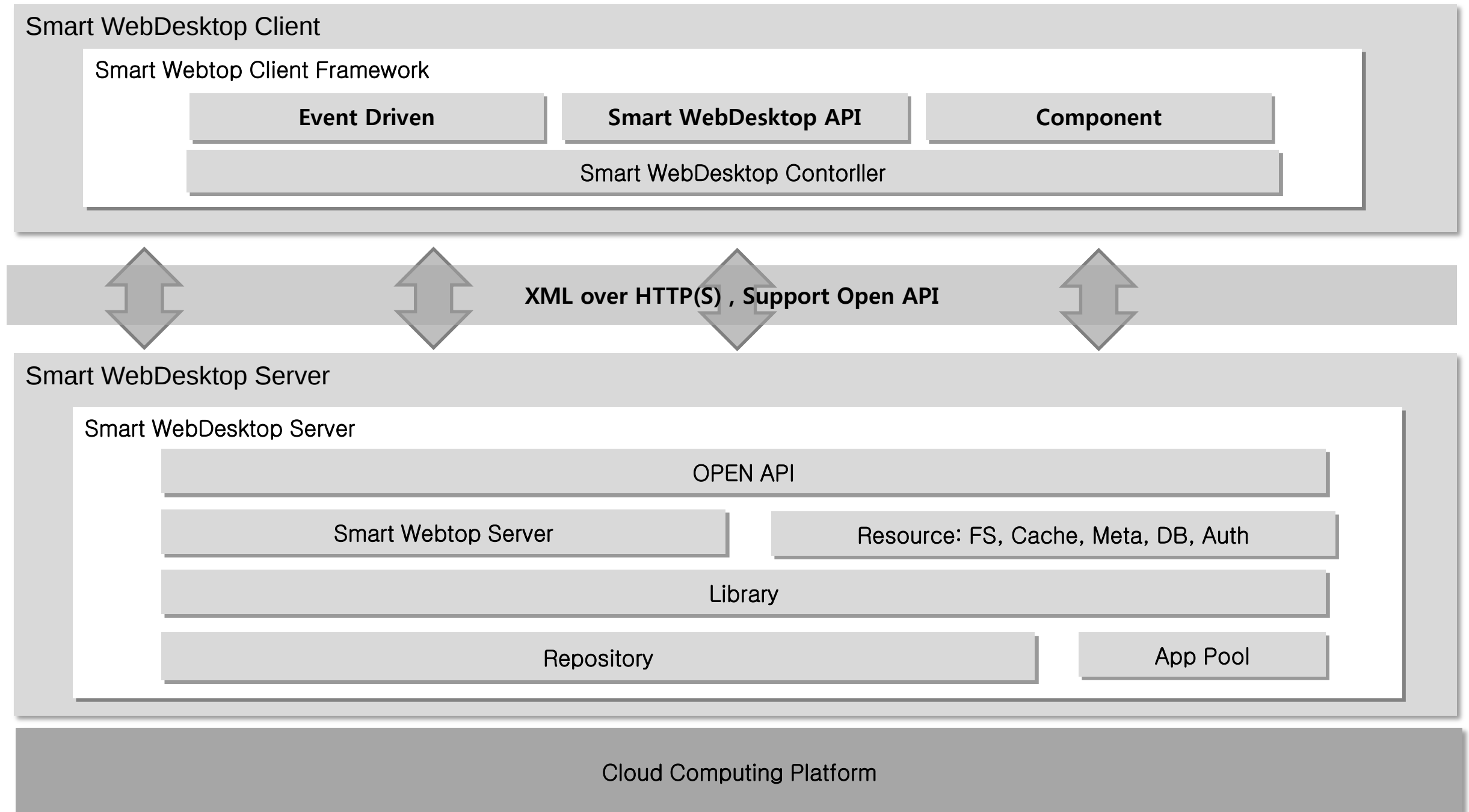
1. 아키텍처 요구 사항

- 대규모 사용자 처리를 위한 클라우드 아키텍처
- 대규모 사용자 서비스를 위한 분산 웹데스크탑 커널 구현
- 서비스 증가, 사용자 증가에 유연한 확장
- 높은 보안
- 개방형 아키텍처(OpenAPI)



