

전자정부 표준프레임워크

표준프레임워크 개요



Contents

1. _ 표준프레임워크 이해
2. _ 표준프레임워크 구성
3. _ 표준프레임워크 적용



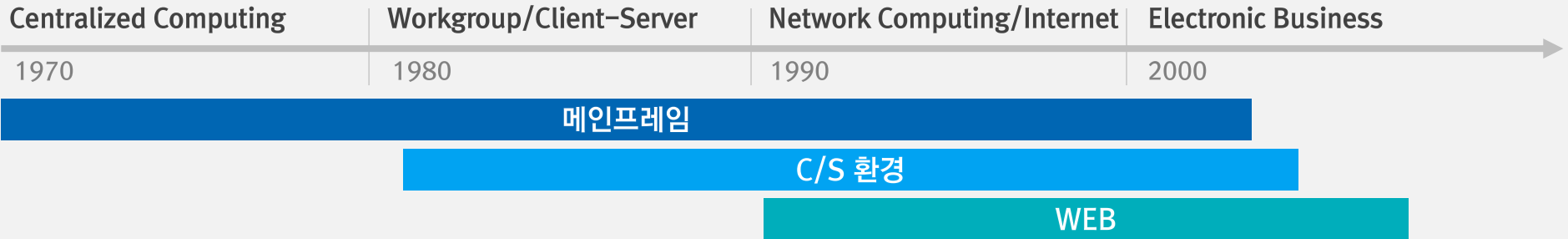
표준프레임워크 개요

들어가기

1. 어플리케이션 환경의 변화
2. 프레임워크 등장배경

1. 어플리케이션 환경의 변화

들어가기



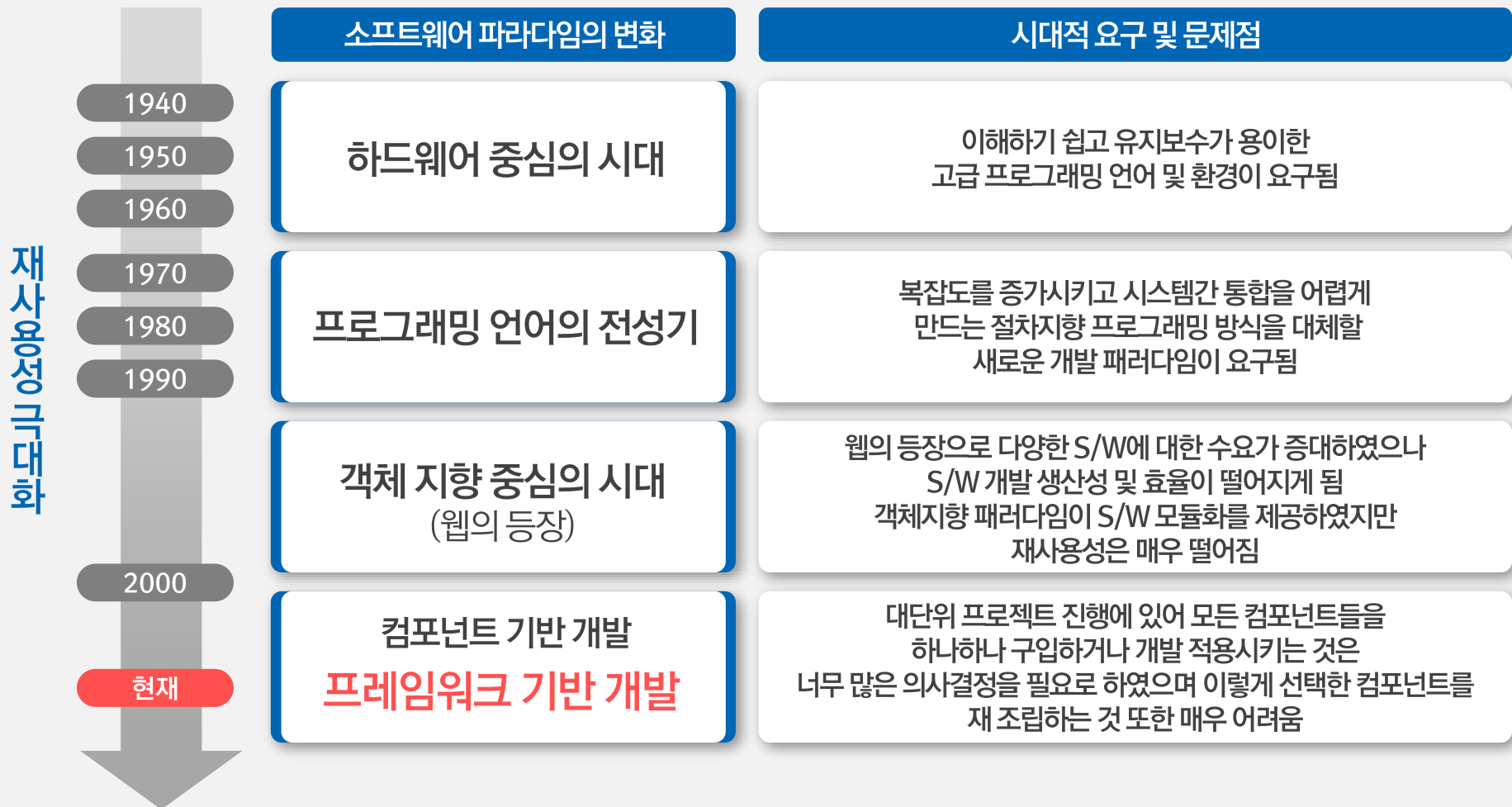
	메인프레임	C/S	WEB
구조	<div>메인프레임</div> <div>단말기</div> Data+Biz Logic+ Presentation Logic	<div>DBMS</div> <div>클라이언트</div> Data Biz Logic+ Presentation Logic	<div>DBMS</div> <div>WAS</div> <div>Web Server</div> <div>브라우저</div> Data Biz Logic Presentation Logic
장점	관리, 배포 편리	저렴한 서버 도입 및 구축비용 편리한 어플리케이션 개발환경 고수준 사용자 인터페이스	관리 배포 편리, 고수준 사용자 인터페이스 쉽고, 표준화된 어플리케이션 구축방식 타 시스템과의 연동, 통합, 확장 용이 DB 및 비즈니스 로직의 재활용성 향상 향후 유지보수 용이
단점	비싼 서버 가격 및 개발 비용 저 수준 사용자 인터페이스 타 시스템과의 연동, 통합, 확장이 어려움	어플리케이션의 배포, 관리의 어려움 타 시스템과의 연동, 통합이 어려움 향후 시스템의 확장이 어려움	

2. 프레임워크 등장배경

들어가기

소프트웨어 패러다임의 변화

소프트웨어 패러다임은 변화하는 시대적 요구에 맞추어 빠르게 변화하여 왔고 궁극적으로 그 변화의 방향은 재사용성 극대화에 맞추어져 왔음

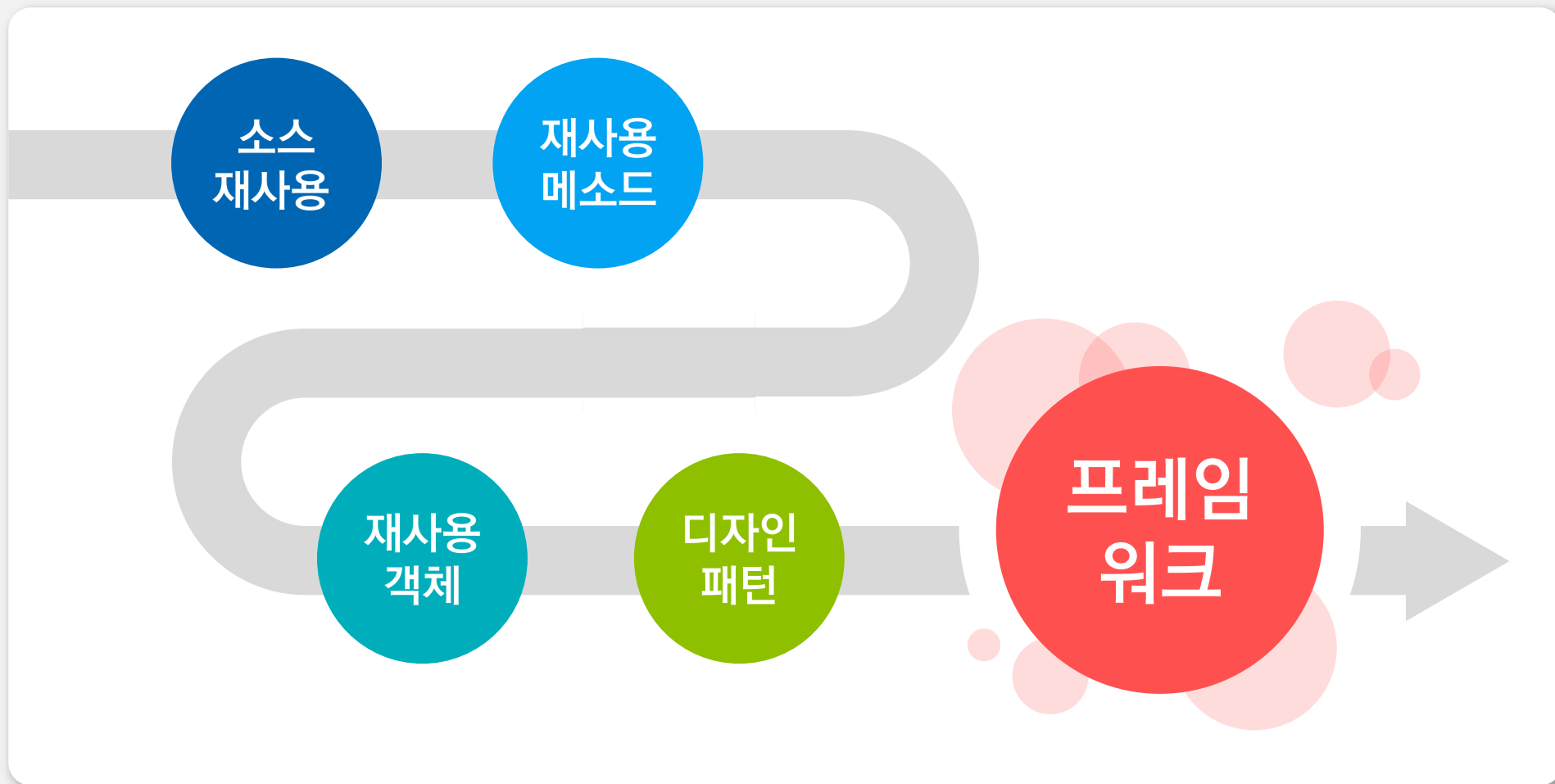


2. 프레임워크 등장배경

들어가기

재사용 방식의 발전 방향

재사용을 극대화 해주고 생산성, 확장성, 성능, 효과적인 유지보수를 지원해주기 위해 재사용 방식은 다음과 같이 발전하여 왔으며 궁극적으로 프레임워크가 등장하게 됨



2. 프레임워크 등장배경

들어가기

재사용 방식의 발전 방향

소스 재사용

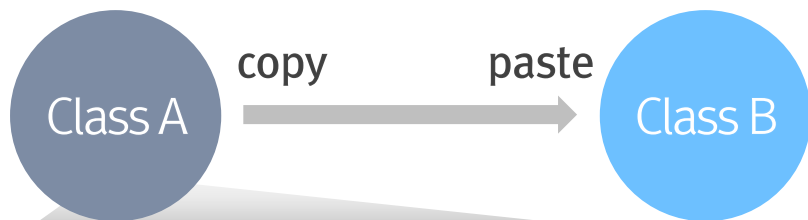
재사용 메소드

재사용 객체

디자인 패턴

프레임워크

초보적인 재사용 방식으로 과거에 유사한 문제를 코딩한 적이 있거나 아니면 비슷한 예제를 다른 소스에서 발견했을 때 해당부분을 복사해서 사용



```
GregorianCalendar date =  
(GregorianCalendar)Calendar.getInstance();  
SimpleDateFormat df = new SimpleDateFormat("yyyymmdd");  
String sdate = df.format(date);
```

한 예로, 클래스 A에서 'Date'를 'String'으로 변환하는 코딩을 해놓았다고 한다면 클래스 B에서 동일한 로직이 필요해서 클래스 A에서 해당부분을 복사해서 사용할 수 있습니다.

