

SW아키텍처 패턴가이드

Part 1 : SW아키텍처 패턴가이드 소개

2015. 5. 19 [제131호]

Contents

- I. 개요
- II. 공통 아키텍처 프레임 워크
- III. 결론

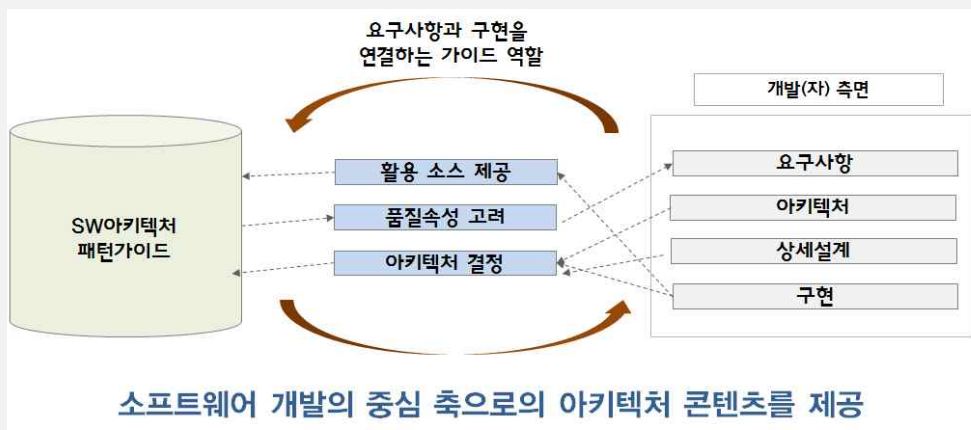
I. 개요

SW설계와 개발에 필수로 활용되는 SW아키텍처 패턴은 그 종류가 많고 또한 패턴의 정의 방법에서 일관성이 없어 특정 패턴을 기억해서 활용하는 것은 용이하지 않다. 이에 다양한 관점과 레벨에서 정의되고 있는 아키텍처 패턴들을 분류하는 방법 및 정의하는 내용을 체계화하는 것이 중요하다. 패턴의 검색 및 활용을 용이하게 하고 보다 더 쉽게 사용하는 것을 도와주는 SW아키텍처 패턴가이드를 소개한다.

1.1 SW아키텍처 패턴가이드란?

SW개발자가 개발단계에서 바로 적용할 수 있는 SW아키텍처 패턴을 도메인별, 아키텍처 패턴별, 품질속성별로 분류하여 아키텍처 패턴 템플릿, 프레임워크 가이드, 소스코드 샘플, 교육자료 등을 활용할 수 있도록 제공하는 서비스 이다.

아키텍처 패턴선정은 기업체와 연구소의 아키텍트, SW개발 PM급을 대상으로 주력 개발제품군, SW아키텍트 및 SW테스트 조직 활용정도, 개발방법론, 주요 개발언어, 개발 실행환경 및 IDE, 주요 아키텍처 패턴 등에 관하여 설문조사 한 데이터를 토대로 선정되었다.



1.2 SW아키텍처 패턴 가이드 구성

1.2.1 도메인 별 분류

개발 실행환경 및 IDE, 주요 아키텍처 패턴 등에 관하여 설문조사 한 데이터

를 토대로 임베디드, 엔터프라이즈, 서비스, 패키지, 금융으로 분류 하였으며, 도메인 별 아키텍처 패턴정보를 샘플시스템에 도입하여 체계적으로 구성 하였다.

표 1_

구 분	샘플 시스템
임베디드	· 항공기센서 데이터 가공 솔루션(Pipes & Filters) · 빌딩보안 시스템(Alarm센서 컨트롤 솔루션(Event&Driven)
엔터프라이즈	· 웹기반 전자상거래 시스템(Event& Driven)
서비스	· 전자상거래 시스템(Event& Driven)
패키지	· Stand-alone 그래픽 에디터 솔루션(Composite)
금융	· 은행 정보계 데이터 분석 솔루션(Pipes & Filters)

II. 공통 아키텍처 프레임 워크

공통아키텍처 프레임워크는 Event-Bus의 경우는, 빌딩보안시스템, Pipe & Filter는 학생정보시스템, Composite은 웹기반 공용 그래픽 편집기에 적용된다. 아래표는 공통 아키텍처프레임워크가 패턴이 적용된 시스템을 보여준다.