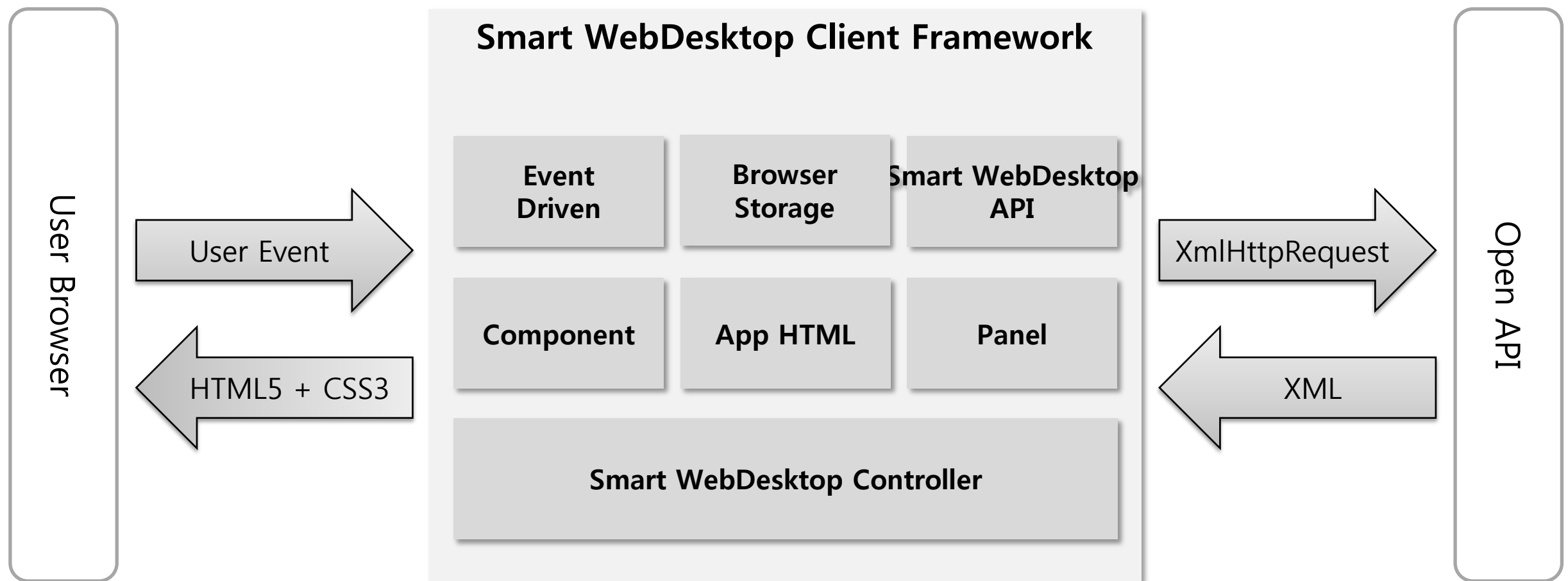
2. Smart WebDesktop 전체 구성도

대규모 사용자 처리를 위한 클라우드 기반의 개방형 아키텍처(OPEN API)와 HTML5기반 zero client



3. Smart WebDesktop Client 아키텍처

HTML5 기반의 zero client

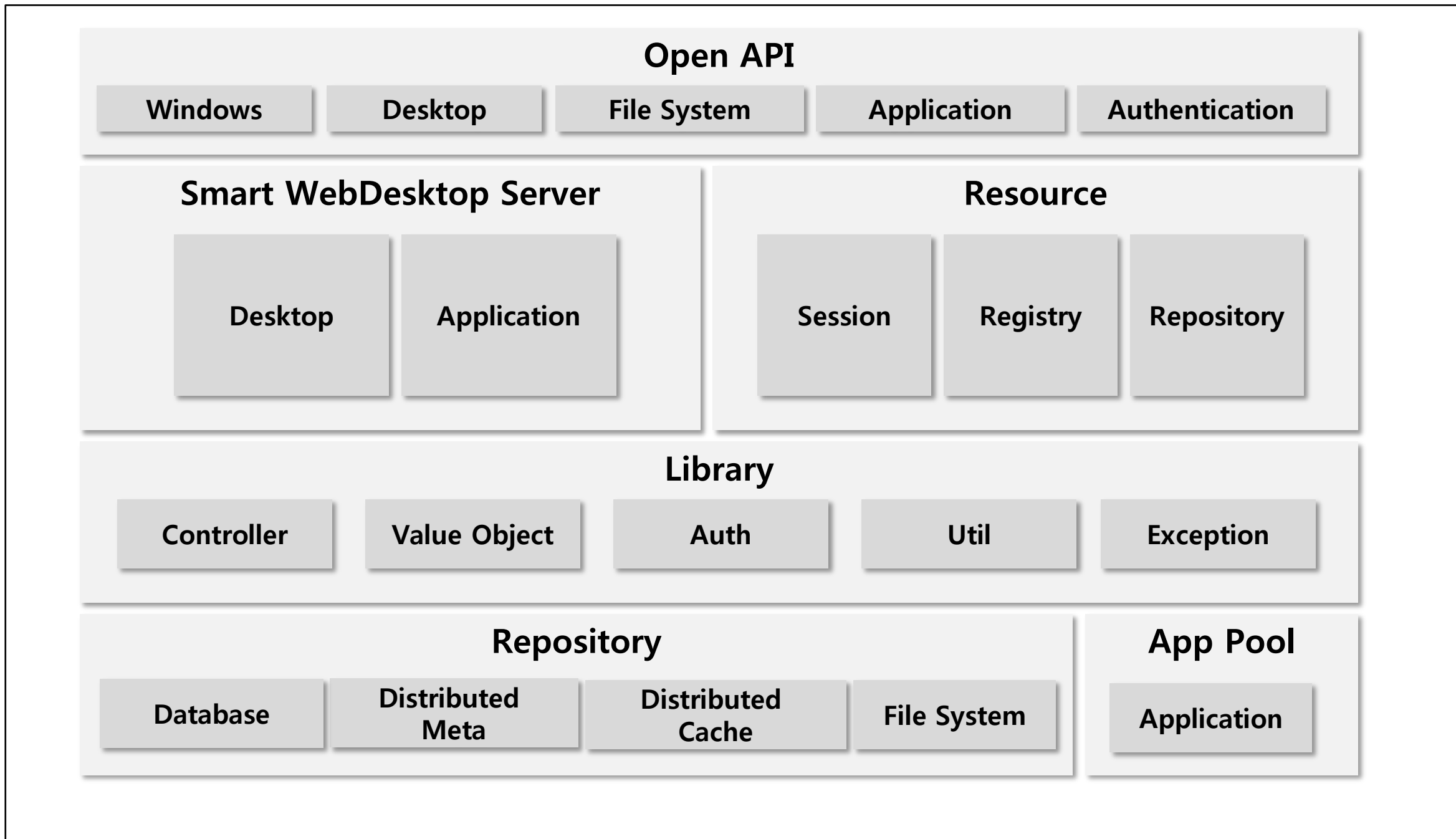


4. Smart WebDesktop Client Framework Component

Component	Description
Event Driven	User Browser에서 발생하는 이벤트 처리를 위한 Component이다.
Smart WebDesktop API	Smart WebDesktop 의 모든 App에서 공통으로 사용하는 API이다.
Open API	Smart WebDesktop에서 제공하는 Open API에 대해 Client에서 일관성 있고 쉽게 사용 할 수 있도록 구성된 Component이다.
