

플랫폼 개발 생명주기(4)

(PDLC: Platform Development Life Cycle)

3 단계 : 설계 - 플랫폼 구조도 설계

✓ 플랫폼 구조도 특징

- 플랫폼의 동작원리 및 구성요소를 한눈에 볼 수 있음
- 이해당사자와 플랫폼 간의 관계가 표현되어 있음
- 한 개 이상의 모듈로 구성되어있으며 각 모듈의 상호관계가 표현되어 있음
- I~3단계(계획, 분석, 설계) 과정이 모두 포괄적으로 표현되기 때문에 플랫폼 개발에서 가장 중요한 작업
- 개발자의 필요에 따라 논리적 또는 물리적으로 설계될 수 있음

✓ 플랫폼 구조도 작성 시 고려해야 할 점

- ① I~2단계(계획, 분석)에서 조사 및 분석한 내용이 명확하고 정확하게 표현되었는가?
- ② 구성요소는 적절히 배치되었는가? - 구성요소의 개수, 상호연관, 모듈화 등
- ③ 환경이 적절한가? - 제안하는 플랫폼과 이해당사자 및 시장과의 연관성 등
- ④ 경계가 명확한가? - 시스템/모듈/기능의 경계(레벨) 표현, 플랫폼과 이해당사자의 경계 표현 등

3 단계 : 설계 - 플랫폼 구조도 설계

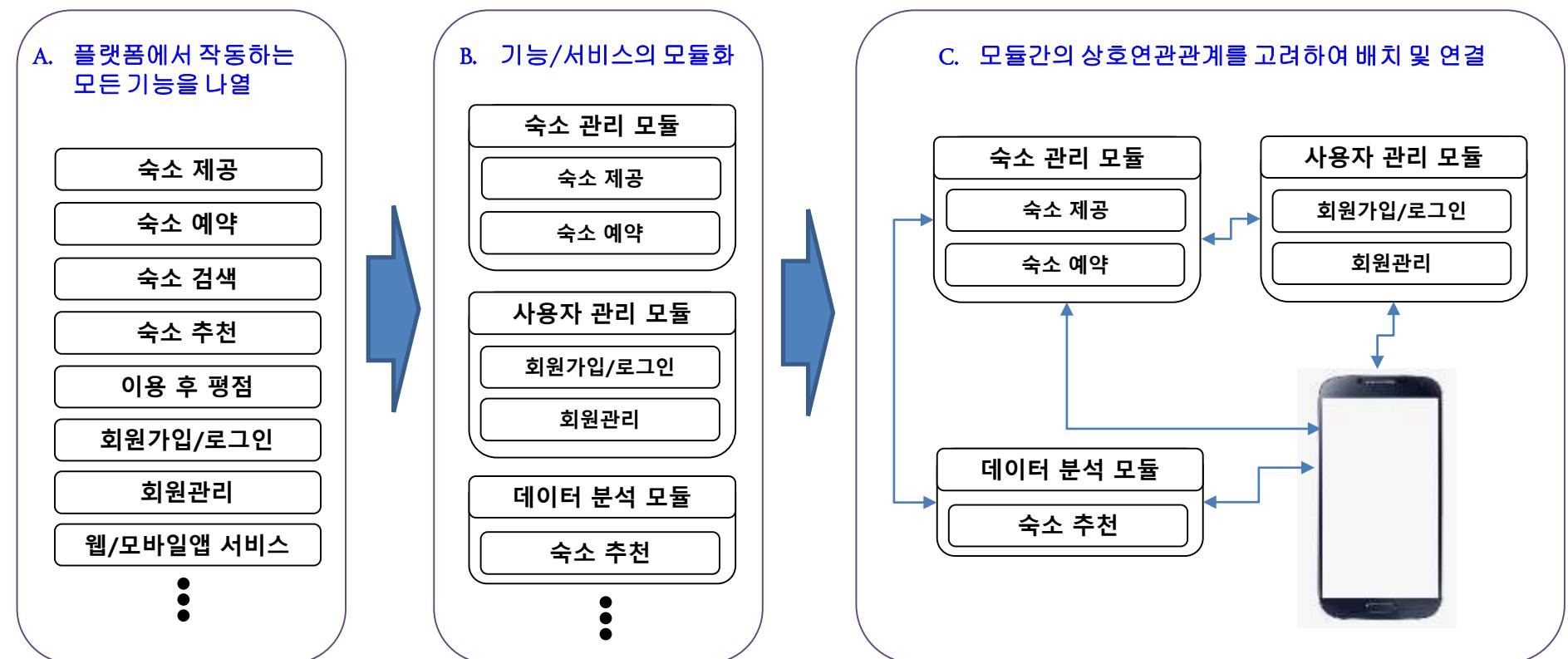
① 1~2단계(계획, 분석)에서 조사 및 분석한 내용이 명확하고 정확하게 표현되었는가?