표 2_공통 아키텍처 프레임워크 패턴

단계	항목	빌딩보안시스템 (엔터프라이즈)	학생정보시스템 (엔터프라이즈)	웹기반 공용그래픽편집기 (Stand-alone)
아키텍처 공통 프레임워크	요구사항정의서 (HTML/EA)	Event-Bus	Pipe & Filter	Composite
	아키텍처설계서 (HTML/EA)			
	CODE 가이드 (HTML)			
	소스코드 (C, JAVA)			

공통아키텍처 프레임워크는 3개의 패턴을 제공하고 각 패턴은 요구사항 정의서와 아키텍처 설계서, Code 가이드 그리고 어플리케이션 코드를 제공한다.

2.1 아키텍처 요구사항 및 설계

공통 아키텍처 프레임워크에서 요구사항 및 설계는 HTML과 UML기반의 Notation을 제공한다. HTML의 경우, UML 도구가 없을 경우에 유리하며, 수정이 필요할 경우 UML툴을 설치하여야 한다.

공통 Framework와 관련된 요구사항과 아키텍처만 제공한다.

요구사항 정의서에는 아래 항목 등을 체계적으로 기술할 수 있도록 구체적인 예시를 제시한다.

2.1.1 요구사항 정의서

• 기능요구사항

– Use case diagram, Use case, Use case scenario, 주요기능 요구사항

그림 1_Use case daigram예시

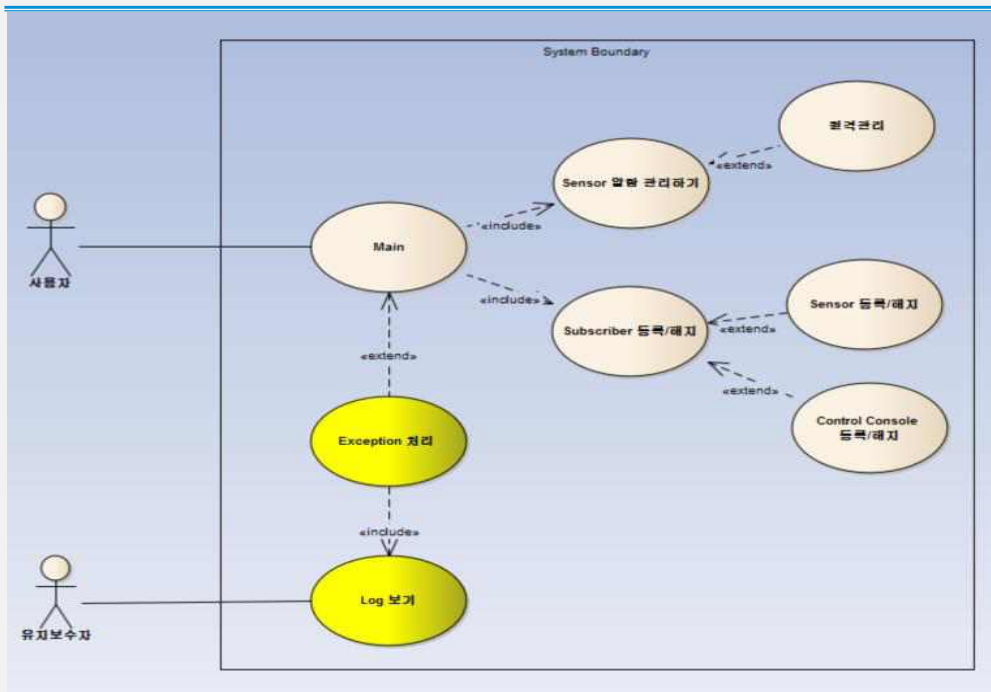


표 3_Use case 예시

이 름	설 명
Control Console 등록/해지	Event Bus에 Control Console을 등록(해지)한다.
Sensor 등록/해지	빌딩 보안을 책임지는 센서들을 등록(해지)한다.
Sensor 알람 관리	Sensor 알람을 발생을 확인하고 해제한다.
Exception 처리	빌딩 보안 시스템의 예외 사항을 처리한다.
Log 보기	빌딩 보안시스템의 로그를 기록하고 모니터링한다.

