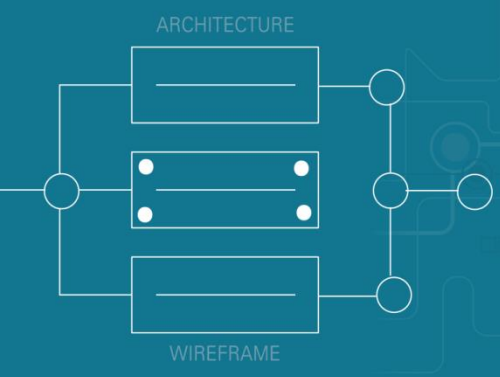
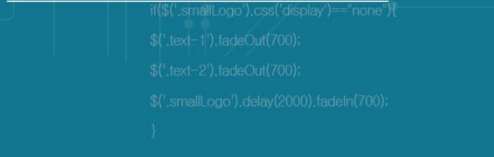


화면 구현 part 1



아키텍처 설계 프로세스

KOREATECH
온라인평생교육원

학습목표

- 화면 구현 및 아키텍처를 이해하고 설명할 수 있다.
- 아키텍처 모델, 역할, 뷰를 설명할 수 있다.
- 아키텍처 설계 과정을 이해하고 상세 절차대로 설계할 수 있다.

학습내용

- 화면 구현 및 아키텍처의 이해
- 아키텍처 모델, 역할, 뷰
- 아키텍처 설계의 중요성 및 원칙

화면 구현의 이해

1. 화면 구현 개요

화면 구현

우리가 일상에서 눈으로 보는 다양한 서비스

우리가 사용할 수 있도록 만들어 주는 것

예 TV 화면, 광고 전광판, 스마트폰 화면, 지하철 목적지 안내

화면 구현이란?

- 우리가 일상에서 눈으로 보는 다양한 서비스를 만들어 내는 것

화면 구현의 이해

2. 소프트웨어에서의 아키텍처

아키텍처

- 건축에서 가장 많이 활용되는 의미로 설계, 뼈대 구성

소프트웨어

- 건축과 달리 무형으로 존재하는 다양한 동작들로 구성

소프트웨어에서의 아키텍처란
무형으로 존재하는 다양한 동작들을
어떻게 구현할지 설계하고 그 뼈대를 만들어 내는 것

화면 구현의 이해

3. 소프트웨어에서의 화면 구현

- **보기만 하는 화면**: TV, 영화 등
- **보는 것 + 사용자 입력**: 무인 발매기, 스마트폰 게임 등

소프트웨어는 다양한 사용자 입력과
그에 따른 출력을 화면에서 모두 구현

소프트웨어에서의 화면 구현은
입력과 출력을 모두 포함하는 개념

아키텍처의 이해

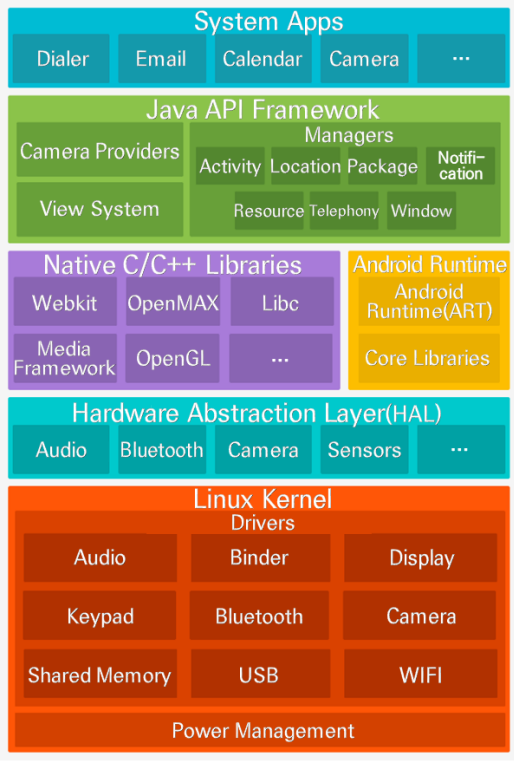
1. 아키텍처란?