Internal Component	Smart WebDesktop에서 자주 사용되는 형태의 Internal Component 라이브러리이다.
App HTML	App이 화면에 보여질 HTML 데이터이다.
Panel	개별적으로 특성을 갖는 각 영역에 대한 layer로서 다음과 같은 Panel이 존재한다.
Smart WebDesktop Controller	OS와 연관된 모든 이벤트를 처리 하기 위해 App의 Process 및 life cycle이 정의된 Controller이다.
Browser Storage	HTML5를 지원하는 브라우저에서 사용할 수 있는 Local Storage이다.

5. Smart WebDesktop Server 아키텍처

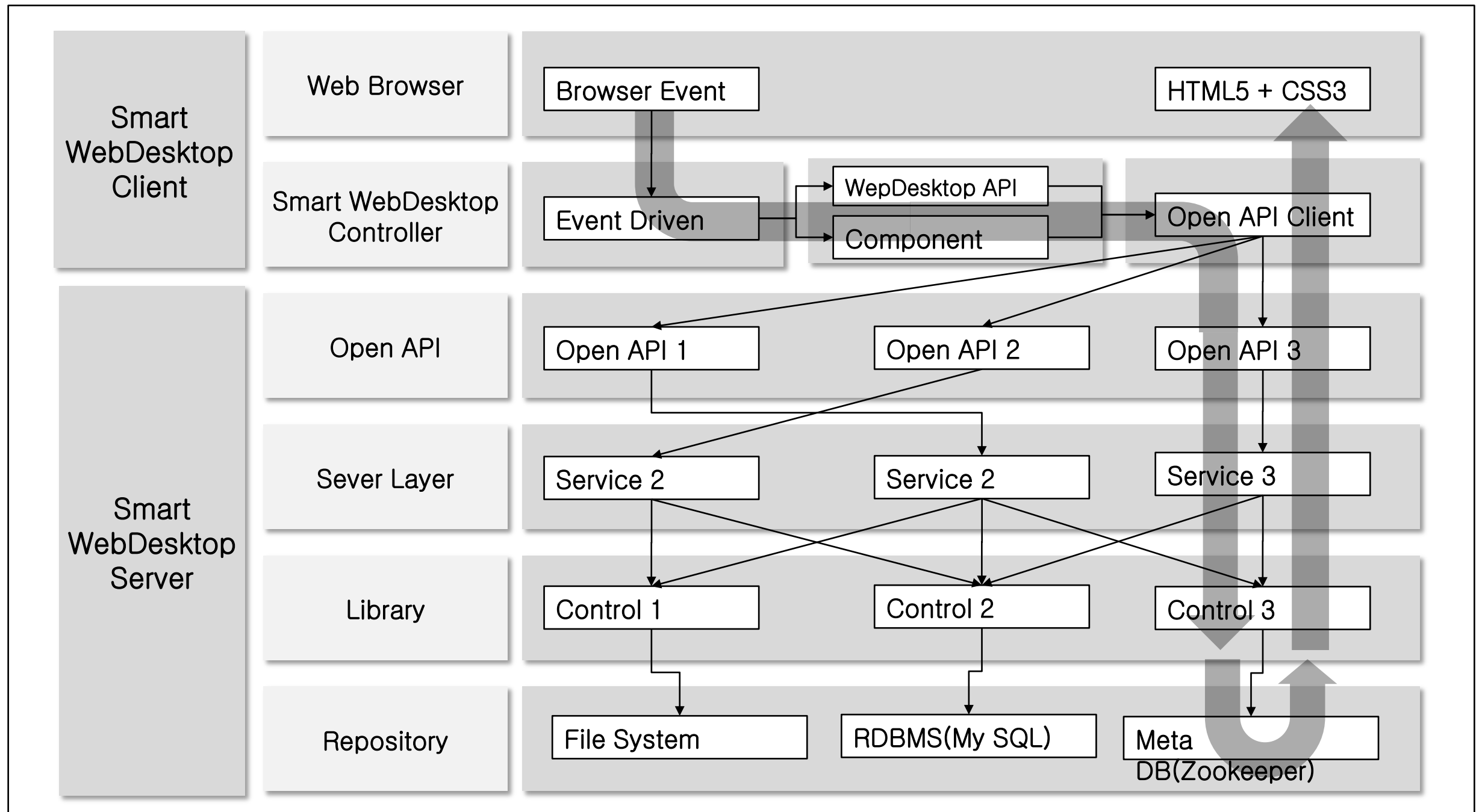
클라우드 아키텍처 기반의 개방형 아키텍처(OPEN API)