- 1단계 계획 및 선정과 2단계 분석에서 수집한 내용
 - 프로젝트 목표 : 프로젝트의 목표를 달성할 수 있는 플랫폼. 사용자의 요구, 기존의 문제점을 해소할 수 있는지에 대한 표현이 명확
 - 기술 보고서(Technical Report): 플랫폼에 적용할 기술을 용도와 방법에 맞게 구조도에 표현
 - 상황도 : 구조도의 전체적인 틀이 되는 것으로 환경적인 요소(이해당사자, 외부 API 등)와 플랫폼의 경계를 표현
 - 데이터 흐름도 및 서비스 모델 : 구조화(모듈화) 및 모델링된 플랫폼의 기능들을 토대로 구체적이고 기술적인 내용을 가미하여 구조도를 작성
- 위 정보를 토대로 구체적이고 포괄적인 구조도를 설계해야 하며, 단순히 위 정보들을 취합하는 것 이 아닌 명확하고 간단하게 플랫폼의 구조를 표현해야 함

3 단계 : 설계 - 플랫폼 구조도 설계

② 구성요소는 적절히 배치되었는가? - 구성요소의 개수, 상호연관, 모듈화 등

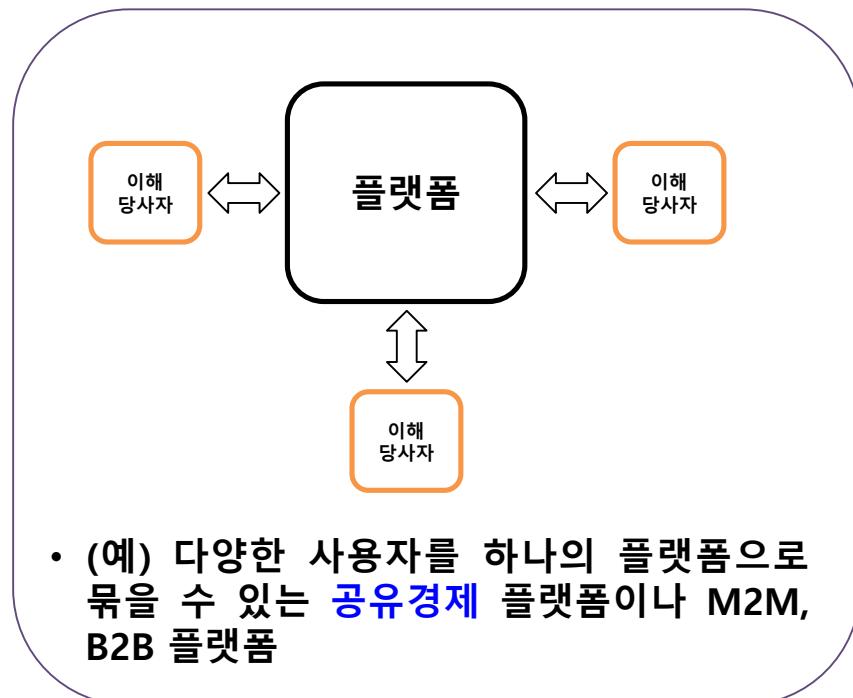
- 논리적 설계 내용을 토대로 플랫폼에서 발생할 수 있는 모든 활동/동작을 시스템 기능으로써 나열하고 유사한 기능들을 모듈화
- 모듈간의 상호연관관계를 고려하여 명확하고 정확하게 배치 및 연결