1) 아키텍처 개념

비즈니스 요구 사항을 만족시키기 위한

전체 시스템의 구조로 정의

시스템을 구성하는
컴포넌트 정의



컴포넌트들 간의
관계 정의

컴포넌트가 다루는
데이터 정의

〈 안드로이드의 아키텍처 예시 〉

출처: https://developer.android.com

아키텍처의 이해

1. 아키텍처란?

2) 아키텍처 설계 고려사항

Check Point 1

사용자 요구 사항에 맞춰 설계

Check Point 2

확장 가능한 형태로 설계

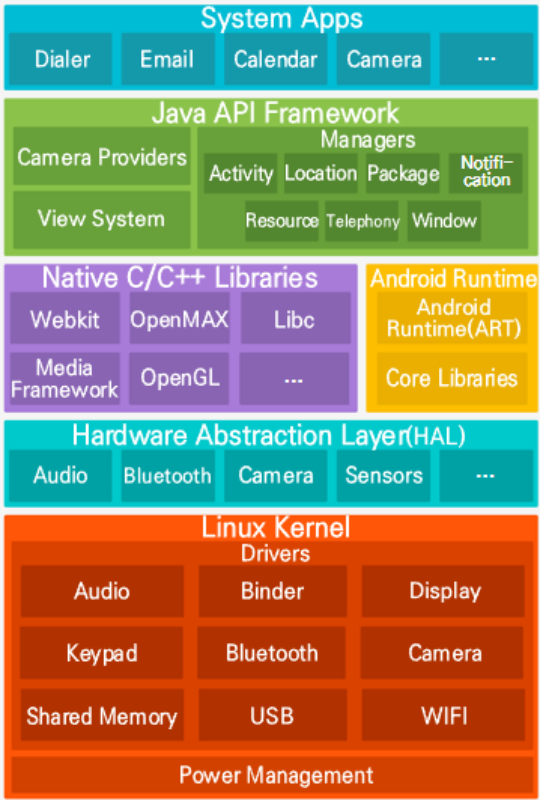
- 변화되는 비즈니스 전략에 대응

Check Point 3

시스템을 사용할 조직에 맞게 설계

- 조직의 기술 수준, 규모, 형태와 비즈니스 형태 분석

3) 아키텍처 특징



- 아키텍처는 소프트웨어 **요소 간의 관계 정보**를 가짐
- 하나 이상의 아키텍처 요소와 한 가지 이상의 연관 관계로 구성될 수 있음
- 시스템의 **공통성을 추상화** 시킴

  → 다양한 행동과 개념, 패턴, 접근방법, 결과 등을 나타냄
- 외부에 드러나는 시스템 요소의 행위는 다른 시스템 요소와의 상호 작용 방법을 제시
- 시스템의 **전체적인 구조**를 표현

아키텍처의 이해

2. 아키텍처 구성요소

참조 모델 | 아키텍처 패턴 | 참조 아키텍처 | 소프트웨어 아키텍처

1) 참조 모델

참조 모델

- 비즈니스 또는 시스템 문제를 해결하는 데 참여하는 일반적인 기능의 구분

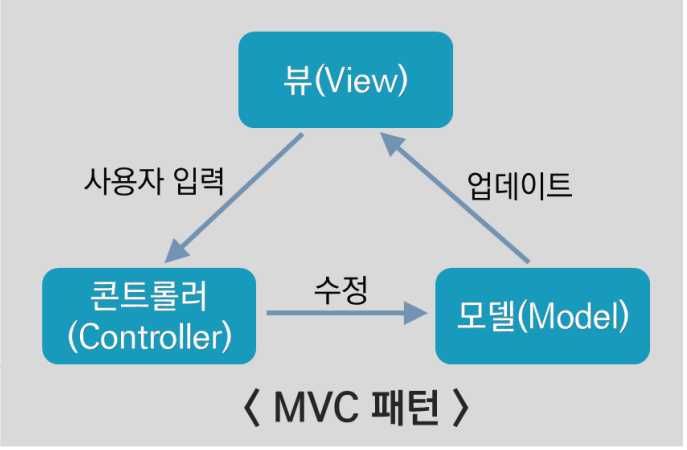
아키텍처의 이해

2. 아키텍처 구성요소

2) 아키텍처 패턴

아키텍처 패턴

- 아키텍처에 있는 일련의 제약사항을 표현, 시스템 구조를 체계적으로 구성하기 위한 기본적인 스키마
- 클라이언트-서버 구조,
  MVC 패턴,
  마스터 슬레이브 패턴 등