문제점

이후에 JDK 버전이 바뀌어 동일한 기능을 제공하는 향상된 인터페이스가 나왔다고 하면, 이런 SimpleDateFormat을 사용한 A, B 클래스를 모두 변경해야 함

2. 프레임워크 등장배경

들어가기

재사용 방식의 발전 방향

소스 재사용

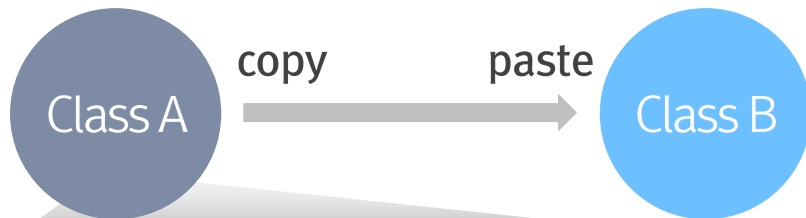
재사용 메소드

재사용 객체

디자인 패턴

프레임워크

복사/붙이기 방식과 동일한 코드가 여러 클래스에서 나오는 것을 지양하기 위한 방법으로 C언어에서 하던 것처럼 자주 이용하는 기능을 라이브러리로 만들어 재사용



유사한 기능을 모아 하나의 클래스에 정의해 놓으면 메소드 라이브러리로 구성되며 해당 라이브러리를 가져와서 재사용 가능
이후에 JDK내용이 바뀌거나 향상된 기능을 제공하고 싶을 경우에도 해당 메소드만 수정하여 제공하면 됨

문제점

메소드 재사용 방법은 동일한 작업 영역에서는 유용하게 사용될 수 있으나 작업 영역이 서로 다른 경우, 특정한 기능의 메소드를 사용하기 위하여 다른 작업 영역의 메소드를 사용한다면 유지보수 시 복잡해 질 수 있음

2. 프레임워크 등장배경

들어가기

재사용 방식의 발전 방향

소스 재사용

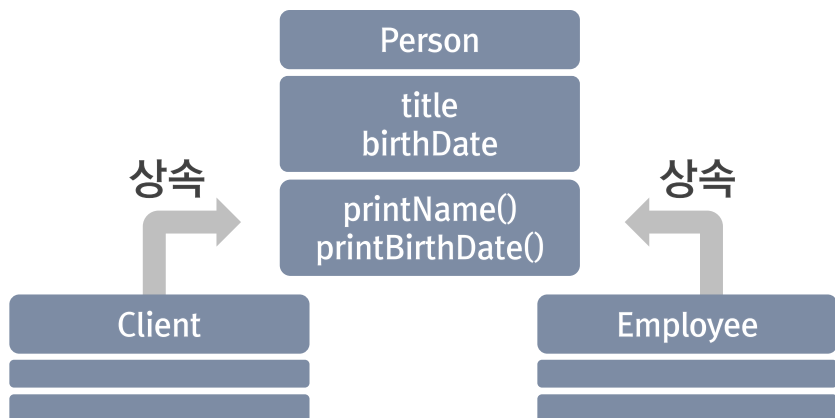
재사용 메소드

재사용 객체

디자인 패턴

프레임워크

앞의 재사용 방식들은 비단 자바뿐 아니라 다른 언어를 사용한 개발자들도 흔히 경험한 재사용 방법임
자바와 같은 객체 지향 언어에서 새로 소개되는 재사용 방식이 클래스를 통한 재사용임



Person 클래스의 printBirthDate()의 내용이 변경되면 이를 상속 받고 있는 Client, Employee에서 사용되는 printBirthDate()도 자동적으로 변경

```
Public String printBirthDate(string format){
    return DateUtility.toStringDate(birthDate, format);
}
```

문제점

구조적인 객체만을 놓고 분석할 때는 이와 같이 **수직적인 재사용** 방식이 유용하나 상황적인 문제를 놓고 보았을 때 다른 접근 방식이 요구됨

2. 프레임워크 등장배경

들어가기

재사용 방식의 발전 방향

소스 재사용

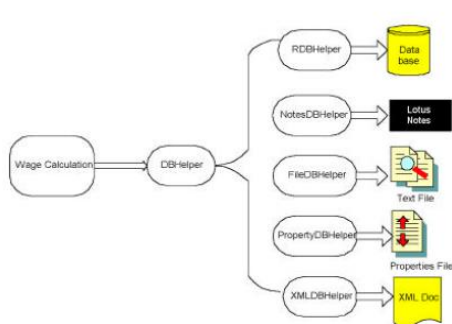
재사용 메소드

재사용 객체

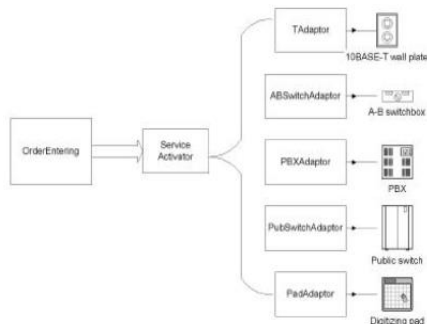
디자인 패턴

프레임워크

예를 들어 인사관리 시스템을 만들 때 구조적 대상을 저장하는 매개자가 관계형 DB로 고정되지 않고 때로는 텍스트파일, XML 등으로 저장되게 하려면 어떻게 할까? 클래스의 재사용 방식이 객체의 수직적인 재사용에 초점을 맞추었다면 디자인 패턴은 위와 같은 **상황적인 문제를 해결하여 주는 재사용 방식**임



Adapter패턴을 통한 객체 저장



운영 시스템 소프트웨어에
적용한 Adapter 패턴

디자인 패턴은 공통적인 로직 문제에 대한 일반화된 해결을 의미. 즉 클래스의 재사용이기보다 **해결책의 재사용**, 또는 **메커니즘의 재사용**이라 할 수 있음. 클래스 자체보다는 주어진 상황을 분석하고 그 상황에서 공통적인 조건을 추상화해서 이를 해결하는 일정한 패턴이 재사용될 뿐이며, 옆의 예에서 보는 바와 같이 'RDBHelper'와 'PadAdaptor'는 별개의 클래스이며 다양한 리소스에 접근하는 Adaptor의 역할을 수행 단지 동일한 패턴 하에서 유사한 역할을 하고 있을 뿐임

문제점

디자인 패턴은 전체 애플리케이션의 입장에서 본다면 시스템의 부분적인 문제를 해결하는 것에 불과함

2. 프레임워크 등장배경

들어가기

재사용 방식의 발전 방향

소스 재사용

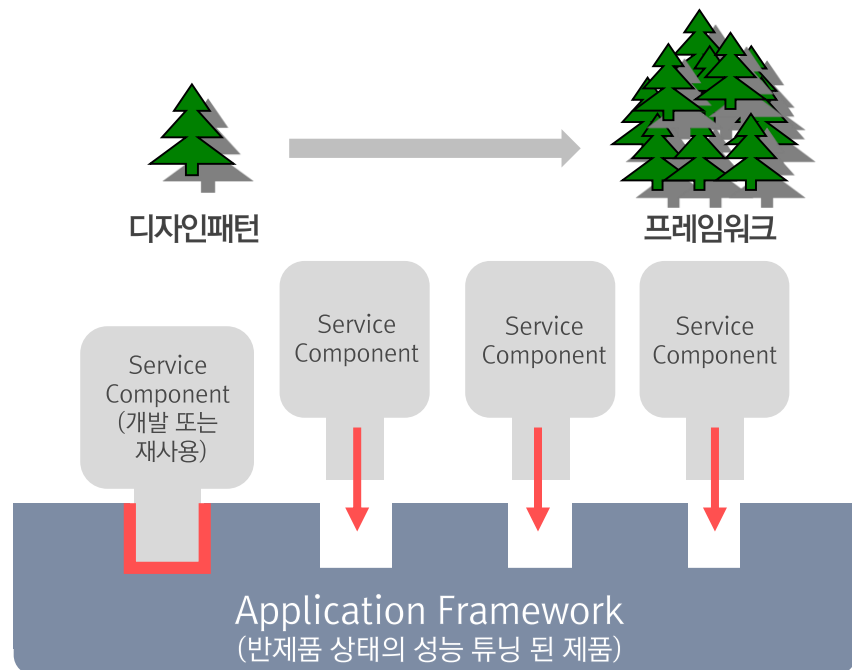
재사용 메소드

재사용 객체

디자인 패턴

프레임워크

하나의 주어진 상황 문제를 어떻게 해결하는지를 보는 것을 협의의 디자인 패턴 또는 매커니즘이라면, 전체 프로그램을 어떻게 구성할 것인가를 생각하는 디자인 패턴이 바로 프레임워크임



프레임워크란 디자인 패턴과 같은 부분적인 해결책을 전체적인 관점에서 통합하여 애플리케이션의 설계 및 구현틀을 제공하는 것으로, 건축설계사에게 기본 설계 템플릿이 있는 것 처럼 프레임워크가 마련되면 애플리케이션 개발 시 시스템의 재사용도가 높아지게 되고 나아가 개발 기간과 리소스를 절감할 수 있음

프레임워크란 반제품 상태의 성능 튜닝된 제품을 토대로 도메인 별로 필요한 서비스 컴포넌트를 만들어 삽입하는 형태로 시스템 구축을 도와줌. 이를 통해 기존 컴포넌트의 재사용성이 확대되고 성능 또한 보장 받을 수 있음

1. 표준프레임워크 이해

1. SW프레임워크 개념
2. 표준프레임워크 개념
3. 표준프레임워크 배경 및 목적
4. 표준프레임워크 추진경과 및 성과

1. SW 프레임워크 개념

1. 표준프레임워크 이해

프레임워크는 클래스 및 패턴의 집합으로 소프트웨어 개발의 효율성과 품질을 높이기 위한 반제품 성격의 소프트웨어이다.

SW 프레임워크 정의

일련의 문제 해결을 위한 추상화된 디자인을 구현한 클래스들의 집합
클래스 보다는 큰 규모의 재사용을 지원함

구체적이며 확장 가능한 기반 코드, 설계자가 의도하는 아키텍처와 디자인 패턴의 집합

실전에서 얻은 최적화 개발 경험을 반영한 재사용 가능한 API 집합

반제품 성격의 소프트웨어

라이브러리와 달리 애플리케이션의 틀과 구조를 결정, 그 위에 개발된 개발자의 코드를 제어

SW 프레임워크 정의

어플리케이션 개발을 도와주는 재사용 가능한 디자인 패턴 및 소스코드의 집합임

각 구축 사업에서 구현되어야 하는
업무 기능들로 개발프레임워크에서
제공하는 디자인 및 소스코드를
이용하여 개발됨

어플리케이션의 동작을 위해
어플리케이션의 하부에서
동작하는 SW
(예: 오라클 DB, Unix 등)



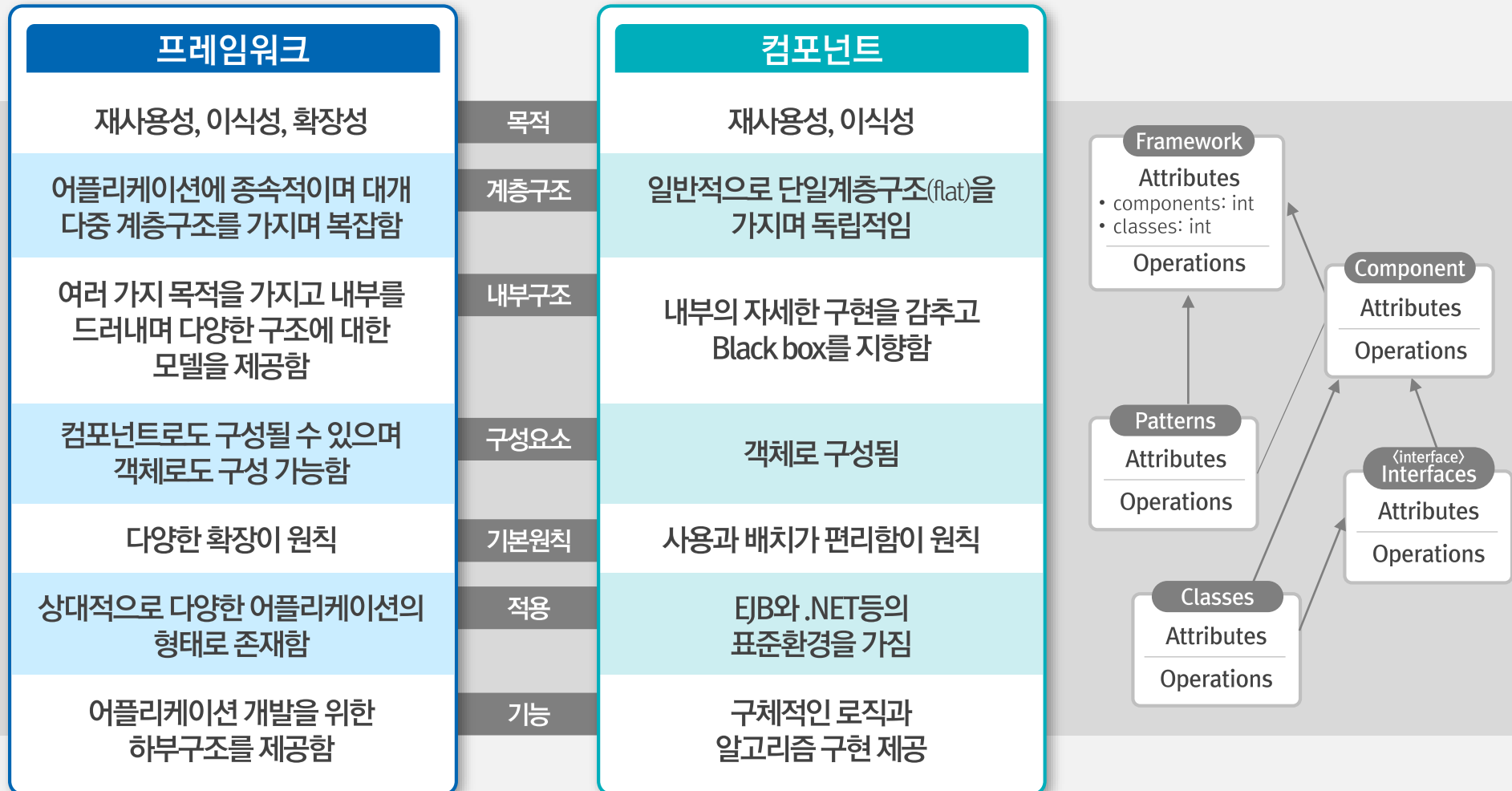
특정 업무 요구사항을 만족하기
위한 자동화 시스템
(예: 인사관리 시스템 등)

특정기술 또는 어플리케이션을
구성하는 기반구조로
SI 프로젝트에서 **반복적**으로
사용되는 **디자인 및 기반 소스
코드**를 정의한 **재활용** 자산임

1. SW 프레임워크 개념

1. 표준프레임워크 이해

SW 프레임워크는 다중계층적인 구조를 가지고 여러 클래스 및 컴포넌트들의 특정한 패턴형태로 구성되어 있음



1 SW 프레임워크 개념

1. 표준프레임워크 이해

SW 프레임워크를 활용하면 개발 및 운영 용이성을 제공하고, 시스템 복잡도 감소, 재사용성 확대 등의 장점이 있다.