- 외부인터페이스요구사항
- 내부인터페이스요구사항
- 내부데이터요구사항
- 보안요구사항
- 소프트웨어품질요구사항

2.1.2 SW아키텍처 스타일

- SW아키텍처를 수립하는 수단으로 아키텍처 스타일을 사용하기도 하며, 아키텍처 스타일은 패턴 중 Runtime view, Module view, Allocation view에 자주 사용되는 패턴을 표준화 한 것이다.
- 각종 뷰를 작성할 때 참조할 수 있는 아키텍처 패턴으로 Call & Return style, Layer style, Uses style, Decomposition style이 많이 쓰이고 있다.
아래 그림은 Runtime view, Module view, Allocation view의 예시를 보여준다.

그림 2_Runtime View

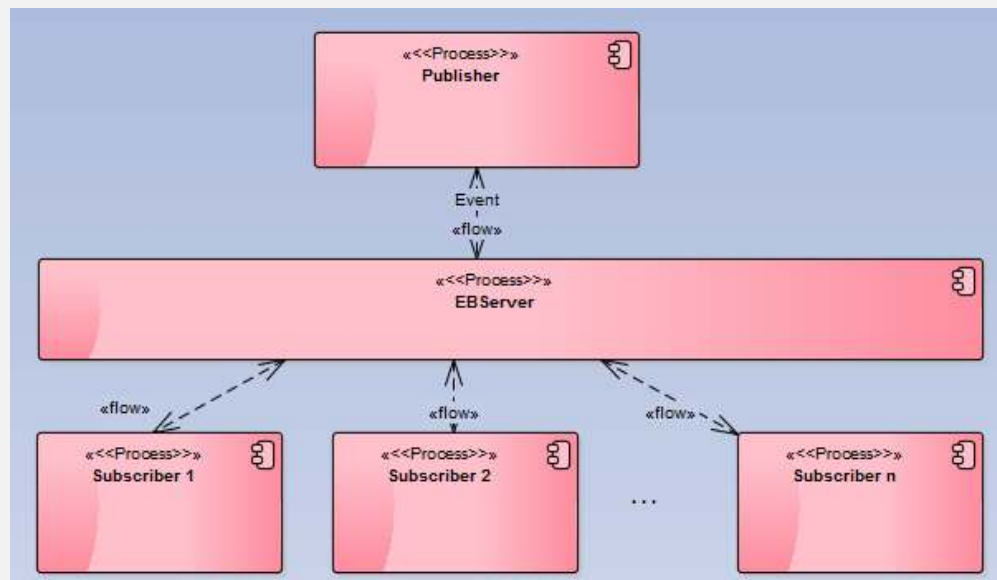


그림 3_Module View

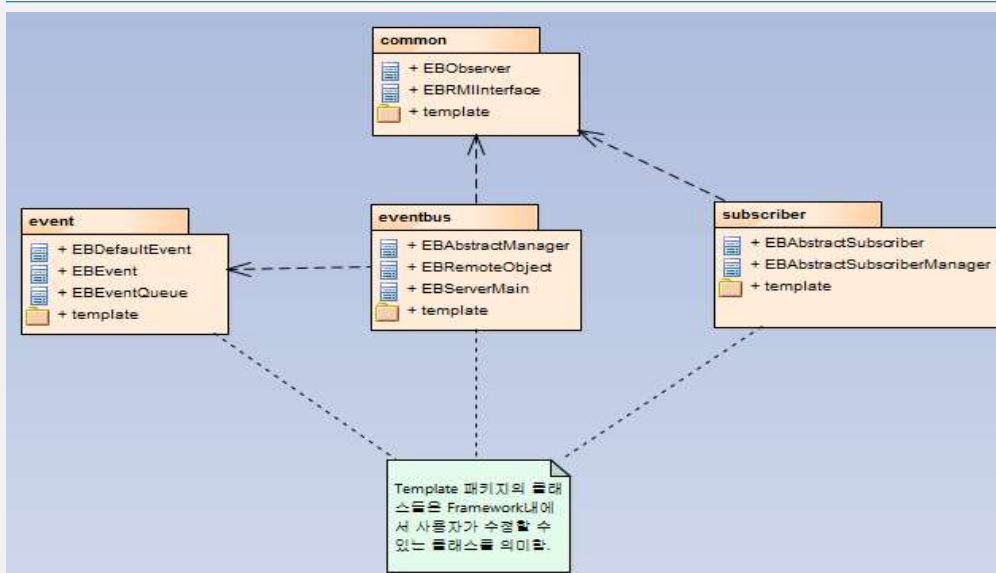
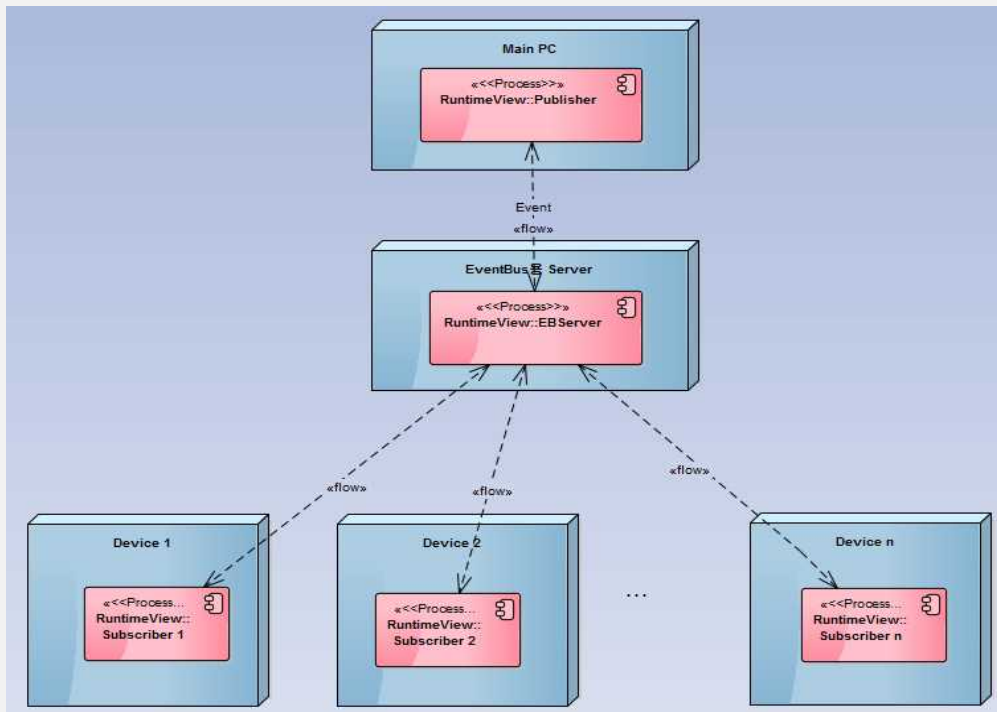


그림 4_Allocation View



2.2 Framework Code 가이드

Framework Code 가이드는 어플리케이션 코드의 모든 패키지와 클래스 그리고 API 등을 체계적으로 보여주는 Code View 이다.
 각 아키텍처 공통 Framework는 아래와 동일한 구조의 Code가이드 구조를 제공한다.
 Code 가이드의 경우, 공통 아키텍처 Framework의 이해와 어플리케이션의 코드의 원활한 개발을 돕는데 활용된다.

그림 5_빌딩보안 시스템의 Framework Code 가이드 Overview

All Classes

Packages

common
event
eventbus
eventbus.template
subscriber
subscriber.template

All Classes

EBAbstractManager
EBAbstractSubscriber
EBAbstractSubscriberManager
EBConstants
EBDefaultEvent
EBEvent
EBEventQueue
EBManagerTemplate
EBObserver
EBRemoteObject
EBRMIInterface
EBServerMain
EBSubscriberMain
EBSubscriberManagerTemplate

OVERVIEW

PACKAGE

CLASS

USE

TREE

DEPRECATED

INDEX

HELP

PREV

NEXT

FRAMES

NO FRAMES

Packages

Package	Description
common	
event	
eventbus	
eventbus.template	
subscriber	
subscriber.template	