〈 MVC 패턴 〉

3) 참조 아키텍처

참조 아키텍처

- 참조 모델을 구성하는 요소들을 소프트웨어로 구현한 단위와 이들 간의 데이터 흐름
- 각 참조 모델의 기능에 대해 시스템 또는 애플리케이션의 분할된 구조
  - ✓ 참조 모델은 기능 분할된 구조를 나타냄

아키텍처의 이해

2. 아키텍처 구성요소

4) 소프트웨어 아키텍처

소프트웨어 아키텍처

- 목표 시스템의 기능·비기능적 요구 사항, 기술적, 자원적 제약 등을 고려하여 하나 이상의 참조 아키텍처를 수정·보완

3. 아키텍처 중요성

- **간략화**
  - 이해하고 **추론**할 정도의 간결성 유지
- **추상화**
  - 추상적인 표현을 사용하여 **복잡도 관리**
- **가시성**
  - 시스템이 포함해야 할 것들을 **시각적으로 표현**
- **관점 모형**
  - **이해당사자의 관심사**에 따른 모형 제시
- **의사소통 수단**
  - 이해당사자 간 **원활한 의사소통**의 수단으로 이용

아키텍처 모델, 역할, 뷰

1. 개념 모델(IEEE1471-2000)

이해관계자

- 시스템 아키텍처의 이해관계자를 의미
- 종류: 아키텍처 개발자, 시스템운영자, DBA, 최종사용자, 경영자, PM, 비즈니스 분석가 등

관심

- 기능 요구 사항과 비기능 요구 사항
- 정적인 구조와 동적인 동작
- 개념, 논리, 물리 레벨의 관점 및 표시
- 성능이나 보안, 서비스 내용 및 배치 등 개발에 관계되는 요소

뷰포인트

- 뷰를 기술 및 분석하기 위한 모델
- 모델에 적용하는 모델링 기법, 분석 기법 등을 규정

뷰

- 여러 이해관계자의 관심사에 본 시스템의 전체를 표현

모델

- UML 다이어그램 등

아키텍처 모델, 역할, 뷰

1. 개념 모델(IEEE1471-2000)

IEEE1471-2000란?

- IEEE 1471는 소프트웨어 구조에 대한 기술을 규정한 IEEE 표준
- 'ISO/IEC 42010, 시스템과 소프트웨어 공학 – 구조 기술'을 대체

2. 아키텍처 역할

1) 고객 대응

고객 대응

- 시스템의 품질 속성을 도출하고 Trade-off 분석을 통해 구현 가능한 아키텍처를 제시
- 의사결정을 조정하고 근거자료 확보를 위해 고객 및 시스템 사용자와의 의사소통을 담당

아키텍처 모델, 역할, 뷰

2. 아키텍처 역할

2) 프로젝트 관리 파트 대응

프로젝트 관리 파트 대응

- 개발 프로세스의 일정을 계획할 때 아키텍처 관점에서 의견 제공
- 개발 위험 요소 파악
  → 관련 의사 결정권자에게 통지
  → 이를 완화 시키는 방법을 강구