개발용이성

패턴 기반 개발과 비즈니스 로직에만 집중한 개발 가능
공통 기능은 프레임워크가 제공

시스템 복잡도의 감소

복잡한 기술은 프레임워크에 의해 숨겨짐
미리 잘 정의된 기술 셋 적용

이식성

플랫폼 비의존적인 개발 가능
플랫폼과의 연동은 프레임워크가 제공

품질보증

검증된 개발 기술과 패턴에 따른 개발
고급 개발자와 초급 개발자의 차이를 줄여줌

운영 용이성

변경이 용이함
비즈니스 로직/아키텍처 파악이 용이

개발코드의 최소화

반복 개발 제거
공통 컴포넌트와 서비스 활용

변경용이성

잘 구조화 된 아키텍처 적용
플랫폼에 비 의존적

설계와 코드의 재사용성

프레임워크의 서비스 및 패턴의 재사용
사전에 개발된 컴포넌트의 재사용

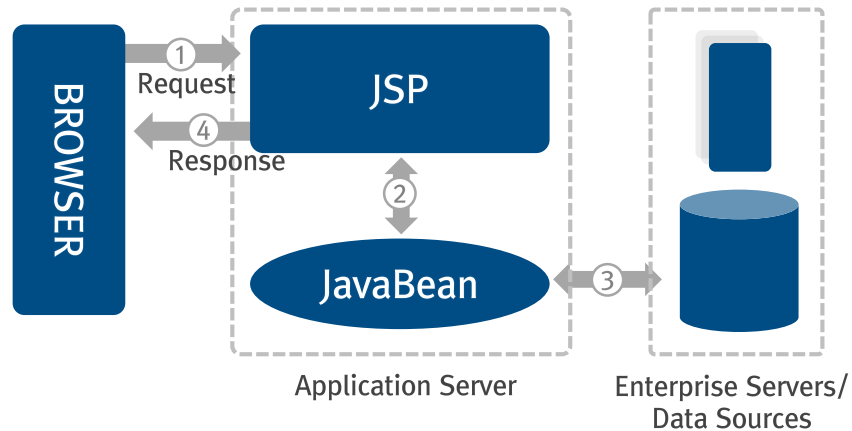
프레임 워크

1. SW 프레임워크 개념

1. 표준프레임워크 이해

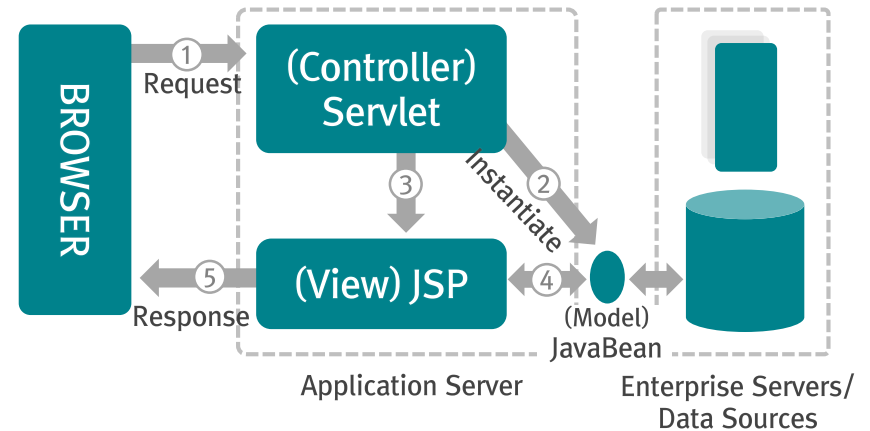
Model2는 소스코드가 화면, 로직, 데이터 등 역할별로 구조화 되어 구성되어 유지보수와 분업이 편리하여 대규모 시스템에 적용됨 (SW프레임워크의 시초)

Model 1



- 개발 속도가 빠르다
- 배우기 쉽고 빠르게 적용 할 수 있다
- JSP 페이지에 프리젠테이션 로직과 비즈니스 로직이 혼재되어 복잡하고 분업 작업 어려움
- 코드를 알아보기 힘들

Model 2 (MVC)

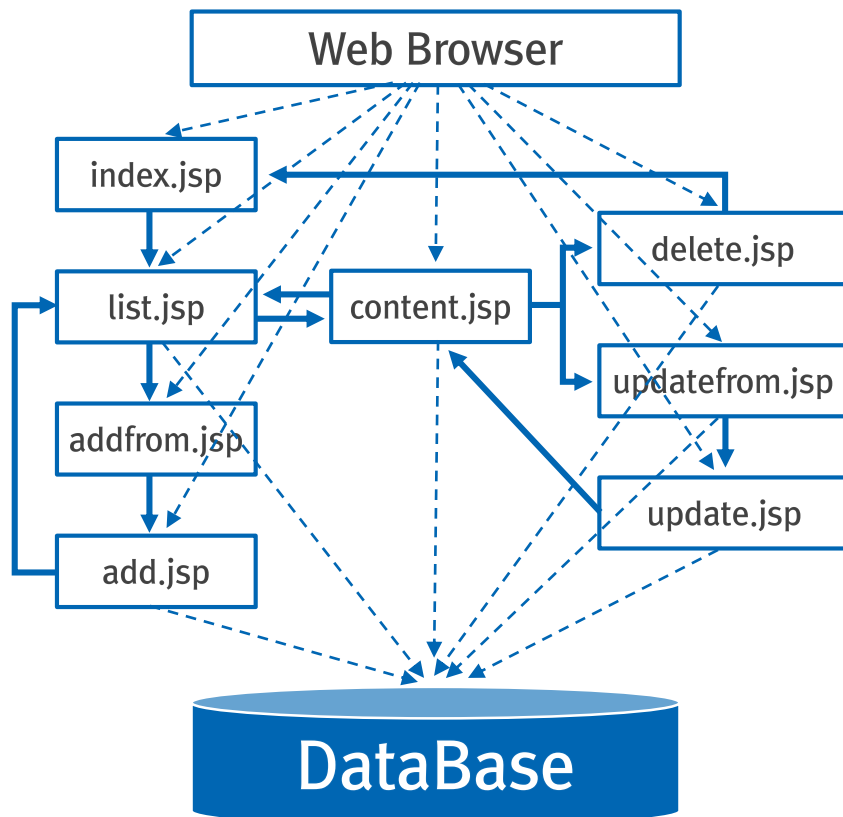


- 소스코드가 구조화 되어 대규모 시스템에도 잘 적용
- 유지보수가 편리하고 분업이 쉬움

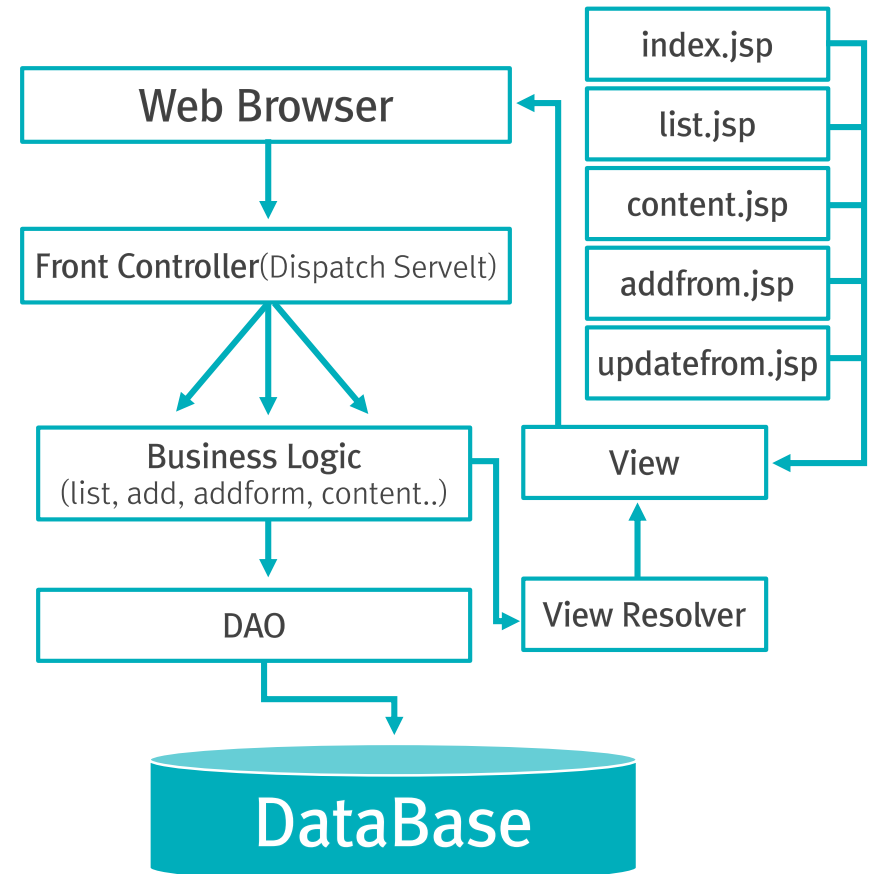
1. SW 프레임워크 개념

1. 표준프레임워크 이해

Model 1

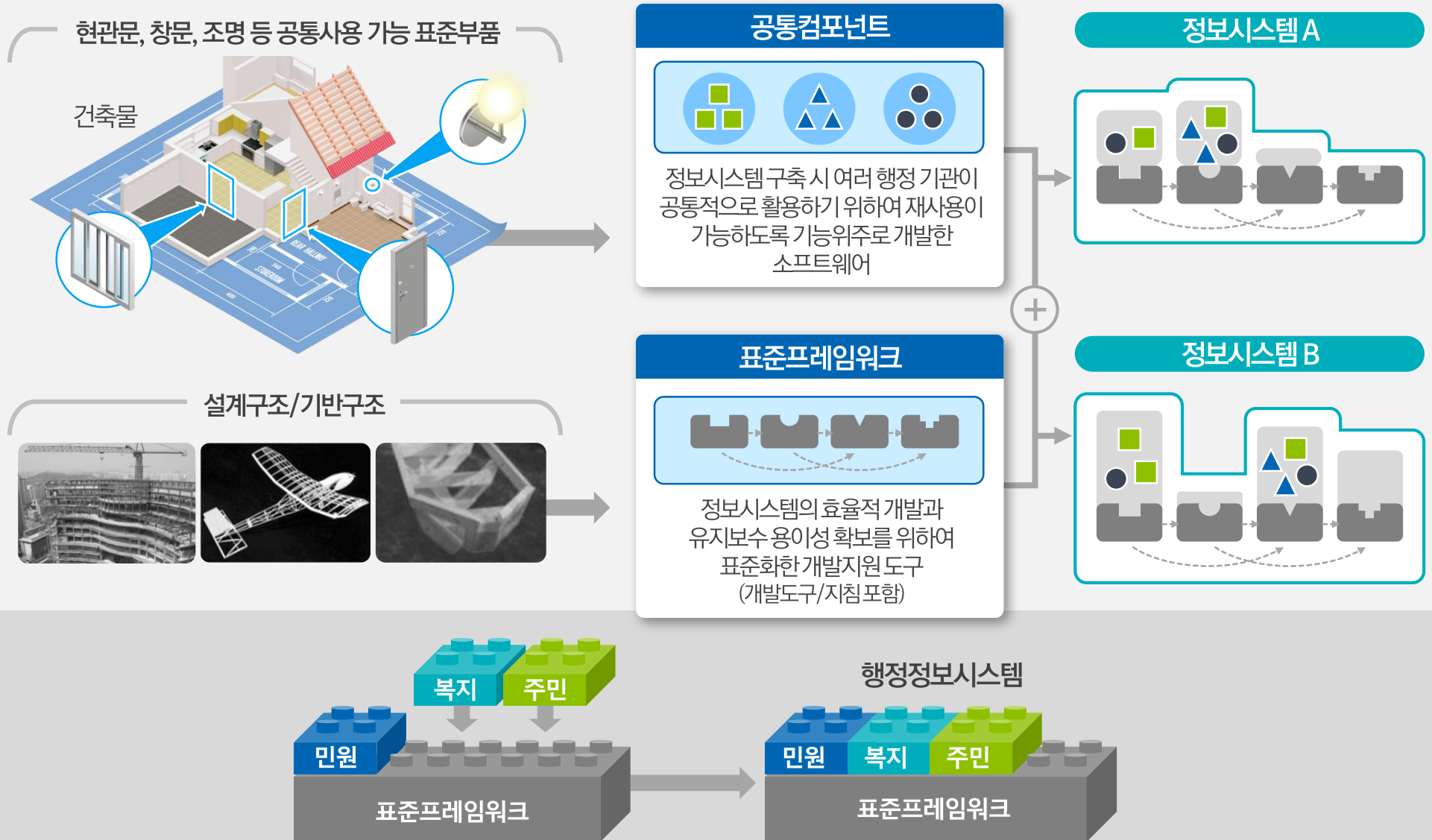


Model 2 (MVC)