OVERVIEW

PACKAGE

CLASS

USE

TREE

DEPRECATED

INDEX

HELP

PREV

NEXT

FRAMES

NO FRAMES

2.3 패턴적용 가이드

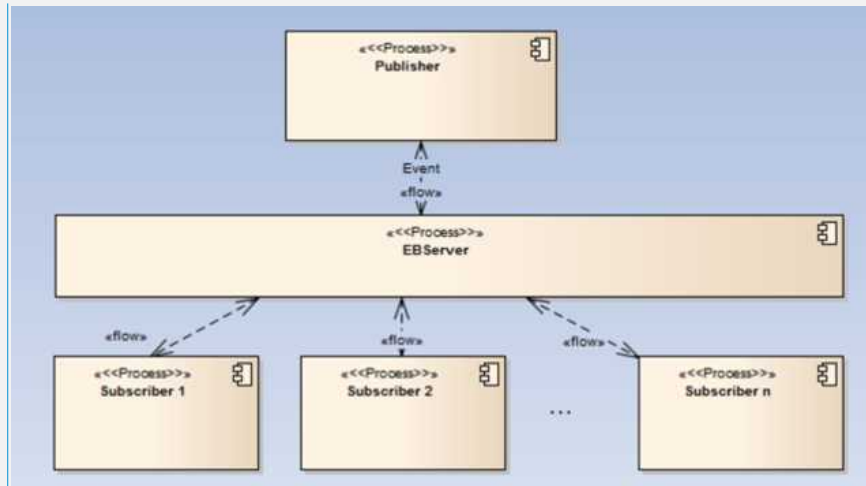
패턴적용가이드는 아래 표에서 보듯이 3개의 어플리케이션/패턴별로 구성되어 있으며, 시스템에 적용한 사례를 구체적으로 볼 수 있다.

표_4_어플리케이션 별 패턴 적용가이드 구성

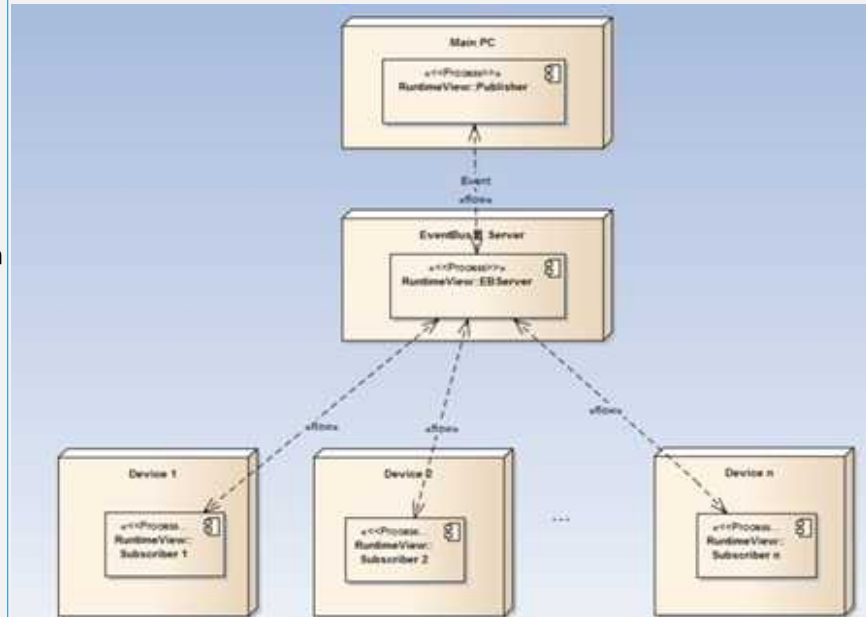
구 성	빌딩보안시스템 (임베디드)	항공기 센서 시스템 (임베디드)	웹기반 공용그래픽편집기 (Stand-alone)
패턴적용가이드	Event-Bus	Pipe & Filter	Composite

표_5_빌딩보안 시스템 패턴 적용가이드 예시(Event-Bus pattern)