3) 개발 파트 대응

개발 파트 대응

- 개발팀과 도출된 아키텍처를 구현하는 데 고려해야 하는 여러 가지 설계 이슈 가이드 제공
- 리뷰에 관한 의사소통 개발

아키텍처 모델, 역할, 뷰

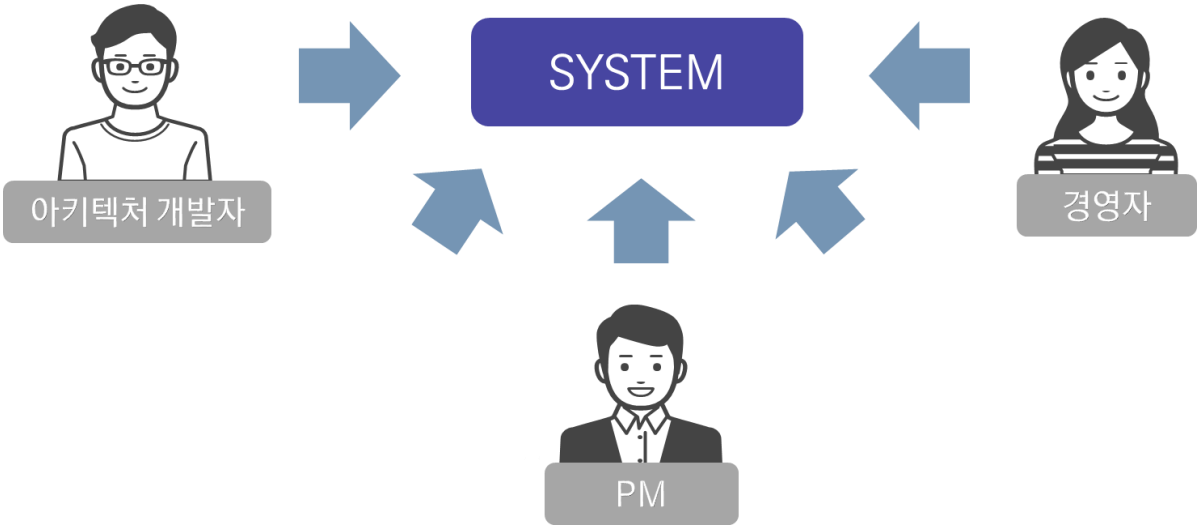
2. 아키텍처 역할

4) 솔루션 담당

솔루션 담당

- 아키텍처 구현을 위해 사용되는 다수의 솔루션 적용에 필요한 기술적 또는 기능적 이슈에 대해 소통
- 솔루션 벤더, 컨설턴트 등과 의사소통을 담당

3. 아키텍처 뷰(View)



시스템의 여러 가지 측면을 고려하기 위해
다양한 관점을 바탕으로 정의

아키텍처 모델, 역할, 뷰

3. 아키텍처 뷰(View)

Perry and Wolf's Model | Shaw and Garlan's Model | 4+1 View Model

Perry and Wolf's Model | Shaw and Garlan's Model | 4+1 View Model

- **요소(Element)**: 프로세싱(Processing), 데이터(Data), 연결(Connecting)
- **표현법(Form)**: 속성과 관계로 나타냄
- **근거(Rationale)**: 아키텍처를 정의하는데 고려되는 다양한 선택에 대한 근거(기능성, 성능, 신뢰성, 경제성)

Perry and Wolf's Model | Shaw and Garlan's Model | 4+1 View Model

- **컴포넌트(Component)**: 할당된 임무를 제공하는 실행 가능한 요소
- **커넥터(Connector)**: 컴포넌트 간 상호작용 중재자
- **패턴(Patterns)**: 컴포넌트와 커넥터가 조합되는 방법에 대한 제약사항

아키텍처 모델, 역할, 뷰

3. 아키텍처 뷰(View)

Perry and Wolf's Model | Shaw and Garlan's Model | 4+1 View Model

- **사용자 사례 관점(Use Case View)**: 시스템의 외부 사용자 관점에서 사용 사례와 이들 간의 관계를 정의, 시스템 아키텍처를 도출
- **설계 관점(Design View)**: 상위 수준에서 시스템의 논리적인 구조와 행위 클래스인터페이스 협력관계로 정의, 시스템의 기능 요구 사항을 설명
- **구현 관점(Implementation View)**: 시스템의 병렬 처리 및 동기화 처리를 위한 스레드와 프로세스를 정의
- **프로세스 관점(Process View)**: 독립적으로 실행되는 컴포넌트와 이들 간의 관계를 정의
- **배치 관점(Deployment View)**
  : 실행되는 시스템 하드웨어와 소프트웨어의 관계를 정의