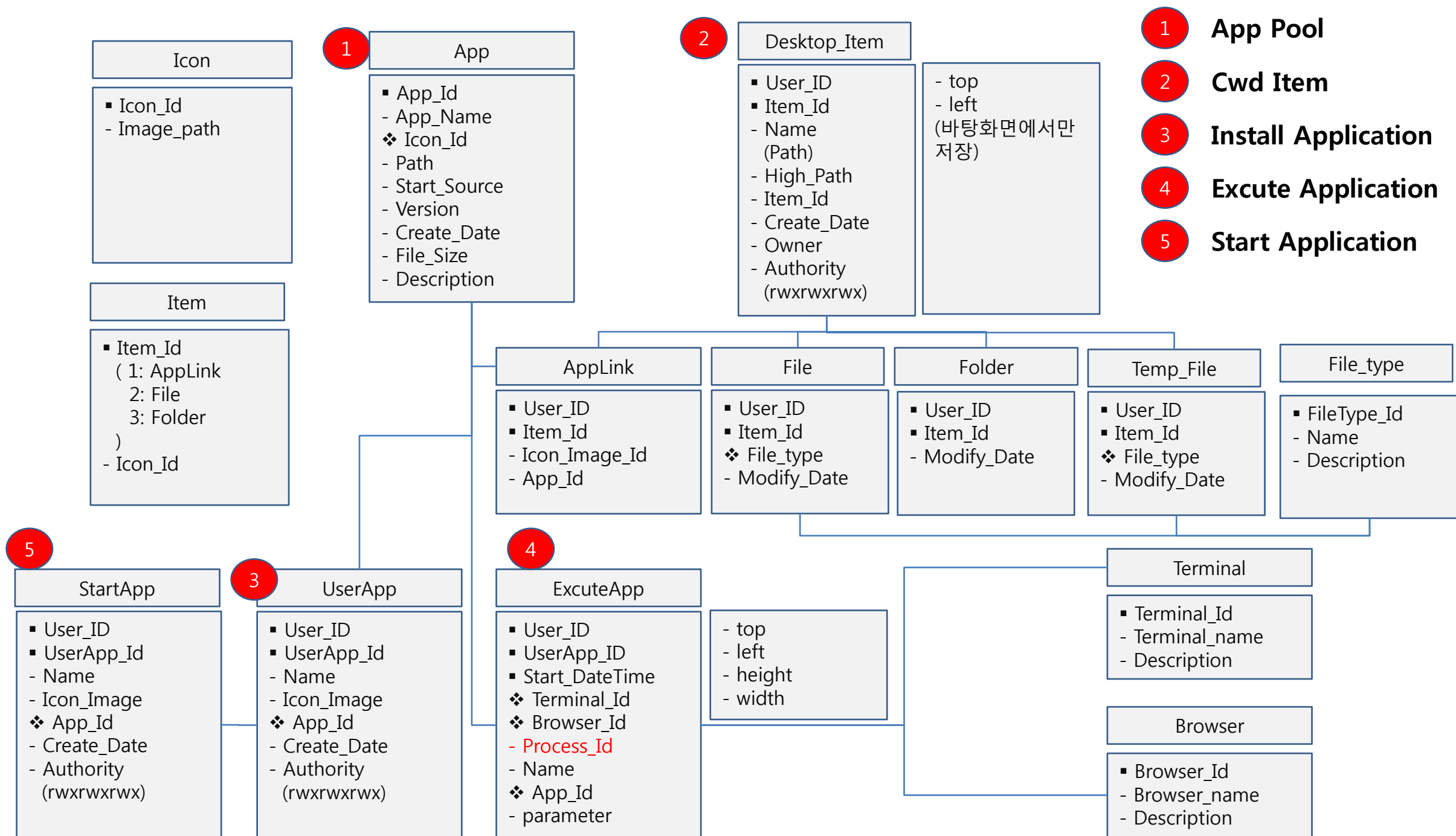
6. Smart WebDesktop Server Component

Component	Description
Smart Webtop Server	Smart WebDesktop의 서비스 Process를 정의하는 레이어
Resource	Smart WebDesktop에서 사용하는 분산 캐쉬, 분산 메타, file system을 Read/Write하기 위한 인퍼페이스 레이어
Library	Smart WebDesktop구성에 필요한 컴포넌트들이 Built-In 되는 레이어
Repository	Smart WebDesktop의 임시 또는 영구 저장에 필요한 데이터 레이어.
App pool	Smart WebDesktop에 등록되는 사용자 Application을 관리하는 레이어