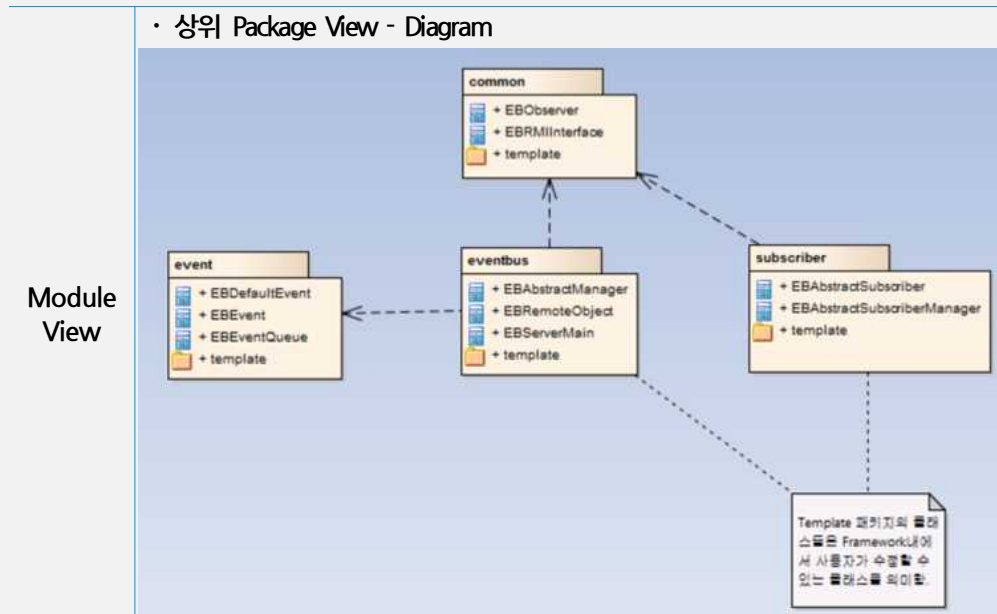
구 분	Event-Bus 공통 Framework
정 의	하나의 Publisher가 다수의 Subscriber에게 상태가 변경되었음을 단방향 전파로 통지하는 패턴협력 컴포넌트들의 상태를 동기화 하는데 유용함. Observer 패턴, Dependents 패턴, Event 패턴으로 사용됨
정 황 (Context)	한번의 호출로 다수의 협력 컴포넌트(Subscriber)의 상태를 변경해야 하는 경우
문 제 (Problem)	특정 컴포넌트에서 발생하는 상태 변경 정보를 하나 이상의 컴포넌트에서 수신 해야 함. Publisher와 Subscriber는 서로 tightly coupled 되어서는 안된다.
해 법 (Solution)	하나의 컴포넌트를 Publisher로 두고 상태 변경을 받을 컴포넌트 들을 subscriber로 둔다. Subscriber는 Publisher에서 제공하는 인터페이스를 통해서 등록한다. Publisher의 상태가 변경되면 등록된 Subscriber에게 변경상태를 전송한다.
장.단점	<ul style="list-style-type: none"> · 장 점 다수의 컴포넌트에게 동시에 이벤트 공지를 할 수 있음. GUI 인터페이스를 쉽게 만들 수 있어 GUI 프레임워크를 쉽게 만들 수 있음. · 단 점 Non-deterministic 한 문제가 발생할 수 있음. 이벤트 버스와 컴포넌트 로직이 Tightly coupled 될 수 있음.
Runtime View	Process View 분산환경을 지원하기 위해 Event-Bus(EB) Server와 Publisher를 포함한 모든 Subscriber들은 프로세스로 발생한다. Subscriber들은 EBServer를 통해 정해진 Event를 주고 받는다.

**Deployment View**

Runtime View에서 별도로 발생하는 프로세스들은 분산된 HW(Device 등)에서 동작할 수 있다. 분산 환경을 지원하기 위해 MM에서 제공하는 RM 통신 방식을 지원한다.

Allocation View**Module View**

- **Description common 패키지**
EB Framework에서 공통으로 사용하는 클래스들을 모아놓은 패키지이다.
- **event 패키지**
EBServer와 Subscriber들이 주고 받는 event를 정의하기 위한 패키지이다.
- **eventBus 패키지**
event를 전송하기 통로로서 subscriber들을 등록/삭제하는 기능을 제공하고 등록된 subscriber들에 해당 event를 전송하는 기능을 제공하는 패키지이다.
- **subscriber 패키지**
EB Server에 등록하는 Subscriber를 구현한 패키지이다.



III. 결론

지금까지 SW아키텍처 패턴 가이드에서 제공되는 내용들을 소개했다. SW개발에 필수로 활용되는 SW아키텍처는 중요성이 큰 반면 SW아키텍처와 관련된 콘텐츠 현저히 부족한 실정이다. SW아키텍처 패턴 가이드에서는 도메인 또는 프로젝트 측면에서 실질적으로 활용, 참조 가능한 정보들을 쉽게 사용할 수 있도록 제공하여, SW공학 역량 내재화와 SW품질 향상에 기여하고자 한다.

참고 자료

- [1] SW아키텍처 참조모델, 모바일 오픈 아키텍처 레퍼런스 설계, SW공학센터, 2014
- [2] SW아키텍처 참조모델, 빅데이터 플랫폼 설계 및 구현, SW공학센터, 2014
- [3] SW아키텍처 참조모델, Cloud Computing 기반의 Smart Working 시스템에 대한 소프트웨어 아키텍처 연구, SW공학센터, 2014
- [4] SW아키텍처 참조모델, 의료 정보시스템, SW공학센터, 2014
- [5] <https://www.swbank.kr/helper/architecture/architectureMain.do>