2. 전자정부 표준프레임워크 개념

1. 표준프레임워크 이해

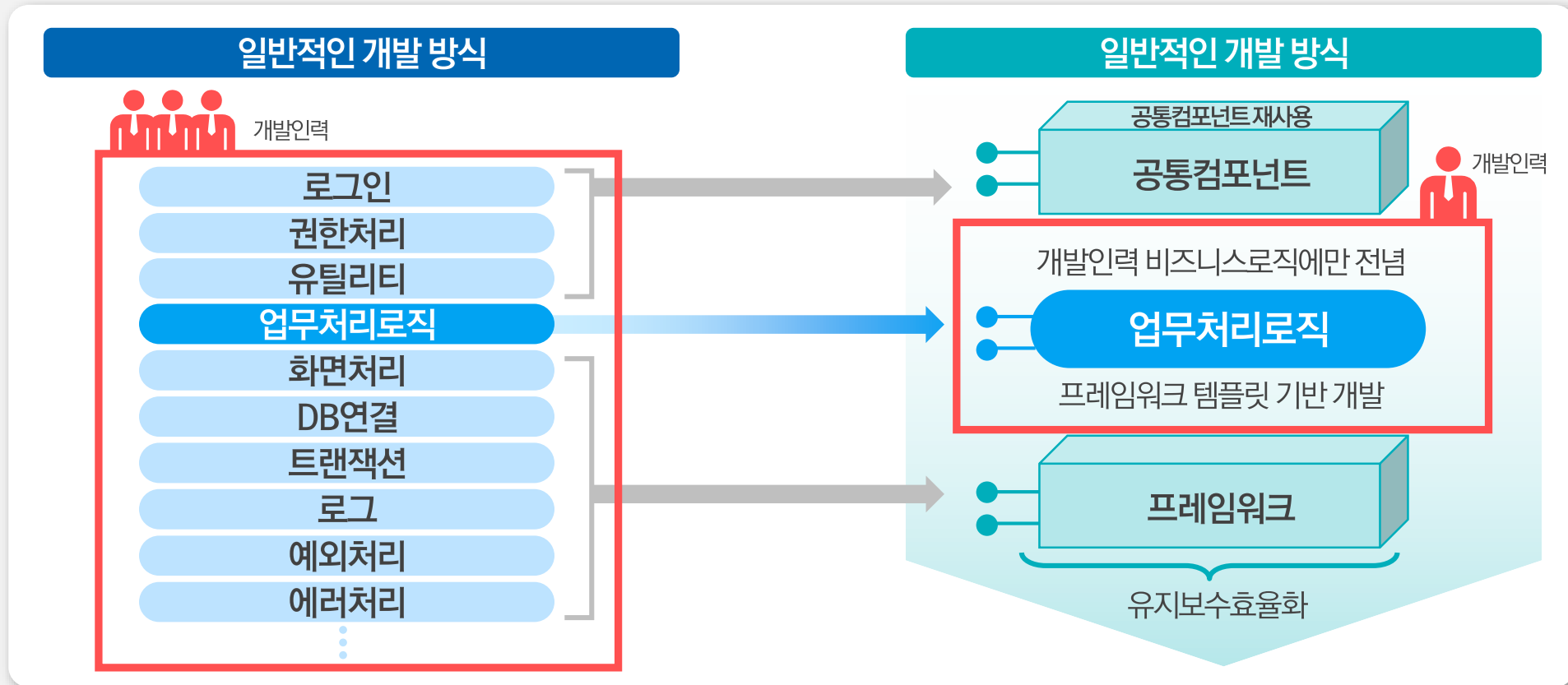


2. 전자정부 표준프레임워크 개념

1. 표준프레임워크 이해

개발방식

프레임워크 기반 템플릿 프로그램을 통해 개발자는 오직 비즈니스 로직 개발에만 전념하여 개발 생산성이 향상되고, 템플릿 기반의 개발 표준화를 통한 품질 보장 및 위험요소 극소화



업무의 단순화

성능 및 품질 보장

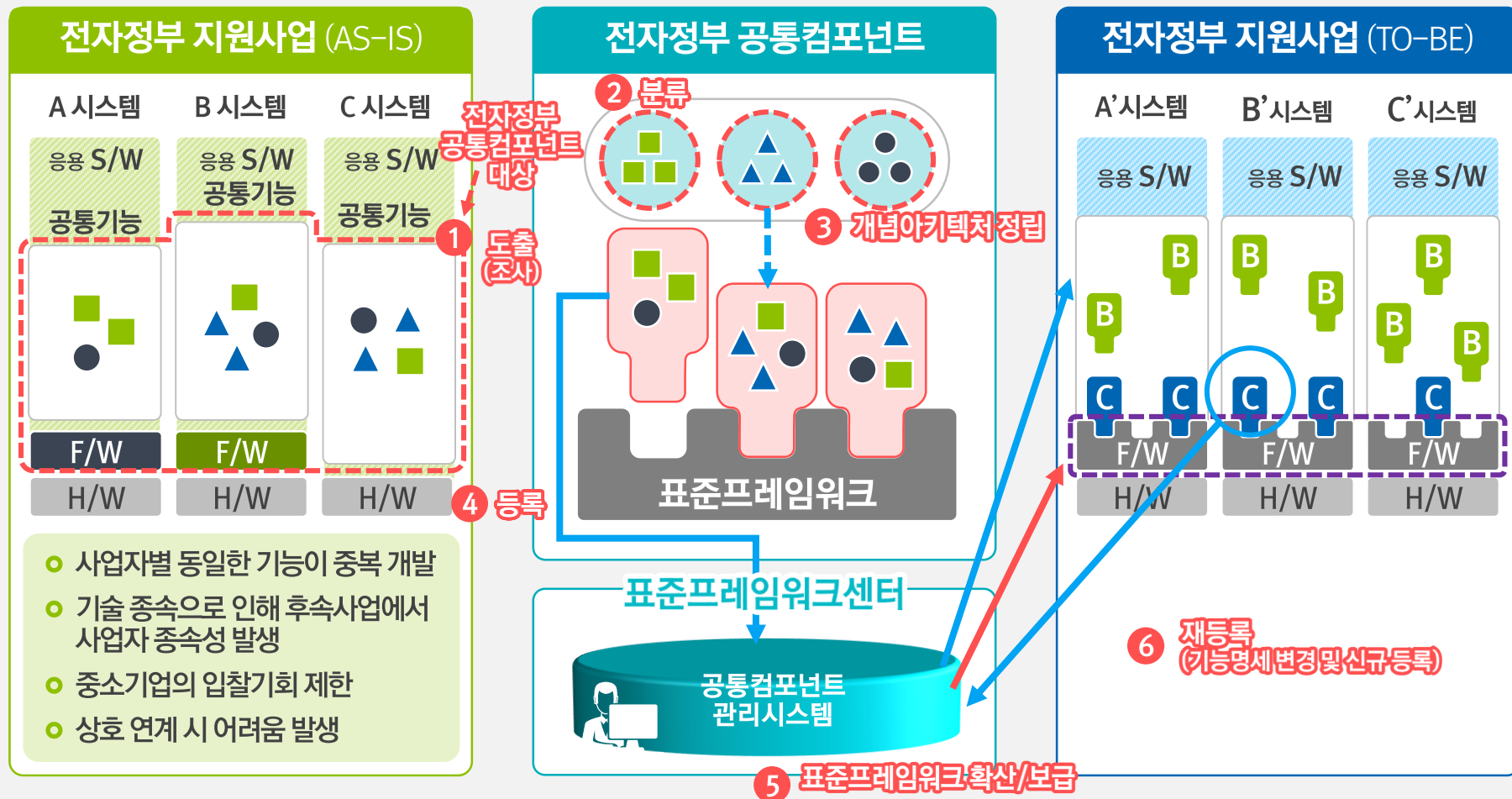
비용 절감

납기 보장

2. 전자정부 표준프레임워크 개념

1. 표준프레임워크 이해

표준화



동일한 표준프레임워크 기반 개발

공통컴포넌트의 재사용으로
생산성 및 서비스 품질 향상

중복개발 방지로 예산 절감 및
중소기업의 사업참여 기회 확대

3. 전자정부 표준프레임워크 배경 및 목적

1. 표준프레임워크 이해

전자정부 서비스 품질향상 및
정보화 투자 효율성 향상

국가 정보화
투자 효율성 제고

중소 SI업체
경쟁력 확보

선진 국가정보화
추진 기반환경 제공

공통컴포넌트 재사용성 제고

공통컴포넌트 및
표준프레임워크 개발

표준프레임워크 활성화

차세대 전자정부 공통컴포넌트 및 표준프레임워크 구축 및 적용 요구

각 사업별로
공통컴포넌트
중복 개발

기관별/사업별
개별적인 정보화
사업추진

최적화/표준화된
공통개발기반 요구

특정업체 종속성
발생으로 인한
공정경쟁 저하 및
사업자 변경시 예산낭비

개발표준 부재로
시스템간 상호운용성
및 재사용성 저하

4. 표준프레임워크 추진경과 및 성과

1. 표준프레임워크 이해

ISP 수립

- 표준프레임워크 목표모델 정립 및 공통컴포넌트 219종 도출
※ 전문가 협의회(총7회), 공청회('08.4), 차관주재간담회('08.10) 개최

'07.12~'08.6

1~3단계 구축 사업

- 표준프레임워크 실행환경, 개발환경, 관리환경, 운영환경 구축
- 로그인, 게시판 등 공통컴포넌트 219종 구축
- 표준프레임워크 포털시스템 및 영문포털
- 국가대표포털, 통합인증체계, 여수엑스포, 국토부공간정보 등 92개 사업 적용
- 표준프레임워크 교육 1643명 수료
- 정규교육 31회 859명, 대기업자체 22회 440명, 감리원교육 1회 31명, 현장교육 21회 313명 등
- '09년 대한민국 SW 기술대상 우수상 수상(지경부)
- 싱가포르 주관 미래정부시상식(FutureGov Awards) 3개 부문 수상
- 올해의정부상(대상), 기술선도상, 정부혁신상
- 2010 공개SW 활용상 지식경제부장관상(대상) 수상(지경부)
- 국제공인인증 CMMI 획득

'08.11~'10.11

'11.4~'11.11

- 표준프레임워크 오픈소스 버전 업그레이드
- 표준프레임워크 경량화 및 모바일 표준프레임워크 구축
- 모바일 웹 공통컴포넌트 40종 구축

4단계 및 모바일

'12.4~'12.11

- 표준프레임워크 배치프레임워크 개발
- 모바일 표준프레임워크 디바이스 API 개발

5단계 구축 사업

'11.1~현재

- 공공정보화사업 적용지원, 중소기업 개발자 교육

운영관리 · 기술지원

2. 표준프레임워크 구성

1. 표준프레임워크 구성
2. 표준프레임워크 실행환경
3. 표준프레임워크 개발환경
4. 표준프레임워크 관리환경
5. 표준프레임워크 운영환경
6. 공통컴포넌트
7. 모바일 표준프레임워크

1. 표준프레임워크 구성

2. 표준프레임워크 구성

구성원칙

국가정보화 개발 표준 수립을 위해 표준화 원칙을 정의하고 표준프레임워크 구성

표준화 요구 사항

- 1 사용이 편리하고 기능이 풍부한 환경 제공
- 2 다양한 기술, 업무요건 수용 가능한 유연한 구조
- 3 신속한 기술지원 서비스 체계
- 4 지속적인 유지보수와 적시적인 버전 업 체계
- 5 **개방표준(Open Standard)의 준수**
- 6 관련기관 및 업계의 의견을 충분 수렴
- 7 **특정 사업자에 대한 종속성 배제**
- 8 전자정부 표준프레임워크 인력양성
- 9 **기술공개를 통한 중소기업 참여 활성화**
- 10 오픈소스 프레임워크 기술 수용
- 11 **업계의 최신 개발프레임워크 제품 기술 수용**