아키텍처 설계의 중요성 및 원칙

1. 아키텍처 설계의 중요성

여러 이해관계자 간의
의사소통 수단

시스템을 효과적으로 관리할 수 있는 수준에서
고려사항을 표현하고 조율하는 **공용어**

초기 설계 **의사 결정**
방향에 대한 선언

고려사항들의 우선순위를 분석할 수 있는 **최초 산출물**
성능, 개발비용과 품질 간의 절충 등 모두 **아키텍처에 따라 결정**

아키텍처 설계의 중요성 및 원칙

1. 아키텍처 설계의 중요성

재사용 가능하며
타 시스템에도 적용
가능한 시스템에 대한
추상화된 표현

요구 사항이 유사한 시스템에 적용할 수 있으며
이를 통해 **재사용**, 소프트웨어 **제품 라인을 구성**

2. 아키텍처 설계 순서

시스템 환경의 이해 → 요구 사항 분석 → 아키텍처 분석 → 아키텍처 설계 → 검증 및 승인

아키텍처 설계의 중요성 및 원칙

2. 아키텍처 설계 순서

1) 시스템 환경의 이해

- **현재 상황**을 이해하고 분석
- 앞으로의 환경 변화에 따른 **미래 요구 사항 예측**

추가적인 아키텍처 품질 속성 및 요구 사항을 파악

아키텍처에 **영향**을 미치는 요소는?

- 기술적 환경
- 배경과 경험
- 개발 조직
- 이해관계자
- 품질 요구 사항
- ● ●

아키텍처 설계의 중요성 및 원칙

2. 아키텍처 설계 순서

2) 요구 사항 분석

이해관계자의 다양한 요구 사항을 이해하고

분석, 소프트웨어 품질 요구 사항을 정형화하여 명세

요구 사항을 정형화하여 명세하는 방법은?

- 요구 사항 취득 → 식별 → 명세 → 분류 → 검증
- 기능적·비기능적 요구 사항 분류 및 명세

아키텍처 설계의 중요성 및 원칙

2. 아키텍처 설계 순서

3) 아키텍처 분석

아키텍처 분석 1

품질 요소 식별

- 기능성, 신뢰성, 효율성, 유지 보수성, 이식성을 고려

아키텍처 분석 2

품질 요소 우선순위 결정

- 품질 요소의 목표 및 영향도 식별
- 품질 시나리오 작성

아키텍처 분석 3

전술 개발

- 품질 속성별 개발 및 명세

4) 아키텍처 설계

아키텍처 설계 1

관점 정의

- 이해당사자 파악 및 이해당사자별 관점 정의

아키텍처 설계 2

뷰(View) 정의

- 시나리오로 표현된 품질 요구 사항
  → 아키텍처 패턴, 설계 전술 결정
  → 실체화 하고 뷰 작성

뷰(View)의 종류?

Module View, Component Connector View, Allocation view, Code View

아키텍처 설계의 중요성 및 원칙

2. 아키텍처 설계 순서

4) 아키텍처 설계

아키텍처 설계 3

아키텍처 스타일 선택

- Pipe-Filter, MVC, Layer 등 여러 패턴을 혼용하여 적용

아키텍처 설계 4

후보 아키텍처 도출

- Context Diagram 및 각종 뷰별로 다이어그램 작성
- SAD(Software Architecture Description) 기술

5) 검증 및 승인

다양한 설계 고려사항이

합리적으로 결정되었는지 확인하고 검증

아키텍처 설계의 중요성 및 원칙

2. 아키텍처 설계 순서

5) 검증 및 승인

- **아키텍처 평가**

  : 요구 사항 만족 적합성 평가, 품질 속성 간 Tradeoff 관계 평가(ATAM)