7. Smart WebDesktop Data 흐름도



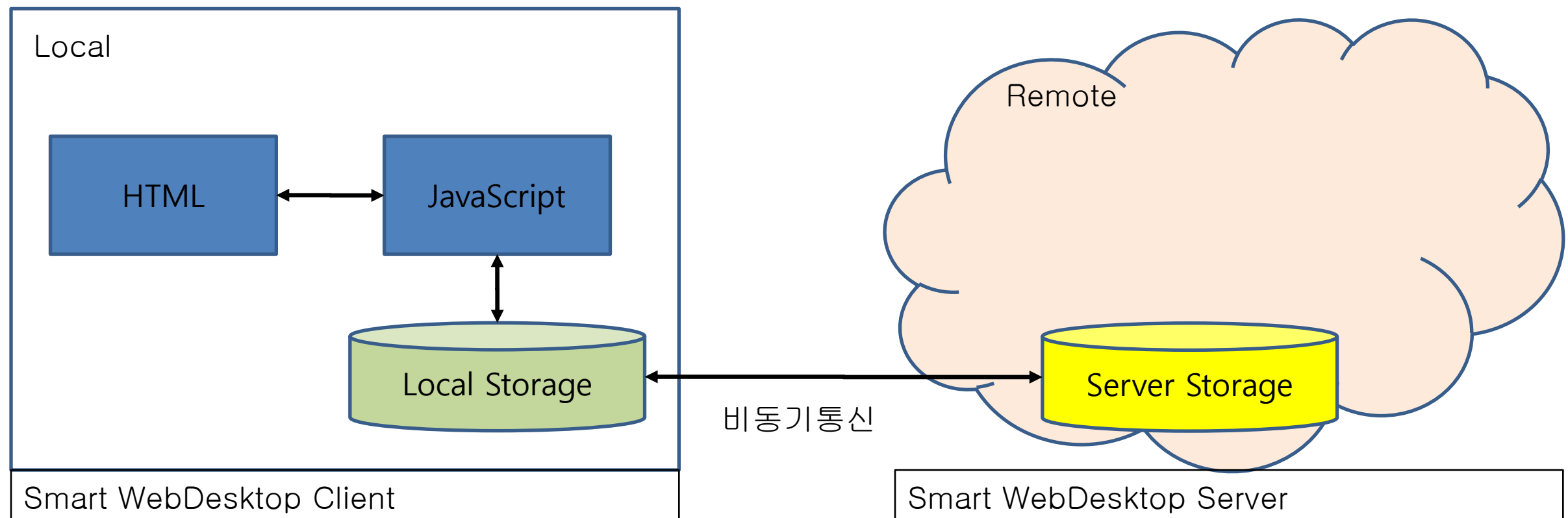
8. Smart Webtop 데이터 설계



9. 주요 적용 기술 : Smart WebDesktop Client

❖ HTML5의 로컬 스토리지(local storage) 와 Ajax 기반의 비동기 통신

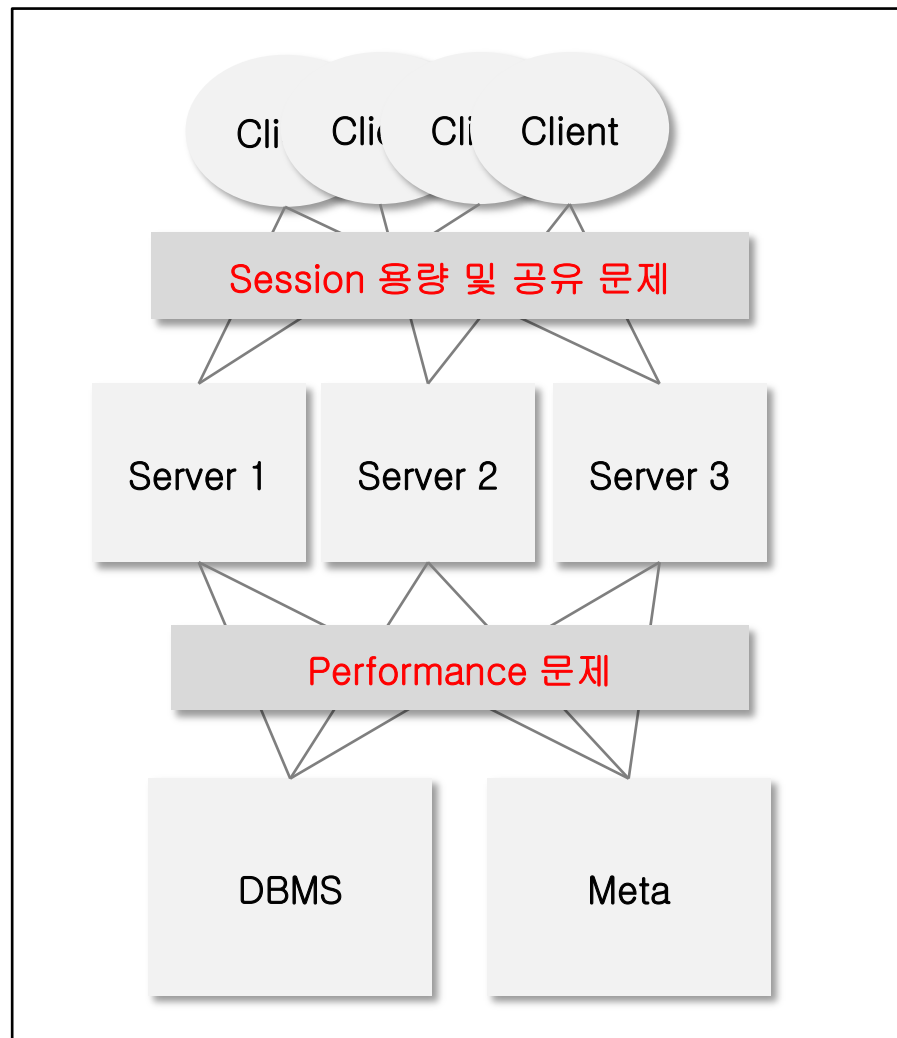
- 무선 네트워크 통신 최소화