표준화 원칙

공공 기관 정보화담당자, 대학교수,
SW업계로 구성된 자문 **협의회의 구성과 운영**

상용 솔루션 관련 영역은 배제하거나
필수적인 기능만 제공

상용 솔루션 (UI Adaptor)과 연동이 가능하도록
표준을 제공하고 연동을 보장

개방형 표준을 활용하고
오픈 소스를 적극 활용하여 기술 종속성 제거

전자정부 사업에서 활용 되는
주요 인프라 솔루션 (WAS, DB 등)과 **호환성을 보장**

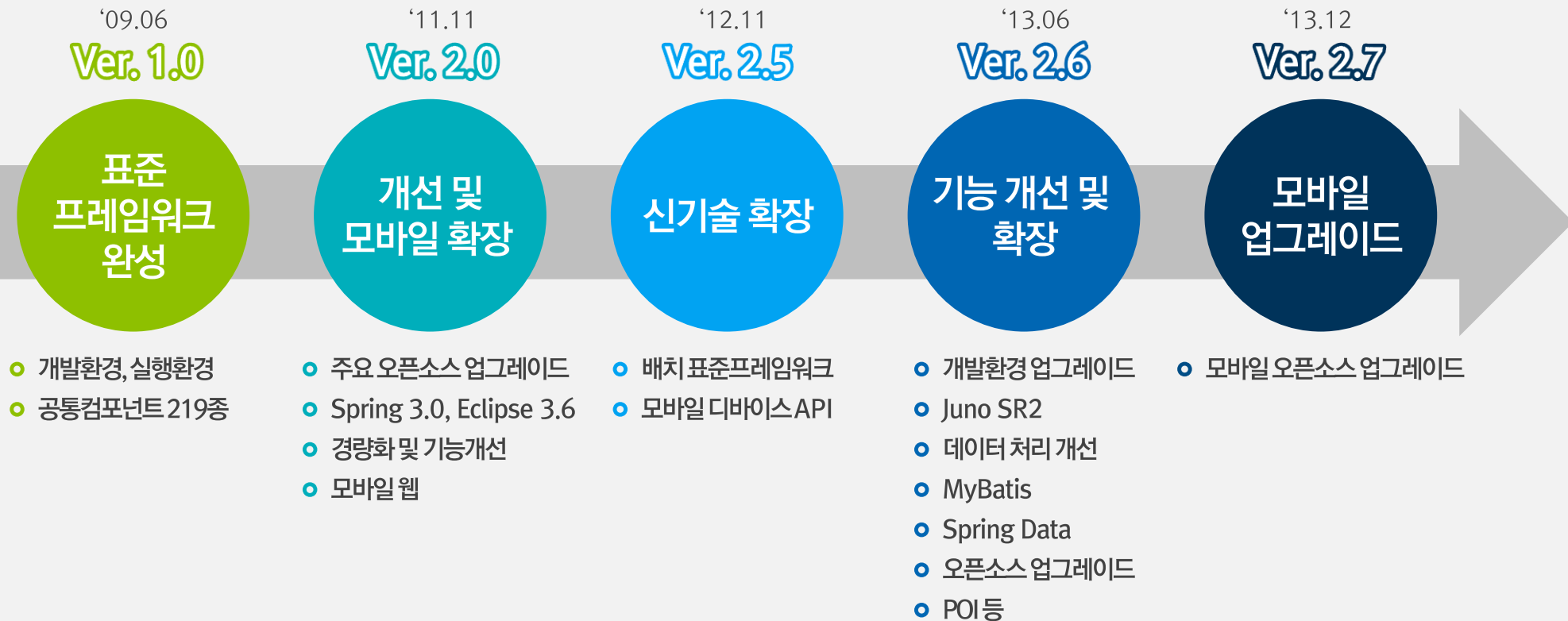
분석 설계 **산출물 및 소스코드의 공개**를 통해
사용자의 활용성을 제고

1. 표준프레임워크 구성

2. 표준프레임워크 구성

표준프레임워크 진화·발전(1/2)

2009년 6월 표준프레임워크 1.0은 오픈 한 이후, 빠른 속도로 보급·확산됨에 따라 많은 환경변화가 발생하였으며, 이에 대한 분석을 통해 표준프레임워크 차기 버전들의 개선 항목을 도출

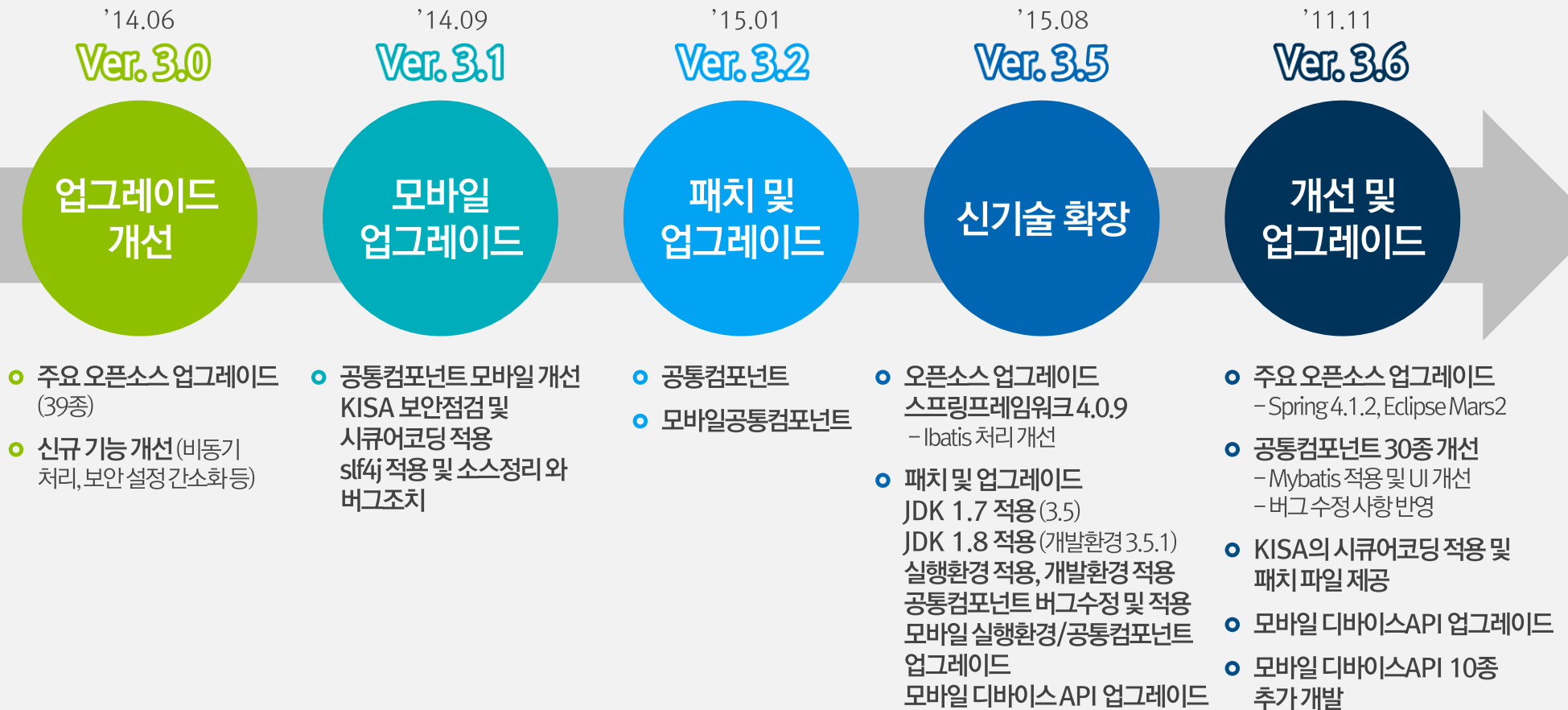


1. 표준프레임워크 구성

2. 표준프레임워크 구성

표준프레임워크 진화·발전(2/2)

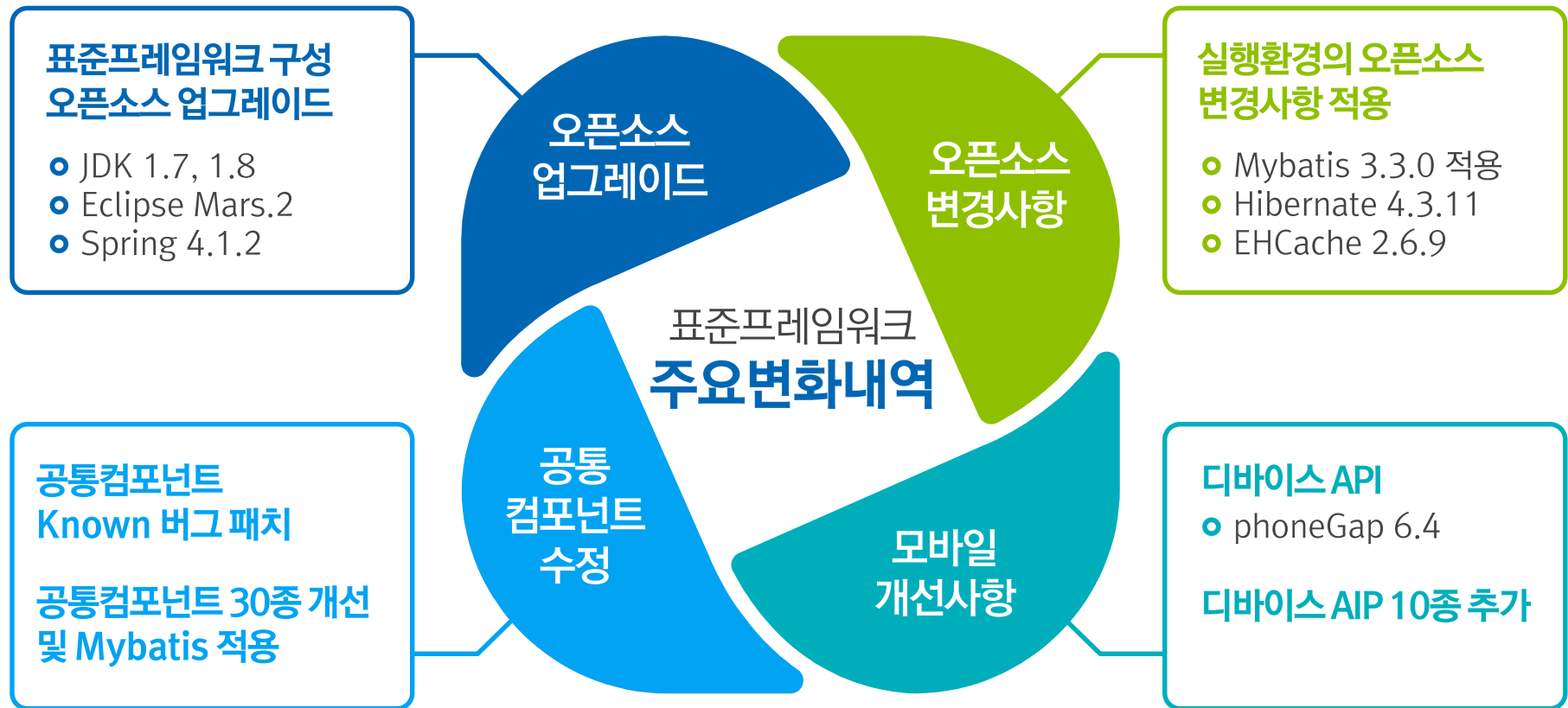
2009년 6월 표준프레임워크 1.0은 오픈 한 이후, 빠른 속도로 보급·확산됨에 따라 많은 환경변화가 발생하였으며, 이에 대한 분석을 통해 표준프레임워크 차기 버전들의 개선 항목을 도출



1. 표준프레임워크 구성

2. 표준프레임워크 구성

표준프레임워크 3.6 개선사항



1. 표준프레임워크 구성

2. 표준프레임워크 구성

구성도

표준프레임워크는 실행, 개발, 관리, 운영 등 4개의 환경과 모바일 표준프레임워크, 공통컴포넌트로 구성

공통컴포넌트

모바일

모바일전용 공통컴포넌트

기존 공통컴포넌트 모바일 전환

웹

공통기술컴포넌트

요소기술컴포넌트 (유틸리티)