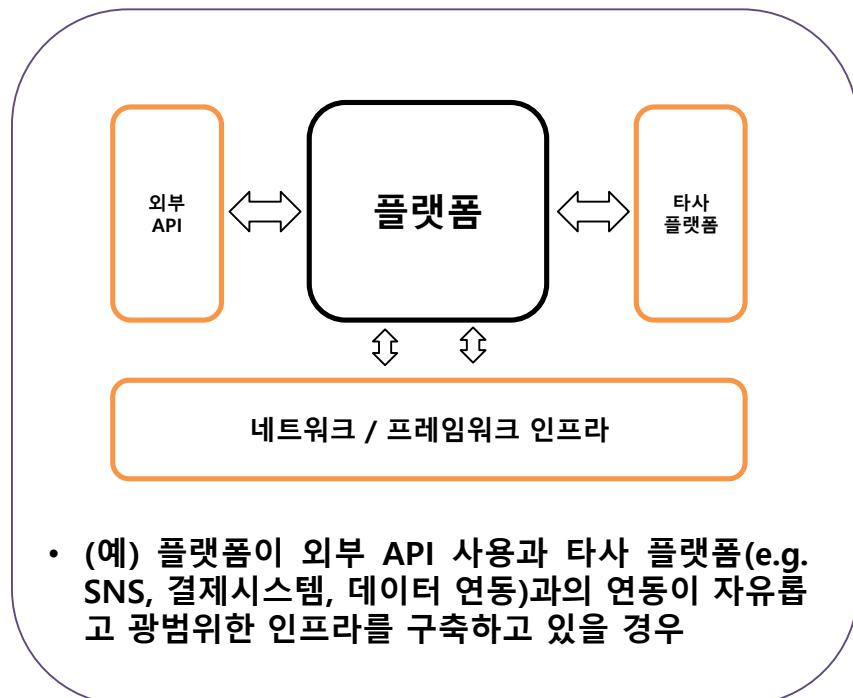
3 단계 : 설계 - 플랫폼 구조도 설계

③ 환경이 적절한가? - 제안하는 플랫폼과 이해당사자 및 시장과의 연관성 등

- 플랫폼의 주변 환경(이해당사자, 외부 API, 시스템/플랫폼 등)을 고려하여 작성
- 플랫폼이 제공하는 서비스/상품의 시장에서 연관될 수 있는 요소들을 고려하여 활용



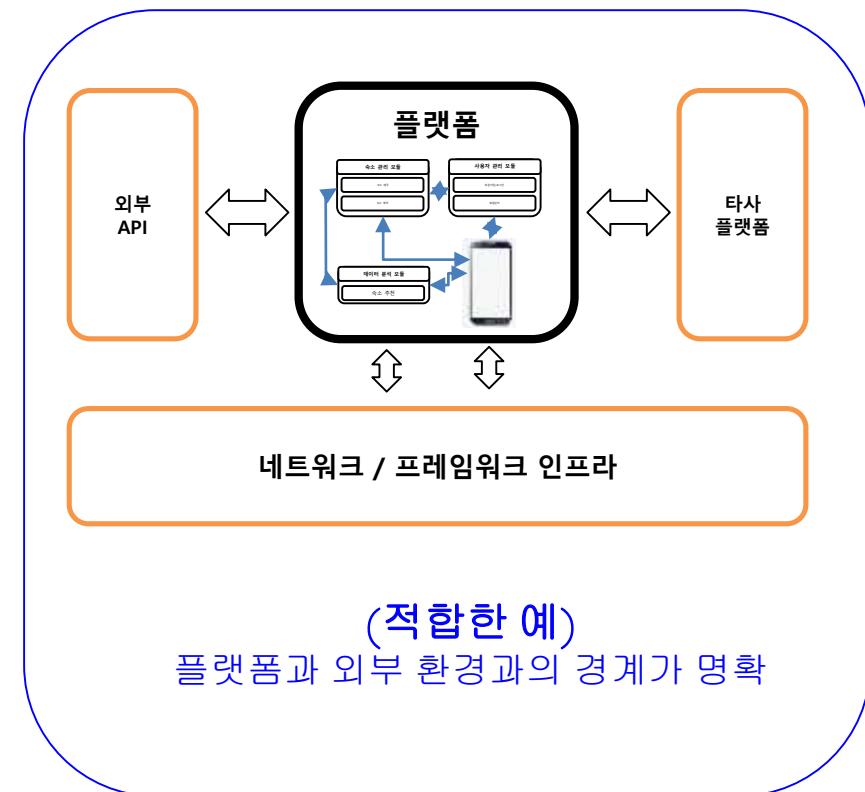
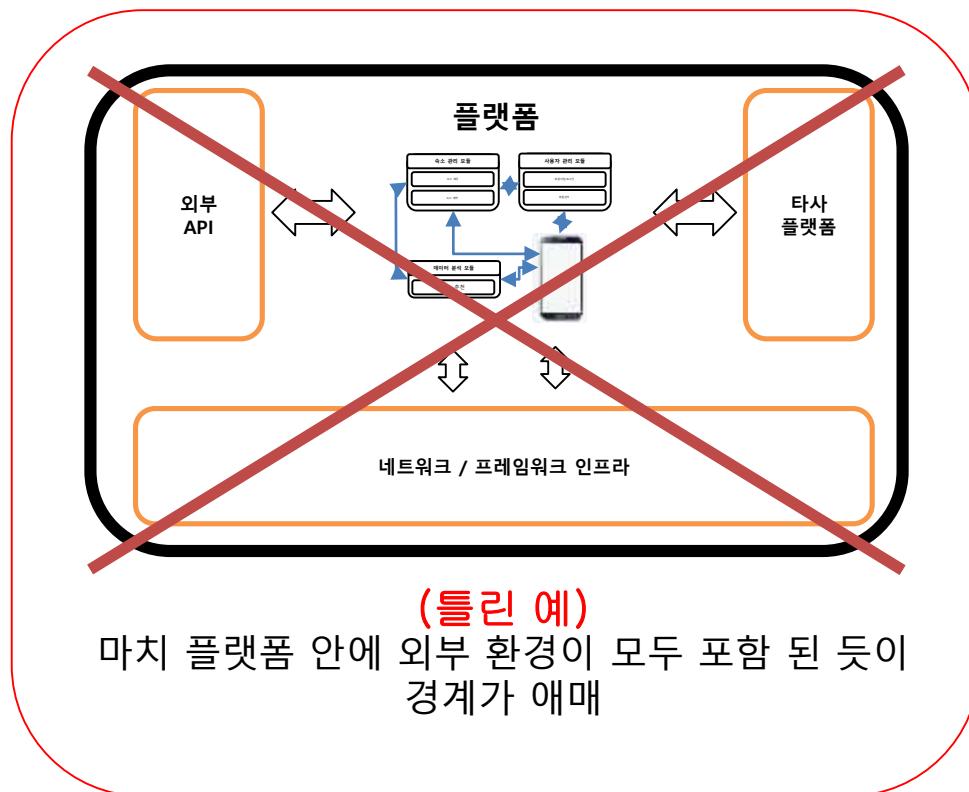
- (예) 다양한 사용자를 하나의 플랫폼으로 묶을 수 있는 **공유경제** 플랫폼이나 M2M, B2B 플랫폼