- 무선 네트워크 불안전성 극복



9. 주요 적용 기술 : Smart WebDesktop Server

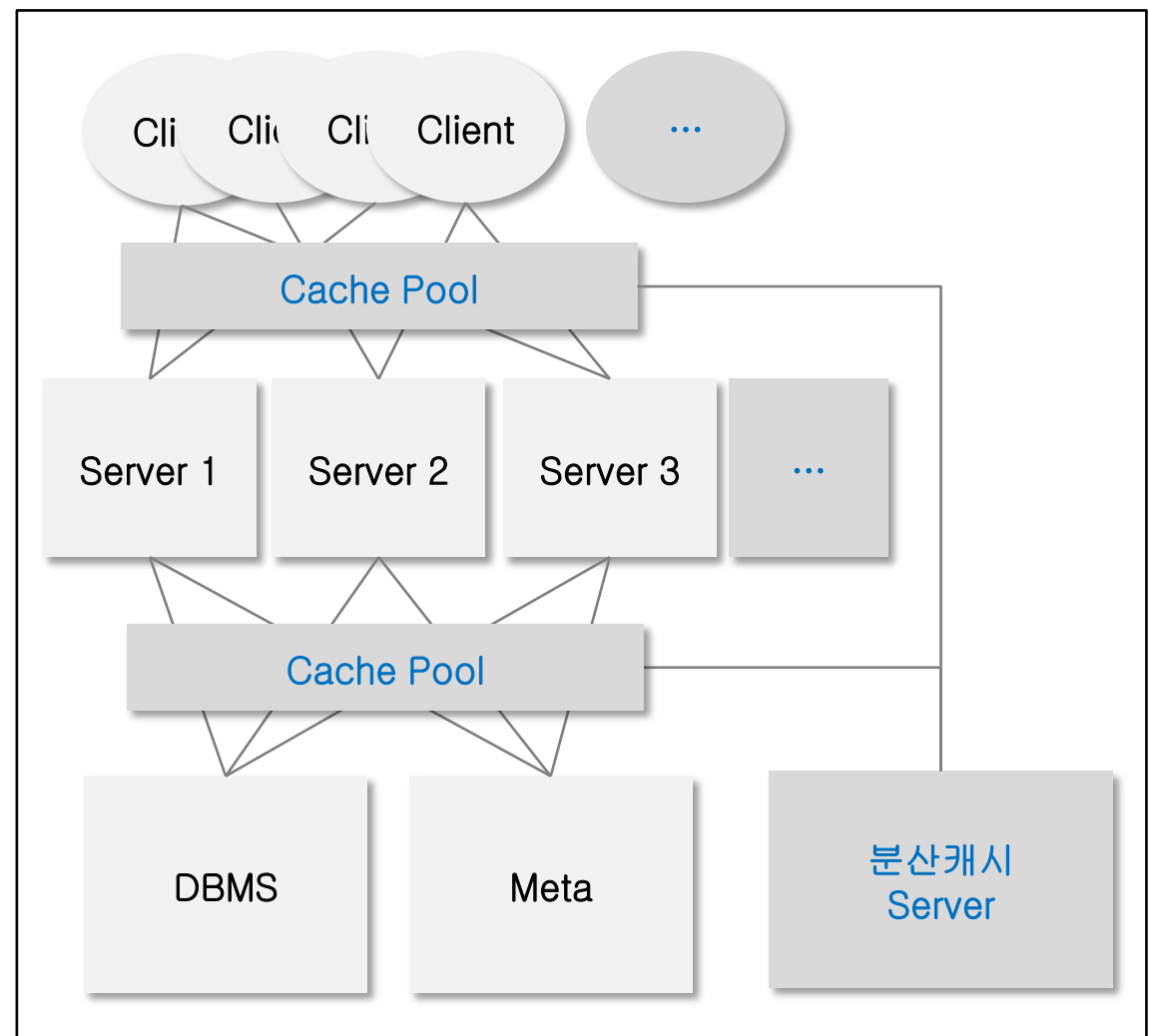
클라우드기반 대용량 트래픽을 처리하기 위한 분산 캐쉬 적용한 클라우드 인프라

[일반적 아키텍처]



- 사용자 정보 및 WAS의 세션, DB에 사용
- 성능, 용량, 확장성에 제약 사항이 있음

[분산 캐쉬서버가 적용된 아키텍처]