표준프레임워크

개발환경

구현도구

배포도구

테스트도구

형상관리도구

모바일 디바이스
API 개발도구

실행환경

화면처리

업무처리

데이터처리

연계통합

공통기반

배치처리

모바일
화면처리

모바일 디바이스
API 실행환경

관리환경

서비스요청관리

변경관리

현황관리

표준관리

운영환경

모니터링도구

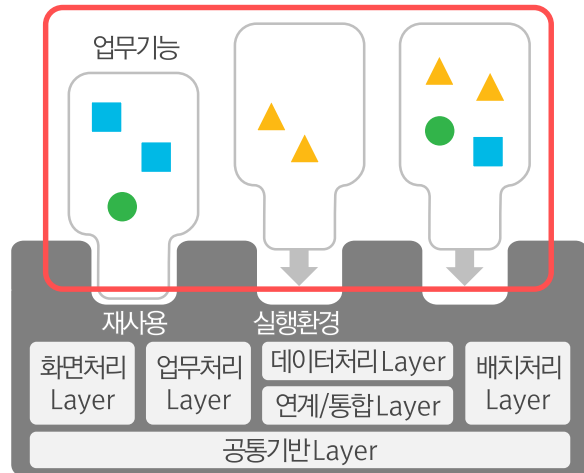
운영관리도구

배치운영도구

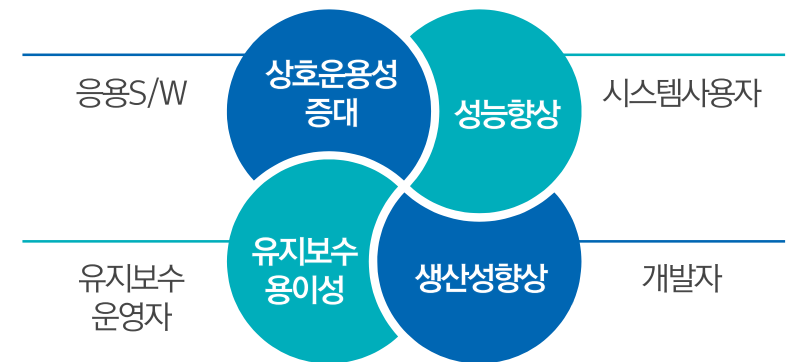
2. 표준프레임워크 실행환경

2. 표준프레임워크 구성

화면처리, 업무처리, 데이터처리, 연계처리, 공통기반, 배치처리 등 6개 레이어의 표준프레임워크 실행환경 구현



유연하고 통합된 어플리케이션 실행 환경 제공

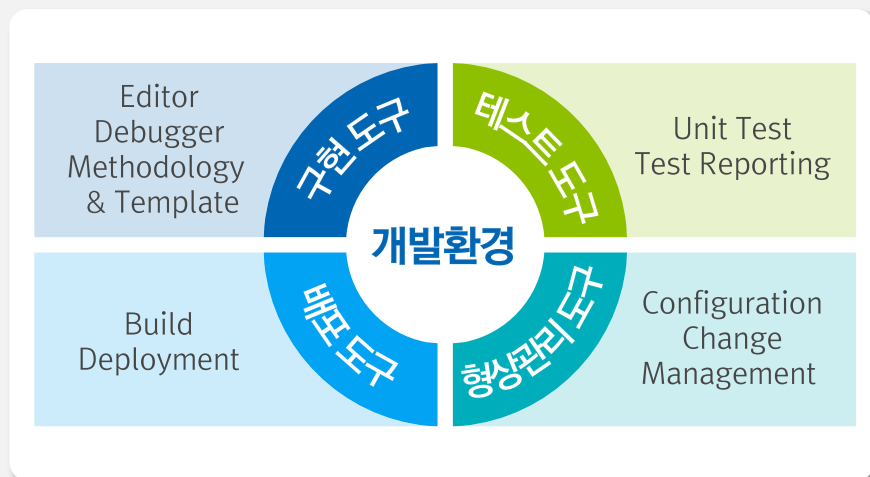


구현기능		구현내용
화면처리Layer	Ajax Support 등	UI 컴포넌트에 대한 인터페이스 및 화면구현에 필요한아키텍처제공 (UI Adaptor 제공 등)
업무처리Layer	Spring 등	비즈니스로직을 서비스로 구성하여 처리하는 기능을 제공 (MVC 패턴 제공 등)
데이터처리Layer	MyBatis 등	데이터베이스와 관련된 각종 접속 및 SQL 처리 기능을 제공 (DB 연결, SQL 처리 등)
연계/통합Layer	CXF 등	웹서비스, 연계 메타정보 등의 기능을 제공 (Web Service 제공 등)
공통기반Layer	log4j 등	서버기능의 다양한 재사용 컴포넌트 및 개발에 필요한 유틸리티 제공 (Bean 관리, 공통활용 기능 제공 등)
배치처리Layer	Batch Core 등	대용량 일괄 처리를 위한 설정 및 실행기능을 제공

3. 표준프레임워크 개발환경

2. 표준프레임워크 구성

구현도구, 테스트도구, 형상관리도구, 배포도구 등 쉽고 편리한 프레임워크 개발환경 구현

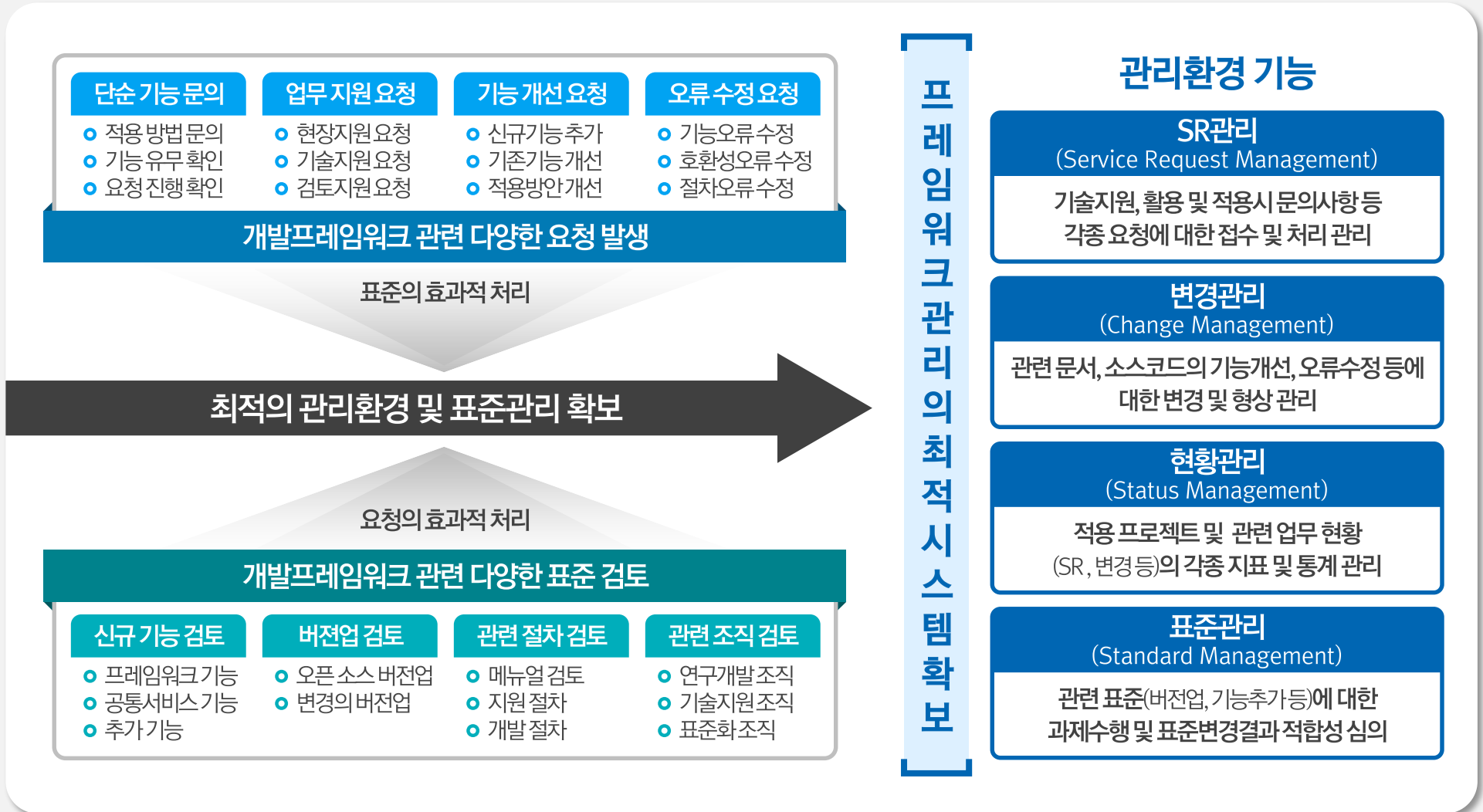


구현기능		구현내용
구현도구 Implementation Tool	UML Editor/ ERD Editor	UML, ERD Notation을 작성할수 있는 도구
	통합플러그인	실행환경 기반 개발을 위해 필요한 각 도구들 직관적으로 사용할수 있는 화면
	DBIO Editor	Sql을 작성하고 수정할수 있으며 테스트 수행을 통하여 sql에 대한 결과값을 확인하는 도구
테스트도구 Test Tool	Test Case	실행 가능한 테스트 코드를 작성할수 있는 도구
	Test Coverage	테스트 수행 커버리지를 분석하고 리포팅하는 도구
	Test Reporting	테스트 결과를 다양한 포맷으로 리포팅하는 도구
배포도구 Deployment Tool	개발자Build 도구	라이브러리 종속성 관리 및 개발자 PC에서 빌드할수 있는 도구
	배포 관리	이관 대상 및 주기를 설정할수 있는 도구
형상관리도구 Configuration Management	Configuration Management	형상 요소의 식별 및 등록, History를 지원하는 형상관리도구
	Change Management	이슈를 관리할수 있는 이슈 트래킹 시스템

4. 표준프레임워크 관리환경

2. 표준프레임워크 구성

관리환경은 표준프레임워크에 대한 다양한 문의 및 서비스 요청에 대한 접수 및 내부 프로세스 처리



5. 표준프레임워크 운영환경

2. 표준프레임워크 구성

전자정부 표준 프레임워크 기반 위에 실행되는 어플리케이션에서 발생하는 동작 정보와 수행로그를 에이전트를 활용하여 수집하고 이를 기반으로 운영자는 시스템 상태를 모니터링

구현 및 운영시에 편리한 환경 제공



- 서비스수행시간
- 자원현황
- WAS상태
- 에이전트상태



모니터링 도구

에이전트 관리

에이전트는 스케줄, 로깅 등의 설정을 기반으로 모니터링 대상 시스템에서 실행

모니터링 정보 수집

에이전트가 실행되면서 시스템 정보 및 프로그램 로그 수집 기록

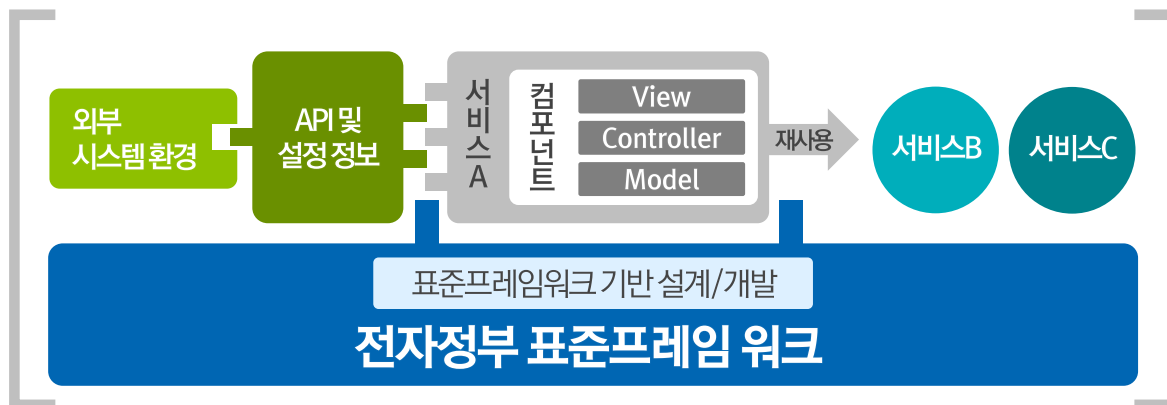
운영자 GUI

운영자에게 수집된 정보를 그래프, 차트를 활용하여 다양한 형태로 표현

6. 공통컴포넌트

2. 표준프레임워크 구성

공통컴포넌트는 표준프레임워크 기반의 표준 준수 및 유연성을 확보하여 재사용성을 극대화하고 또한, 기존 웹 뿐만 아니라 모바일 공통컴포넌트를 추가하여 모바일 웹 구현 시 활용 가능



구분		상세기능
공통 기술 서비스 (129종)	보안	실명확인, 권한관리, 암호화/복호화 등 8종
	사용자디렉토리/통합인증	일반로그인, 인증서로그인, 로그인정책관리 등 3종
	사용자지원	사용자관리, 상담관리, 설문관리, FAQ, Q&A 등 56종
	협업	게시판, 동호회관리, 커뮤니티관리, 주소록관리 등 28종
	시스템 관리	공통코드, 메뉴관리, 로그관리, 기관코드수신 등 25종
	시스템 /서비스연계	연계현황관리, 연계기관관리 등 4종
	통계/리포팅	게시물통계, 접속통계, 보고서통계 등 5종
요소기술서비스 (유틸리티)(91종)		달력, 포맷/계산/변환, 번호유효성/포맷유효성체크 등 91종
모바일 웹 공통컴포넌트 (40종)		게시판, 주소록 등 기존 공통컴포넌트 전환 30종 위치정보연계, 실시간 공지 등 신규 10종

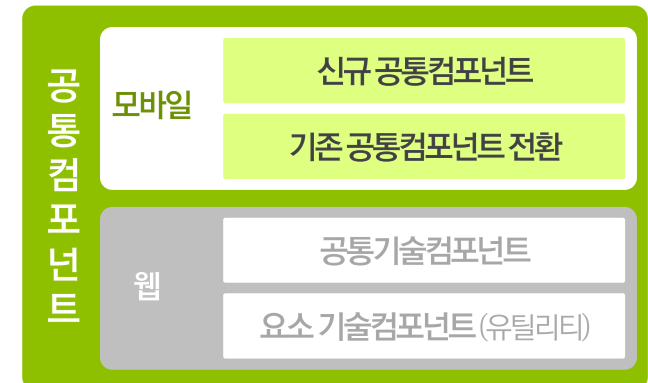
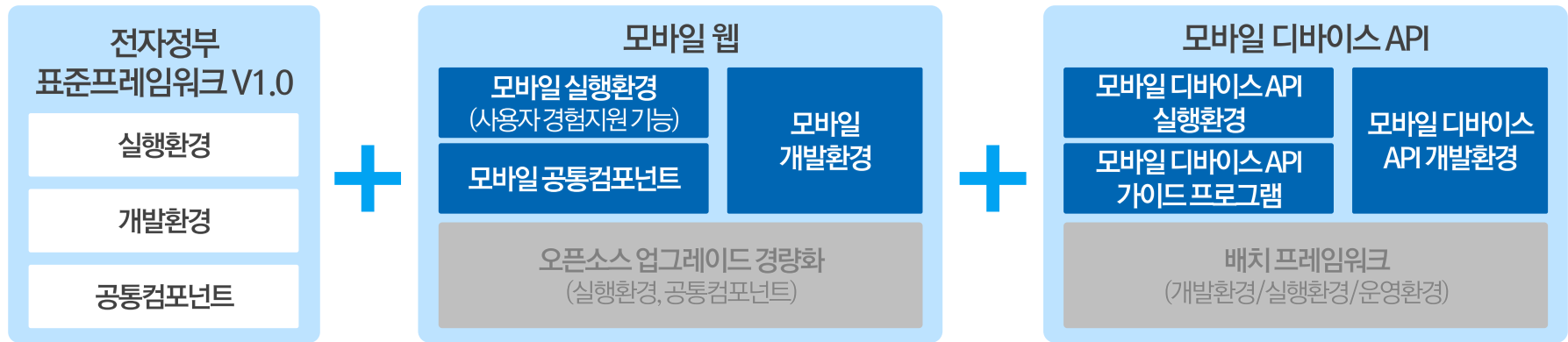
7. 모바일 표준프레임워크