- (예) 플랫폼이 외부 API 사용과 타사 플랫폼(e.g. SNS, 결제시스템, 데이터 연동)과의 연동이 자유롭고 광범위한 인프라를 구축하고 있을 경우

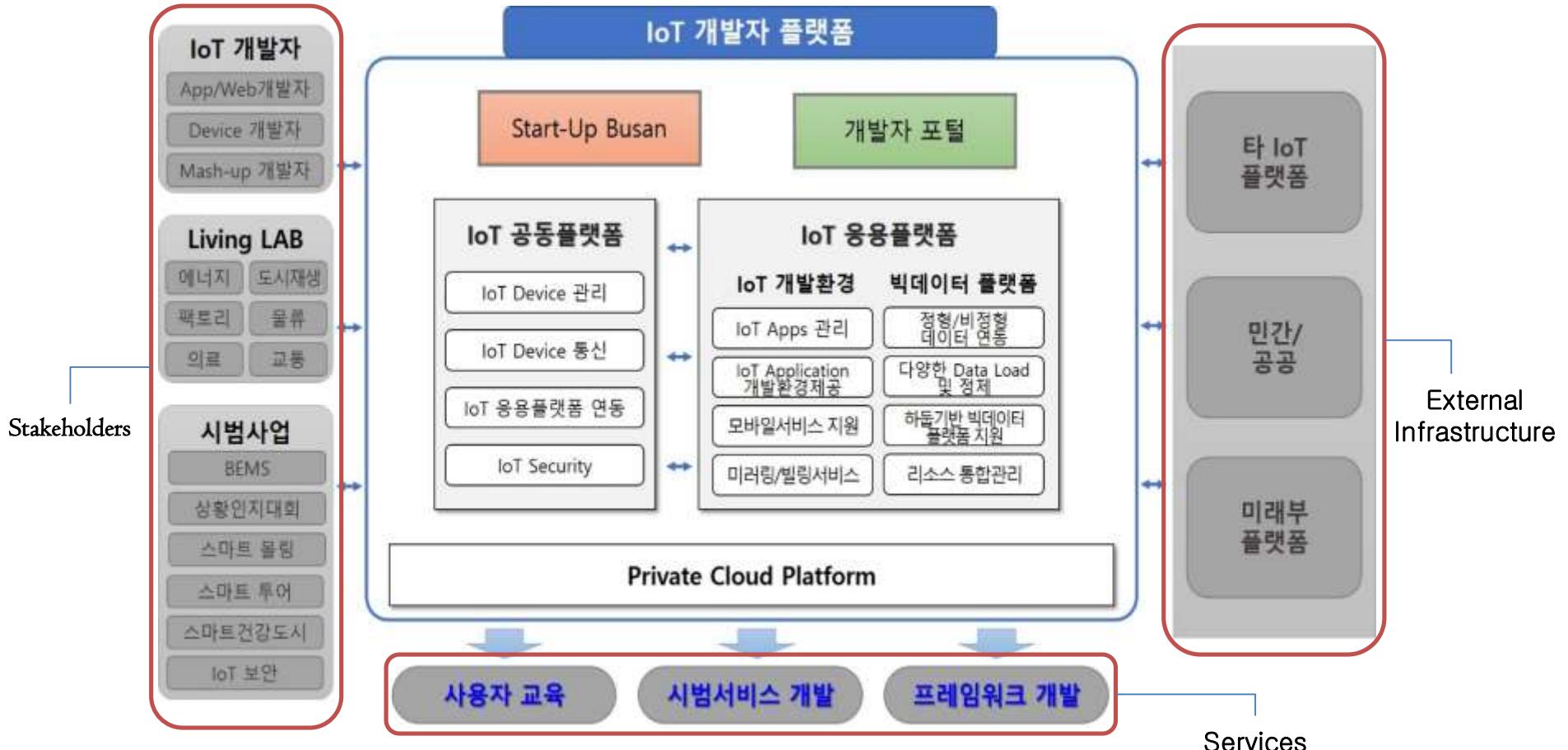
3 단계 : 설계 - 플랫폼 구조도 설계

- ④ 경계가 명확한가? - 시스템/모듈/기능의 경계(레벨) 표현, 플랫폼과 이해당사자의 경계 표현 등
- 플랫폼 구조도에서 플랫폼과 외부환경과의 경계가 명확해야 함