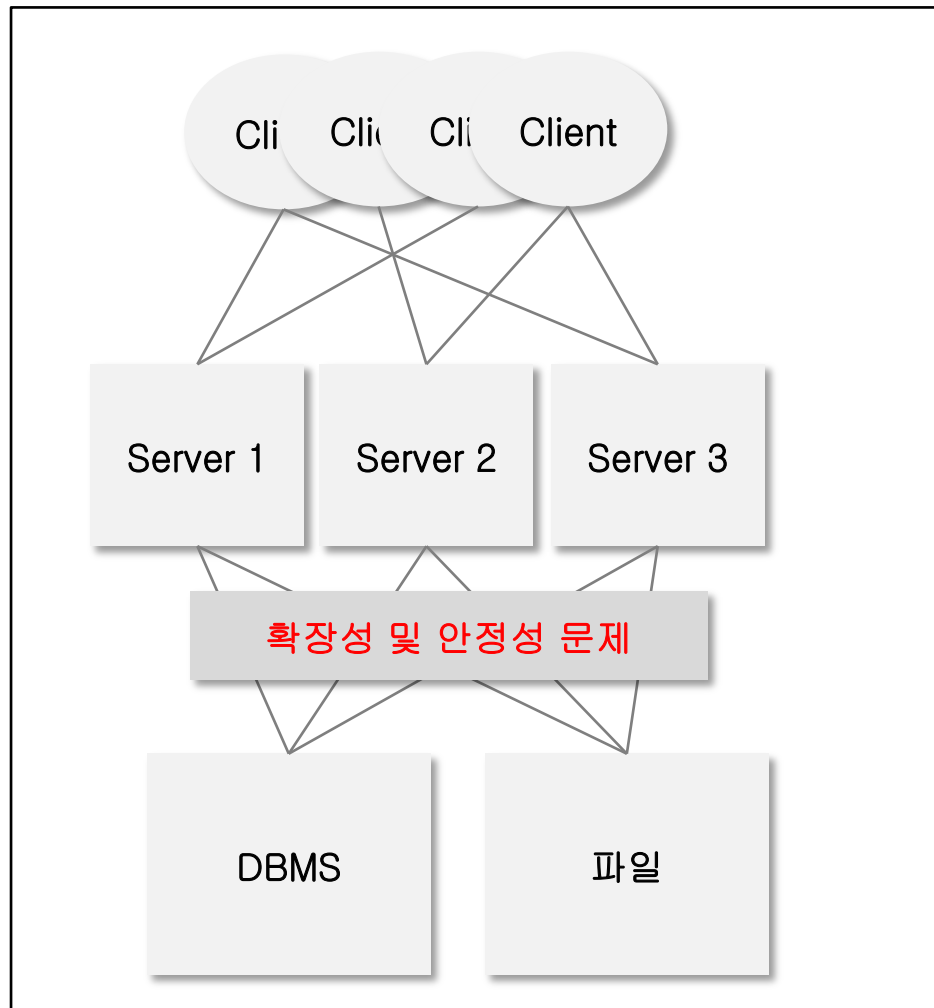


- 사용자 프로파일 정보 및 Smart WebDesktop의 실시간 정보를 메모리에 캐쉬
- 성능, 용량, 확장성 매우 높음

9. 주요 적용 기술 : Smart WebDesktop Server

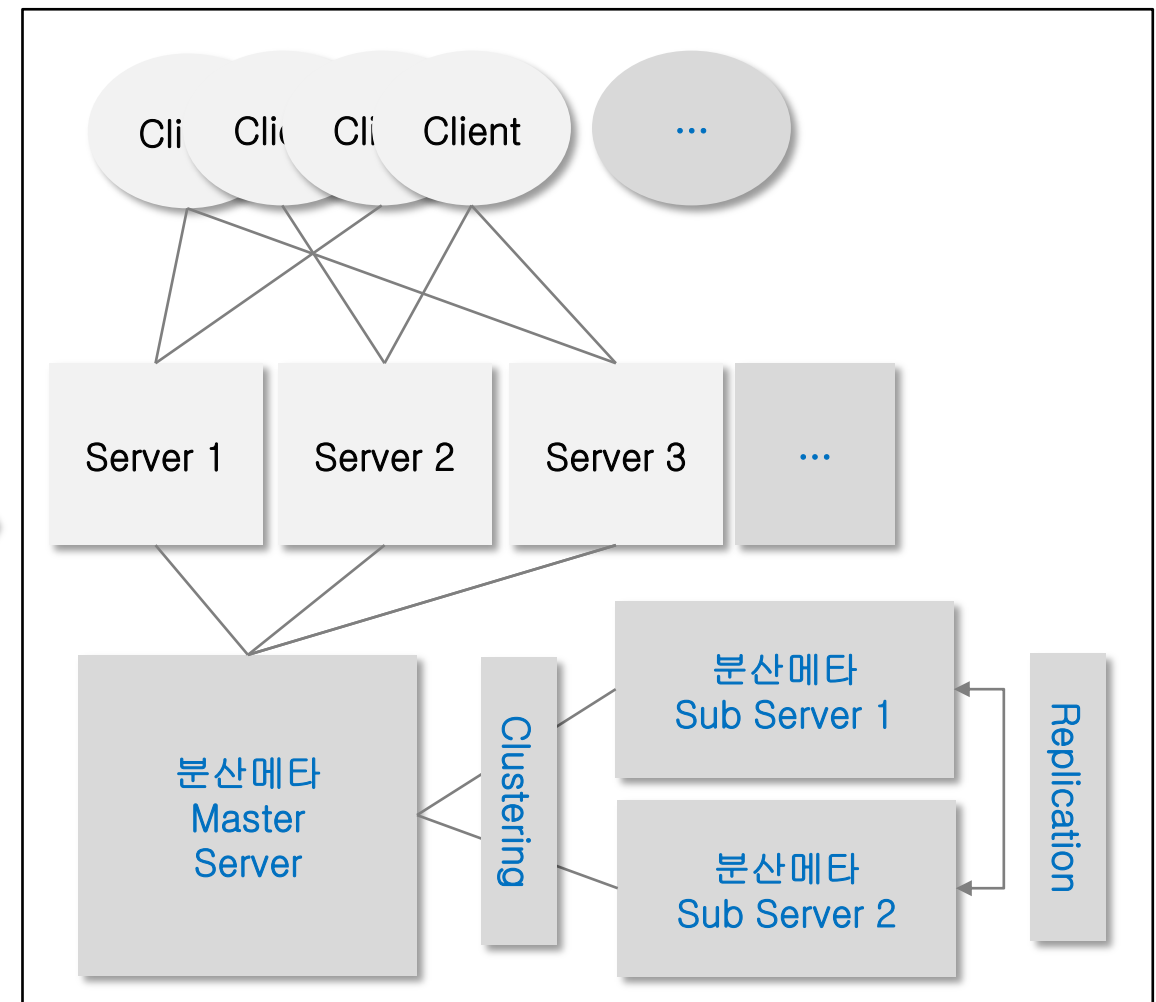
클라우드기반 대용량 트래픽을 처리하기 위한 확장성, 안정성을 고려한 분산 메타 적용