2. 표준프레임워크 구성

모바일 표준프레임워크는 표준프레임워크를 기반으로 모바일 서비스 제공을 위한, 사용자 경험 지원 기능, 모바일 공통컴포넌트 등을 추가적으로 구현한 모바일 웹 프레임워크

전자정부 표준프레임워크 V2.0

전자정부 표준프레임워크 V2.5 이상



3. 표준프레임워크 적용

1. 표준프레임워크 적용방안
2. 표준프레임워크 활용
3. 표준프레임워크 적용사례

1. 표준프레임워크 적용방안

3. 표준프레임워크 적용

표준프레임워크 적용을 위한 절차는 아래와 같음

도입 검토

표준프레임워크가 요구하는 시스템 요건 검토

- JVM(Java Virtual Machine)을 설치하고 운용할 수 있는 서버 플랫폼
- J2EE, JDBC 스펙을 준수한 WAS 및 DBMS에 대한 제약은 없음
- JDK1.8 이상에 최적화되어 있으며, .NET 환경에는 적용 불가
(JDK 1.5인 경우 표준프레임워크 실행환경 2.7 적용 필요)
(JDK 1.6인 경우 표준프레임워크 실행환경 3.1 적용 필요)
(JDK 1.7인 경우 표준프레임워크 실행환경 3.5 적용 필요)
- 자바 기반 웹 어플리케이션 구축 사업에 원칙적으로 적용 가능

사업계획 수립

표준프레임워크 도입에 대한 사업계획 수립

- 표준프레임워크 개념에 대한 이해
(표준프레임워크포털 자료활용)
- 표준프레임워크 도입관련 사업계획서 반영
- 재사용 가능한 공통컴포넌트를 선정하고 소요예산을 조정
- 표준프레임워크 경험 인력 확보 방안 수립

도입 검토

표준프레임워크 적용 전략 수립

- 도입되는 HW/SW 등과의 목표 아키텍처 설계
- POC(Proof Of Concept) 등을 통한 목표아키텍처 검증
- 표준프레임워크 실행환경 기반의 UML 설계
- 개발자 및 SW아키텍트 대상 표준프레임워크 정규 교육 수강

사업계획 수립

표준프레임워크 적용 및 프레임워크 기반 개발

- WAS, DB, 형상관리 등 구축 환경 설정
- 표준프레임워크 실행환경, 개발환경 설치 및 환경 설정
- 프레임워크 템플릿 커스터마이징 (UI 톨과의 연동 템플릿 작성)
- 표준프레임워크 아키텍처 및 개발가이드(개발표준 포함) 수립

1. 표준프레임워크 적용방안

3. 표준프레임워크 적용

기존 개발프레임워크가 있는 경우 표준프레임워크 적용방안



1. 표준프레임워크 적용방안

3. 표준프레임워크 적용

● 라이선스

표준프레임워크는 Apache v2.0, MIT 라이선스로 배포로 자유로운 사용과 기업의 상업적 활용이 가능함

전자정부 표준프레임워크 라이선스

Glueware 라이선스

Apache 2.0 / MIT

OSS의 라이선스

배포 상호주의(Reciprocity) 조항에 따라 기존 OSS 라이선스를 그대로 준용함

LGPL 2.1



EPL 1.0



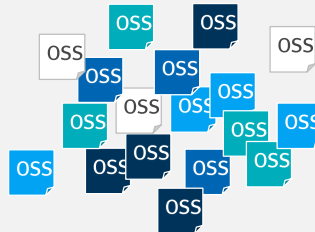
CPL 1.0



MIT License



Apache 2.0



Glueware 란?

OSS를 서로 연결하여 프레임워크의 기능을 수행하도록 하기 위해 개발된 소스코드로서, OSS를 사용하지 않은 순수 개발 소스코드를 말함

상호주의(Reciprocity) 란?

배포에서의 상호주의란 라이선스 적용코드를 제3자에게 배포할 때 원 라이선스와 동일한 라이선스로 배포하도록 요구하는 조항을 말하며, 보통 Copyleft 조항이라고도 함

특징

무료이용

배포허용

소스코드
수정가능

2차적 저작물
공개의무 없음

상업적
이용가능

제약

저작권
표시 의무

수정내용
공지 등

표준프레임워크를 수정하고자 할 시 아래 기준을 고려해야 함

실행환경

임의 변경 불가

- 아파치 2.0 라이선스 준수
- 상호운용성 확보를 위해 변경은 허용되지 않음
- 변경이나 개선 필요 시 표준화 위원회에 변경요청을 하여 검토 후 변경 가능

개발환경

임의 변경 가능
신규개발 가능

- 다양한 개발 환경 및 패턴을 지원하기 위하여 기능 변경 및 추가 가능함
- 구성된 오픈소스 외 상용 솔루션과의 조합으로 사용이 가능함

공통 컴포넌트

임의 변경 가능
신규개발 가능

- 요소기술과 공통기술 컴포넌트를 제공 중
- 업무에 따른 공통컴포넌트를 추가 개발하여 사용 가능함
- 개발된 공통컴포넌트는 별도 관리 또는 공통컴포넌트로 등록하여 공개 사용 가능함

모바일 표준프레임워크

임의변경 불가

- MIT 라이선스 준수
- 상호운용성 확보를 위해 변경은 허용되지 않음

● 적용 지침 (cont.)

표준프레임워크 활용 근거

- 전자정부법 제50조(표준화), 제51조(공유서비스의 지정 및 활용) 및 전자정부법 시행령 제62조(정보자원의 보급·확산)
- 소프트웨어 기술성 평가기준(미래창조과학부 고시 제2014-29호, 2014.4. 10),
기술제안서 평가항목 중 "표준프레임워크 적용"
- 행정자치부 예규 제12호, 전자정부지원사업 관리지침 제35조 (전자정부 표준 개발프레임워크의 활용)
- 행정자치부 고시 [행정기관 및 공공기관 정보시스템 구축운영 지침 제2017-07호] (2017.2.9)

표준프레임워크 적용 조건

- 자바 기반의 웹 응용 시스템(WAS가 존재하는 경우)
- (2.7 기준) JavaEE(J2EE) 5 혹은 JDK1.5 이상의 환경
(3.0 기준) JavaEE(J2EE) 6 혹은 JDK1.6 이상의 환경
(3.5 기준) JavaEE(J2EE) 7 혹은 JDK1.7 이상의 환경
(3.6 기준) JavaEE(J2EE) 7 혹은 JDK1.7 이상의 환경
- 신규 개발시스템으로써, 기존 시스템과 물리적 혹은 논리적으로 구분되는 경우

적용 지침 (cont.)

* 세부 적용 기준 전문은 포탈 관련참고문서에서 다운로드 가능

표준프레임워크 세부 적용 기준 예시

아키텍처 규칙

- Annotation 기반 Spring MVC 준수 : @Controller 및 @RequestMapping을 통한 URL mapping 활용
(View 부분과 model(business logic 및 data) 부분을 controller를 통해 분리)
- Annotation 기반 layered architecture 준수 : 화면처리, 업무처리, 데이터처리에 부분에 대하여 각각 @Controller, @Service, @Repository 활용 (인접 layer간 호출만 가능)
- 업무처리를 담당하는 서비스 클래스(@Service)는 EgovAbstractServiceImpl(또는 AbstractServiceImpl)을 확장하고 업무에 대한 특정 인터페이스를 구현하여야 함
- 데이터처리를 담당하는 DAO 클래스(@Repository)는 EgovAbstractDAO(iBatis) 또는 EgovMapperDAO (MyBatis)를 상속하여야 함 (Hibernate/JPA를 적용한 경우는 예외이며 자세한 사항은 하단 '데이터처리 규칙' 참조)

활용 및 확장 규칙

- 표준프레임워크 실행환경 준수 : 표준프레임워크 실행환경은 적극적으로 활용되어야 함
- 확장 규칙 : 업무 클래스는 egovframework.rte 패키지 내에 정의될 수 없음

2. 표준프레임워크 활용

3. 표준프레임워크 적용

표준프레임워크

전자정부표준프레임워크포탈 (www.egovframe.go.kr)을 통해 표준프레임워크를 다운받아 활용

표준프레임워크 통합개발환경 다운로드



표준프레임워크 설치 (zip파일 복사 및 설정작업)



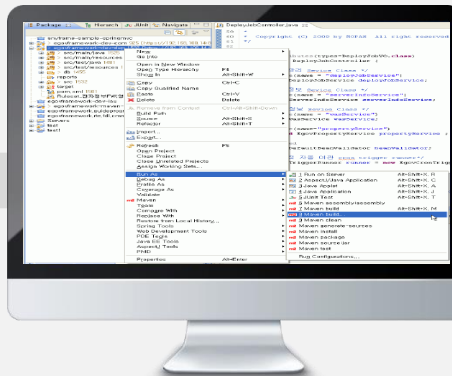
표준프레임워크 템플릿 프로젝트 생성



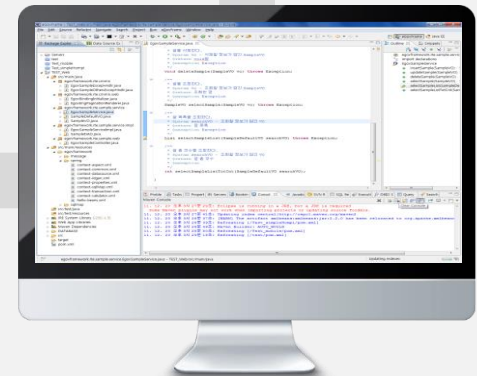
실행 및 테스트



컴파일 및 빌드



업무로직 추가

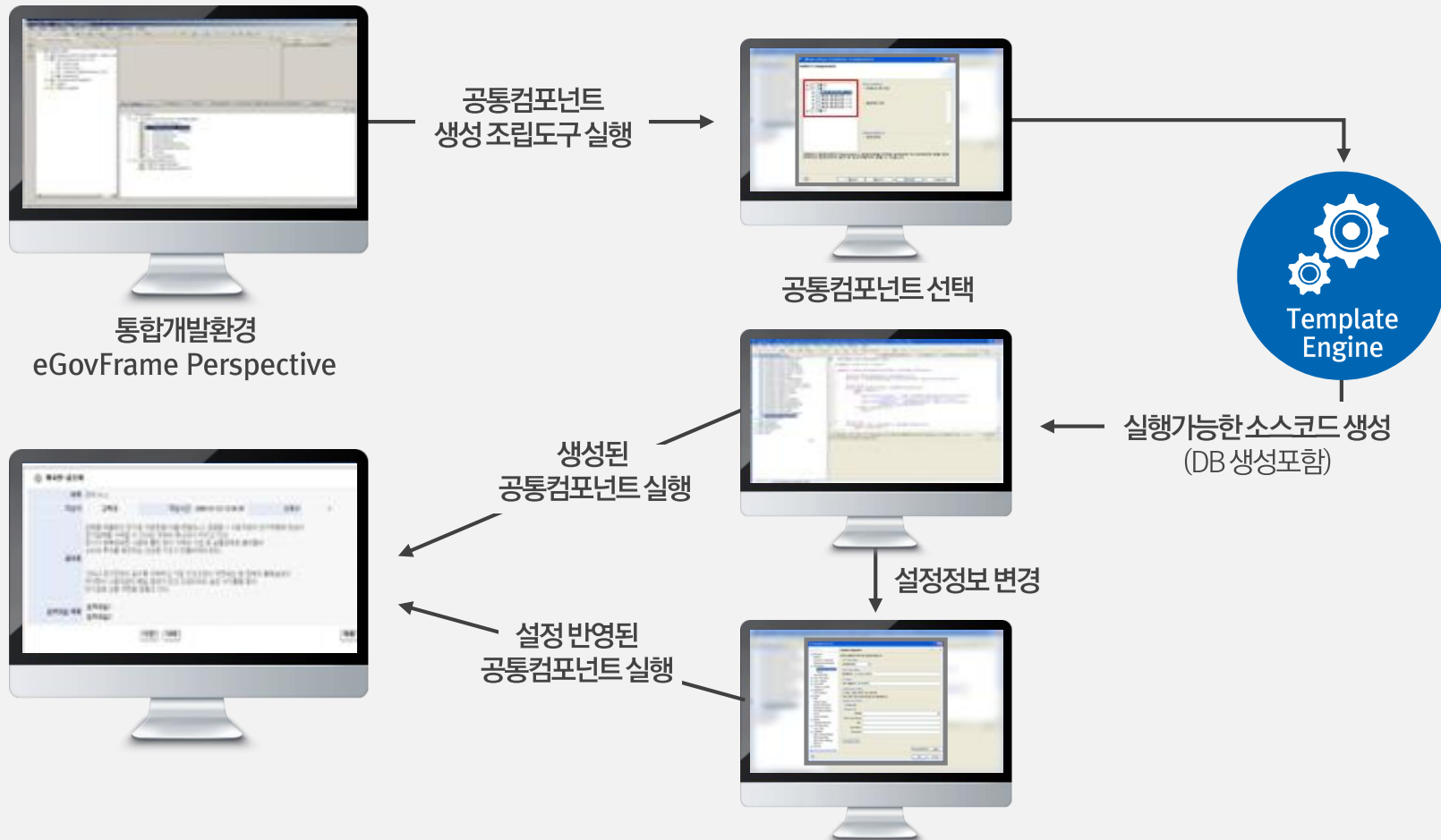


2. 표준프레임워크 활용

3. 표준프레임워크 적용

● 공통컴포넌트

위저드 방식을 이용하여 한번의 클릭으로 공통컴포넌트를 선택 및 설치하는 기능을 제공하며, 공통컴포넌트 설정부터 테이블 생성까지 바로 실행 가능한 코드를 생성하는 기능 제공