Step 3 : Design - Design Platform Architecture(examples)

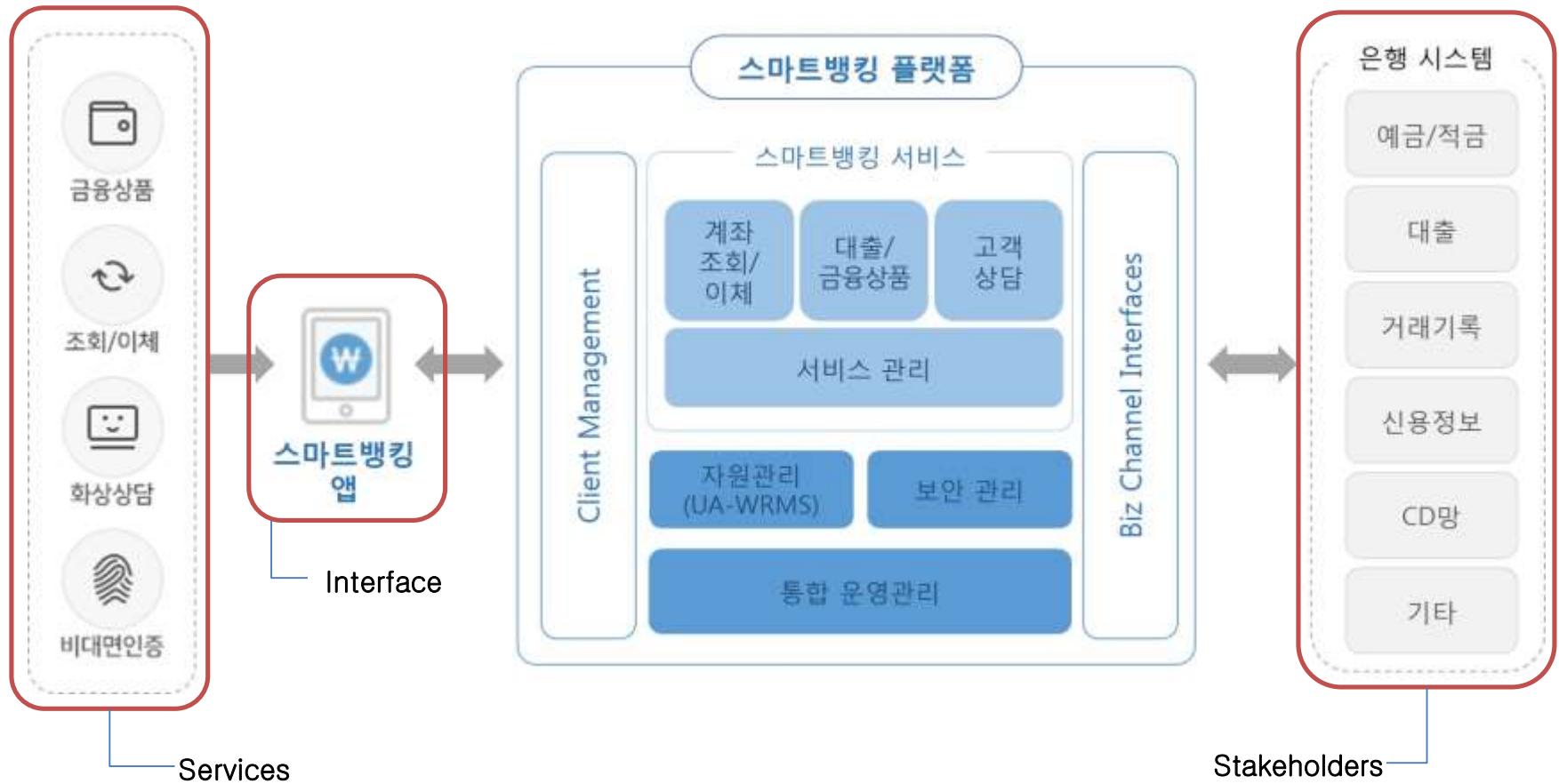
✓ Example of designing of platform architecture (1)



<IoT developer platform architecture case>

Step 3 : Design - Design Platform Architecture(examples)