[일반적 아키텍처]



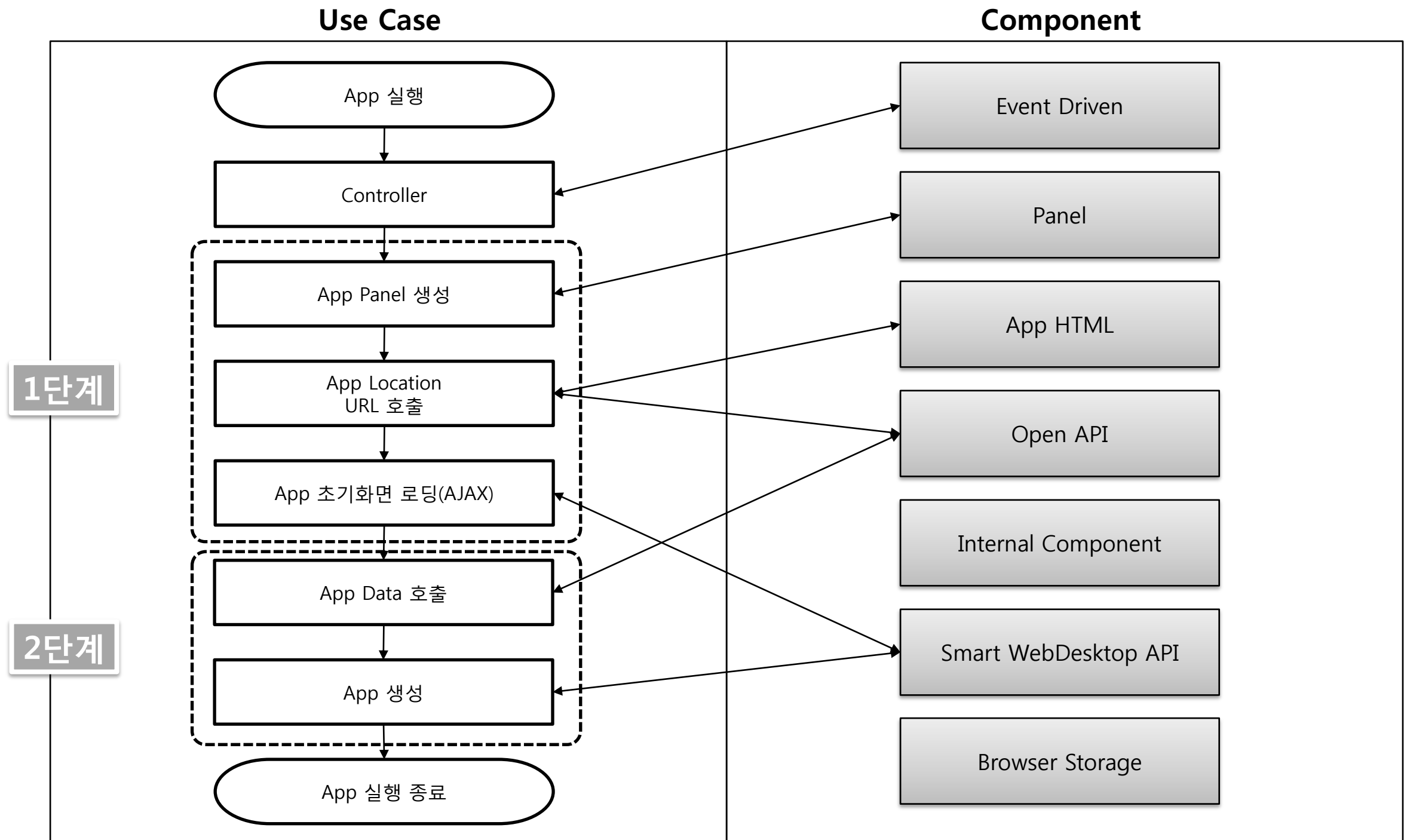
- 메타 정보를 일반적인 DB 및 File 방식 사용
- 확장성, 안정성, 데이터 정합성에 문제 발생

[분산 메타서버가 적용된 아키텍처]



- Smart WebDesktop의 메타 정보를 분산 메타 데이터 서버에 저장 및 관리
- 확장성, 안정성, 데이터 정합성 문제를 해결

10. Use Case : Application 실행(공통)



감사합니다