  Architecture Tradeoff Analysis Method

- **아키텍처 상세화**

  : 반복적으로 진행하며, 설계 메커니즘 도출 및 디자인 패턴 고려

  Persistency, Transaction 등

- **아키텍처 승인**

  : 고객 및 이해당사자 최종 승인

3. 아키텍처 설계 상세 절차

1) 요구 사항 분석

아키텍처 설계의 중요성 및 원칙

3. 아키텍처 설계 상세 절차

1) 요구 사항 분석

요구 사항 검토

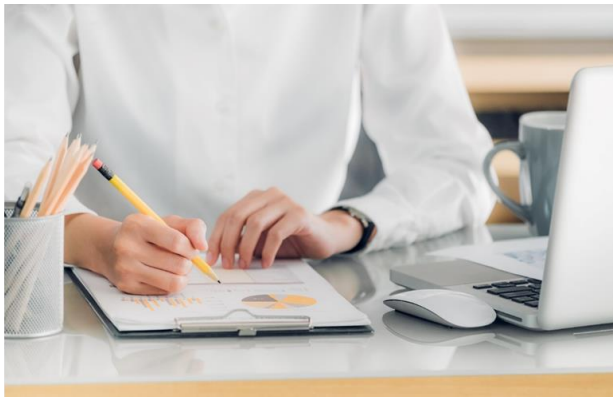
- 활동 및 역할 소개
- 비즈니스 목표 이해
- 시스템 환경 이해

중요 속성 식별

- 중요 기능 요구 사항 식별
- 핵심 물질 속성 식별

시나리오 작성

- 시나리오 도출
- 시나리오 우선 순위화
- 시나리오 정제



아키텍처 설계의 중요성 및 원칙

3. 아키텍처 설계 상세 절차

2) 설계 뷰 작성

아키텍처 요구 사항 검토

아키텍처 실체화

아키텍처 정제 및 명세화

아키텍처 요구 사항 검토

- 아키텍처 요구 사항 확인
- 기능 요구 사항 확인
- 아키텍처 드라이버 식별

아키텍처 실체화

- 아키텍처 패턴 선정
- 모듈 분할 및 책임 담당
- 아키텍처 뷰 작성

아키텍처 설계의 중요성 및 원칙

3. 아키텍처 설계 상세 절차

2) 설계 뷰 작성

아키텍처 정제 및 명세화

- 인터페이스 및 모듈 정제
- 아키텍처 검토 및 반복

3) 설계 검증

아키텍처 이해

아키텍처 분석

아키텍처 검증

아키텍처 설계의 중요성 및 원칙

3. 아키텍처 설계 상세 절차

3) 설계 검증

아키텍처 이해

- 활동 소개 및 역할 소개
- 비즈니스/아키텍처 목표 소개
- 작성된 아키텍처 소개

아키텍처 분석

- 아키텍처 접근 방법 식별
- 품질 속성 시나리오 작성
- 시나리오/아키텍처 상세 분석

아키텍처 검증

- 품질 속성 시나리오 검증
- 아키텍처 접근방법 검증
- 검증 결과 발표 및 문서화

핵심정리

1. 화면 구현 및 아키텍처의 이해

[화면 구현의 이해]

- 화면 구현: 우리가 일상에서 눈으로 보는 다양한 서비스를 만들어 내는 것
- 소프트웨어에서의 아키텍처: 무형으로 존재하는 다양한 동작들을 어떻게 구현할지 설계하고 그 뼈대를 만들어 내는 것
- 소프트웨어에서의 화면 구현: 입력과 출력을 모두 포함하는 개념

[아키텍처의 이해]

- 시스템에 대한 기본 조직 체계로, 시스템을 이루는 구성요소와 구성요소들 사이의 관계, 구성요소와 주변 환경들과 관계 및 시스템의 진화와 설계를 지배하는 원칙들로 실체화
- 이해하고 추론할 정도의 간결성을 유지하게 하며 추상적인 표현을 사용하여 복잡도를 관리하고, 시스템이 포함해야 할 것들을 가시화하여 이해당사자 간의 관심사에 따른 모형을 제시할 수 있고 원활한 의사소통 수단이 됨

핵심정리

2. 아키텍처 모델, 역할, 뷰

- 시스템에 관계되는 여러 이해관계자의 관점을 반영한 다양한 모델의 집합
- 아키텍처는 시스템 품질 속성을 실현하게 해주며 의사결정이 시스템의 진화에 따라 변경 타당성을 파악하고 관리하게 도와줌
- 아키텍처를 통해 이해당사자 간의 의사소통이 가능하며, PM과 아키텍트가 비용 및 일정을 예측할 수 있도록 도움

3. 아키텍처 설계의 중요성 및 원칙

- 아키텍처 구축 프로세스
  - ✓ 시스템환경의 이해 → 요구 사항 분석 → 아키텍처 분석 → 아키텍처 설계 → 검증 및 승인
  - ✓ 요구 사항 분석 단계에서는 이해관계자의 다양한 요구 사항을 이해하고 분석하며 요구 사항을 정형화하여 기능적, 비기능적 요구 사항으로 분류 및 명세 작업을 진행
- 설계 상세 절차: 요구 사항 분석 → 설계 뷰 작성 → 설계 검증