✓ Example of designing of platform architecture (2)



4 단계 : 구현 및 운영

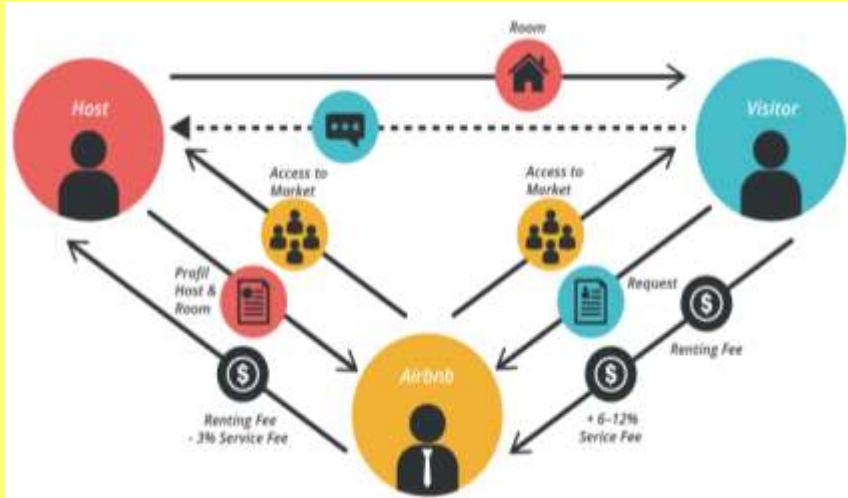
- ✓ 7개의 주요 활동(코딩, 테스트, 설치, 문서화, 교육, 지원, 유지보수)으로 구성
- ✓ 코딩, 테스트, 설치
 - 설계된 플랫폼 구조에 따라 플랫폼을 구성하는 프로그램 코딩
 - 코딩한 프로그램에서 오류를 찾고 수정하기 위한 테스트 진행
 - 프로그램 작동을 위한 설치(사용자 – 앱, 프로그램 등 / 관리자 – 서버)
- ✓ 기타 플랫폼 구현 활동
 - 플랫폼 개발에서 발생한 모든 문서 산출물의 마무리 작업을 진행. 이 과정에서 플랫폼 프로젝트 최종 기술 보고서 혹은 최종 보고서가 작성되어야 함
 - 플랫폼의 운영 및 유지보수를 위한 조직 내 인력에 대한 교육과 지속적인 사용자 보조 및 지원
- ✓ 플랫폼 운영 및 유지보수
 - 플랫폼 운영 과정에서 발생하는 문제점, 개선점을 파악하고 사용자 요구 및 비즈니스 환경의 변화를 반영
 - PDLC의 순환구조에 따라 문제점 보완 및 개선에 대해 계획, 분석, 설계, 개발 단계를 반복

※ 주요 산출물 : 개발 코드/문서, 교육 훈련 절차와 지원 기능, 완성된 소프트웨어, 유지보수 계획

❖ 대표적인 시스템 개발 방법론

- ✓ Agile : 공학 설계 방법에 적합한 규정된(predictive) 방법론이 아닌 상황(사람, 환경 등)에 맞게 유기적으로 판단하여 시스템 개발 활동을 하는 적응적(adaptive) 방법론. 계획과 개발, 출시와 같은 개발 주기가 여러 번 반복되며, 개발 환경에 맞게 요구사항이 추가되거나 변경 (민첩)
- ✓ Waterfall Model : 각 단계를 확실히 매듭짓고 다음 단계로 진행. ① 타당성을 분석하고, ② 사용자의 기능·성능·신뢰도 등에 대한 요구를 분석하며, ③ 소프트웨어를 설계하고, ④ 프로그래밍을 한 뒤, ⑤ 통합 테스트를 거쳐, ⑥ 소프트웨어를 운용하고 유지·보수시키는 등의 단계
- ✓ DevOps : 개발(Development)과 운영(Operations)의 합성어로 시스템 개발과 운영을 병행 및 협업하는 방식. 소프트웨어 개발과 운영관리 간의 협업과 통합을 담당하며 빅데이터 비즈니스가 각광을 받으면서 한 분야의 뛰어난 능력을 보유한 인력 뿐만 아니라, 통계, 프로그래밍, 개발, 오퍼레이션과 같이 융합적인 능력에 대한 요구가 증가함