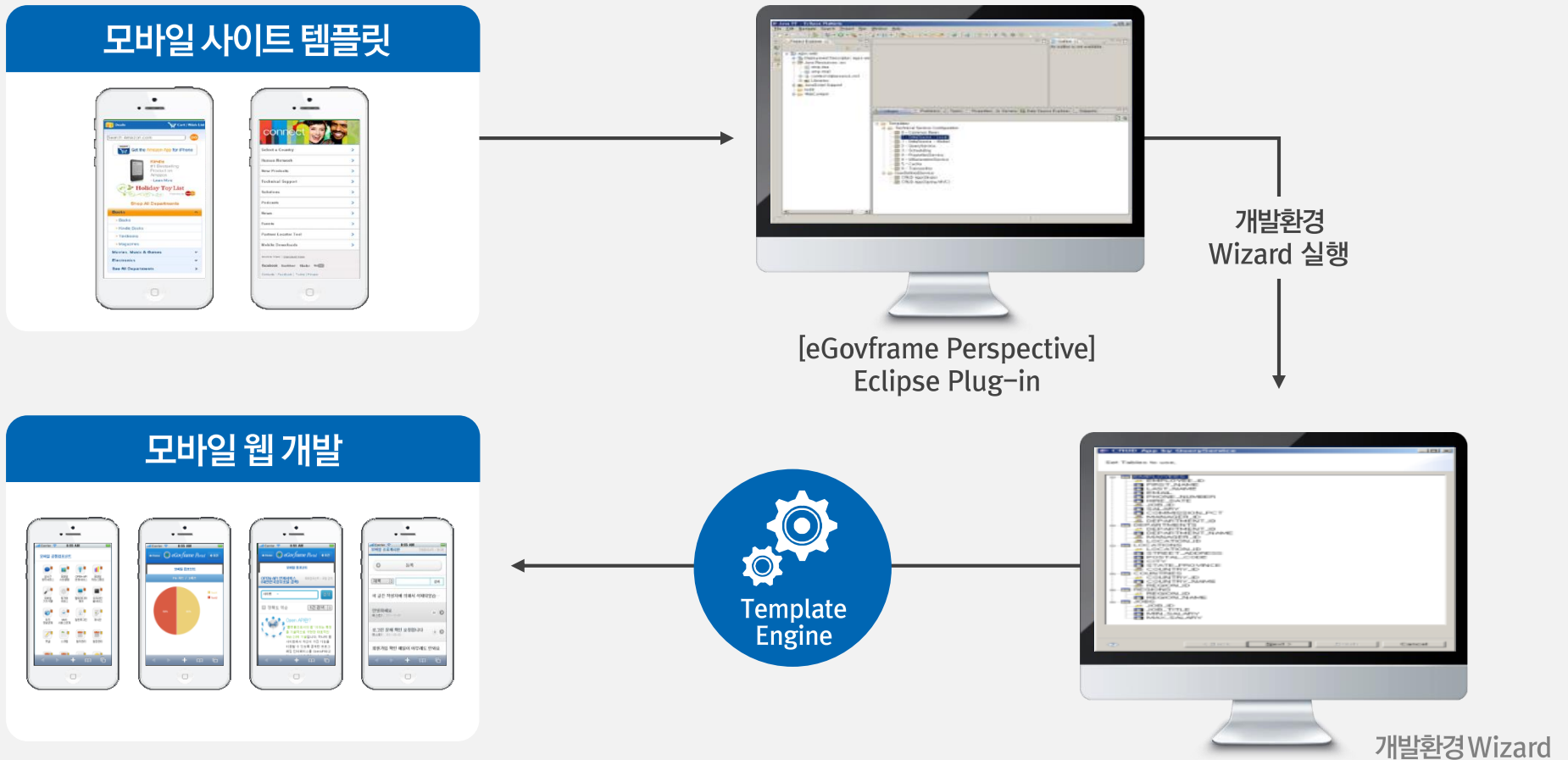


2. 표준프레임워크 활용

3. 표준프레임워크 적용

모바일 표준프레임워크

모바일 표준프레임워크는 전자정부 모바일 웹 사이트 구현을 위한 템플릿을 제공하며, 템플릿 기반의 개발 수행



3. 표준프레임워크 적용사례

3. 표준프레임워크 적용

개요

행정 수요자맞춤형 행정정보공동이용

철도 도시철도이용시스템 구축

교육 충남대학사정보시스템

의료 결행정보통합관리시스템

국방 국방통합원가관리체계

관세 글로벌 물류무역정보망

농업 수입쇠고기유통추적시스템

토지 한국토지정보시스템

지방행정 송파구 정보화사업통합

정보통신 범정부클라우드시범구축

기록물 2010년 기록정보화사업

산업단지 공장설립온라인 지원시스템확산

국세 국세청 ITSM고도화

체육 인천아시안게임 인력 / 모바일 상황관리

주민 주민서비스통합

재난방재 119신고서비스확대

통계 행정자료통합관리시스템

도로 도로공사호환카드정산시스템

복지 독거노인응급안전돌보미

보험 건강보험진료비포털시스템

산림 국가산림정보화기반조성

방송 방송통신정보시스템

특허 3세대 특허넷 구축

검찰 대검찰청홈페이지개편사업

항만 인천항만U단지

물류 글로벌 첨단 항만물류

문화 차세대국회도서관

연금 사학연금연계급여정보시스템

3. 표준프레임워크 적용사례

3. 표준프레임워크 적용

○ 신규 시스템 구축

목표시스템을 구축하기 위한 표준프레임워크 기반 모듈구성도는 다음과 같음



표준 프레임워크

- 표준프레임워크의 개발환경, 실행환경, 공통컴포넌트를 최대한 활용
- 공통컴포넌트를 커스트마이징하고 기타 필요 기능을 추가하여 공통Project 생성
- 소비자포털, 영업자시스템, 공무원시스템은 별개의 독립모듈로 Deploy

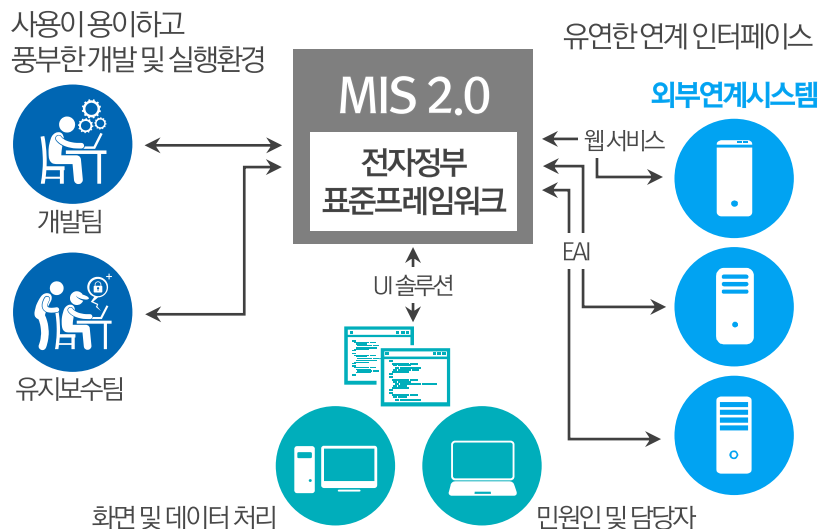
3. 표준프레임워크 적용사례

3. 표준프레임워크 적용

신규 시스템 구축

표준프레임워크는 개발 환경(구현, 테스트, 빌드), 실행환경(화면, 업무, 데이터 처리 등) 및 공통컴포넌트를 활용하여 개발생산성 및 재사용성 향상

개발, 실행환경 제공 및 상호운용성 보장



시스템 요구사항

- OS에 따른 개발 및 운영환경에 영향이 없어야 함 (window 및 unix에 비종속적인 프레임워크 기반 필요)
- Ajax 기반의 플랫폼의 UI툴과의 연동 호환 필요
- 외부기관과의 연계 필요(각 지방해양청, 항만공사등)

Lessons & Learned

적용 결과

- 표준 프레임워크는 사용이 편리하고 기능이 풍부한 개발환경 및 실행환경 활용 (구현(Editing 등), 테스트, 빌드(개인, 통합)환경 제공)
- UI 툴(Ajax 기반의 플랫폼)과의 데이터 연동처리 시 호환성을 제공

시사점

- 화면 UI 툴 업체 Adapter 모듈 적용을 통한 표준프레임워크와의 화면 및 데이터 처리 고려
- 타 기관과의 연계 시 활용되는 기술 중 연계방법 (웹서비스 등)에 따른 표준프레임워크와의 호환성 검토가 필요

3. 표준프레임워크 적용사례

3. 표준프레임워크 적용

자체 프레임워크 구축

다양한 기술 요구사항을 지원하기 위해 전자정부 프레임워크를 확장하여 다음과 같이 정의함

업무 도메인에 특화된 컴포넌트

- 홈쇼핑 업무를 위한 별도의 공통 기능 컴포넌트 개발

X-Internet 기반 공통서비스

- X-Internet을 위한 공통서비스 커스터마이징

UI Adaptor:요구사항 반영 커스트마이징

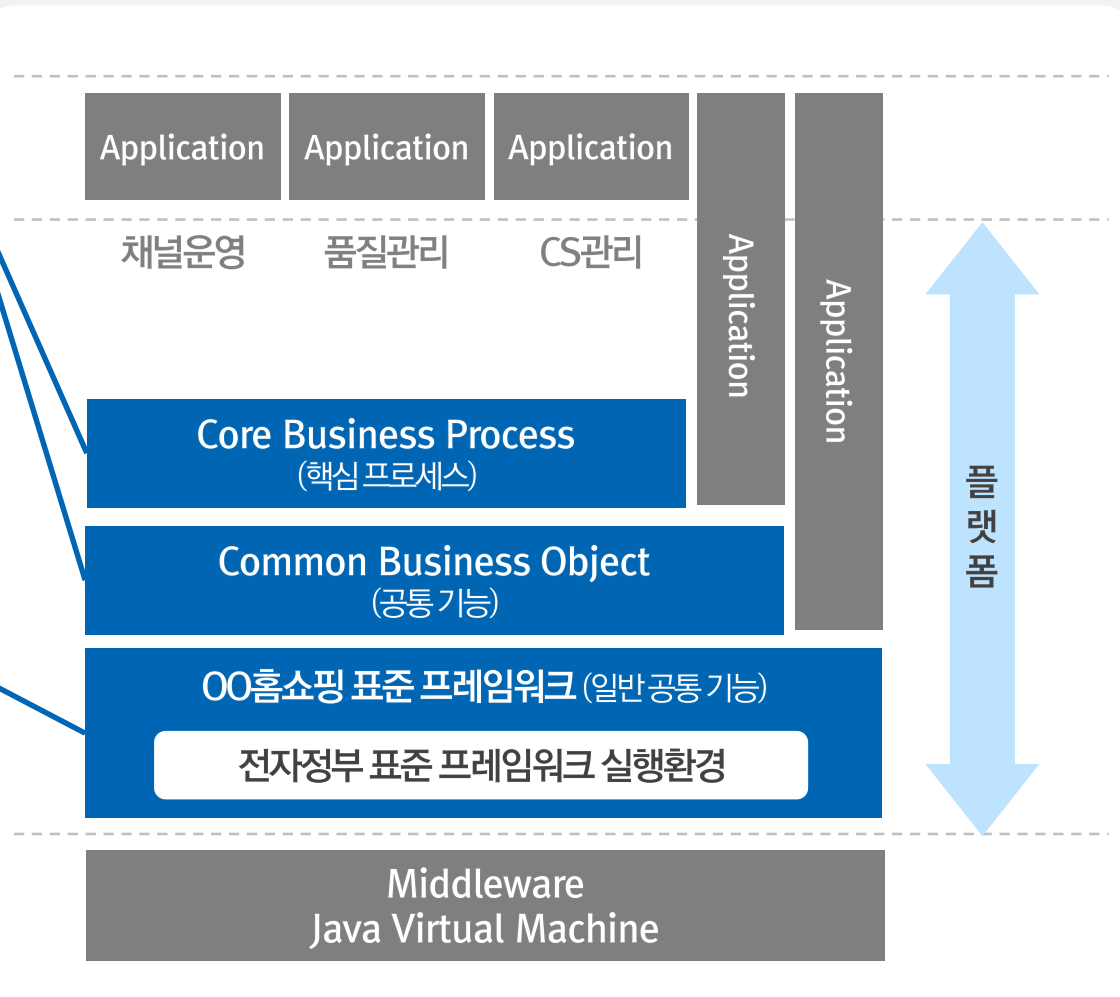
- 다중데이터 처리 기능 정의
- UI Adaptor 예외처리 정의

Security:요구사항 반영 커스트마이징

- XSS, Server Side Include, SQL Injection등 사용자입력값 검증 기능 정의

업무 복잡도를 고려한 아키텍처 모델 정의

- 복잡한 비즈니스 로직 처리를 위한 업무처리 레이어 확장
- 이기종 시스템간 Transaction 처리 고려함



표준프레임워크 개요

Q & A

감사합니다.