❖ Business model cases

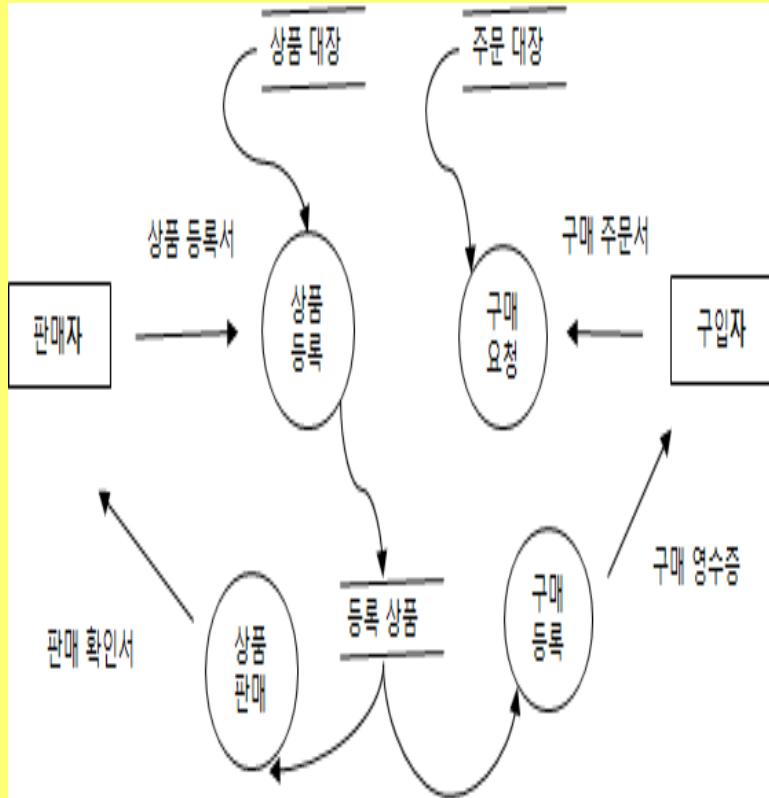


Airbnb business model

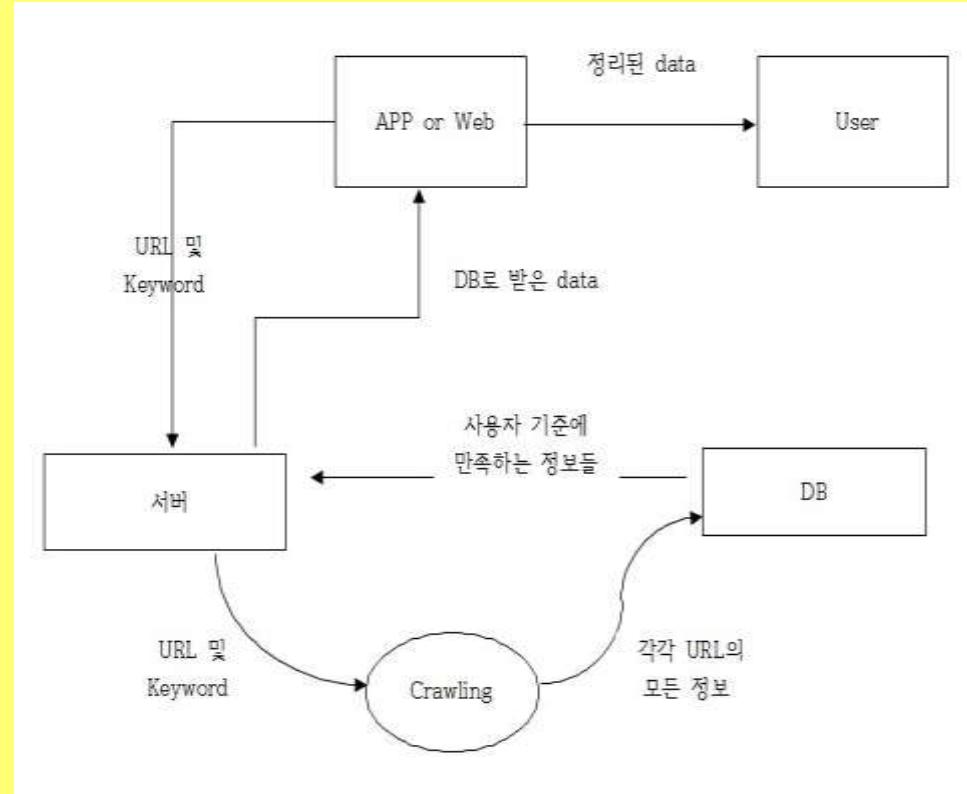


Uber business model

❖ Data flow diagram(service model) cases

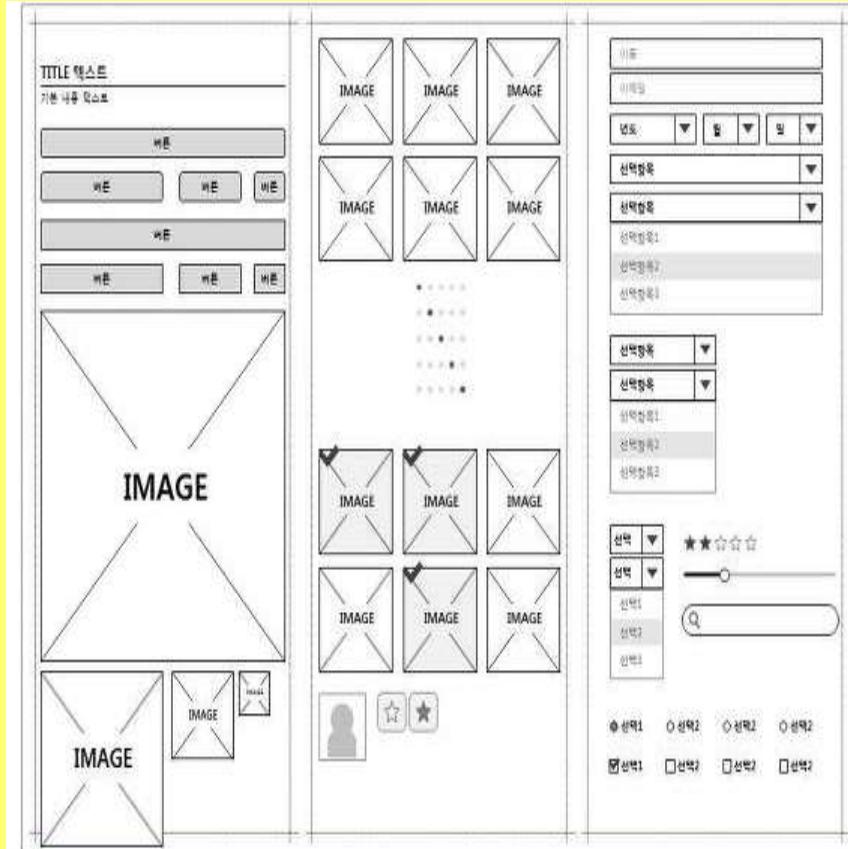


Data flow diagram(service model) of E-Commerce

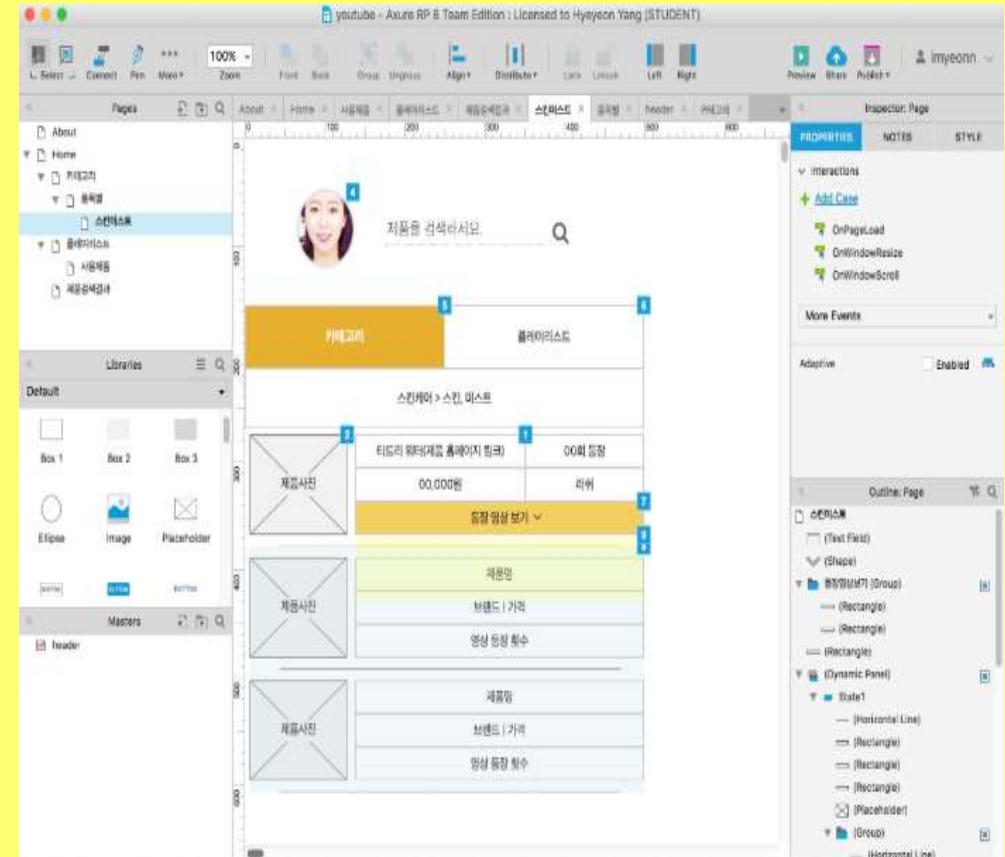


Data flow diagram(service model) of search engine

❖ User Interface(UI) cases



Designing of Mobile Application UI



Designing of Web Shopping mall UI