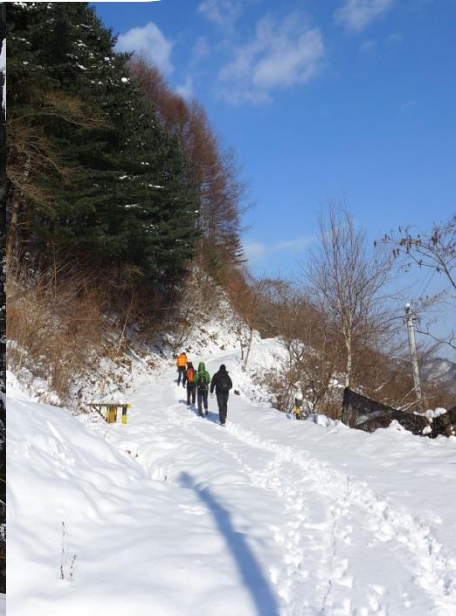




Architecting Process



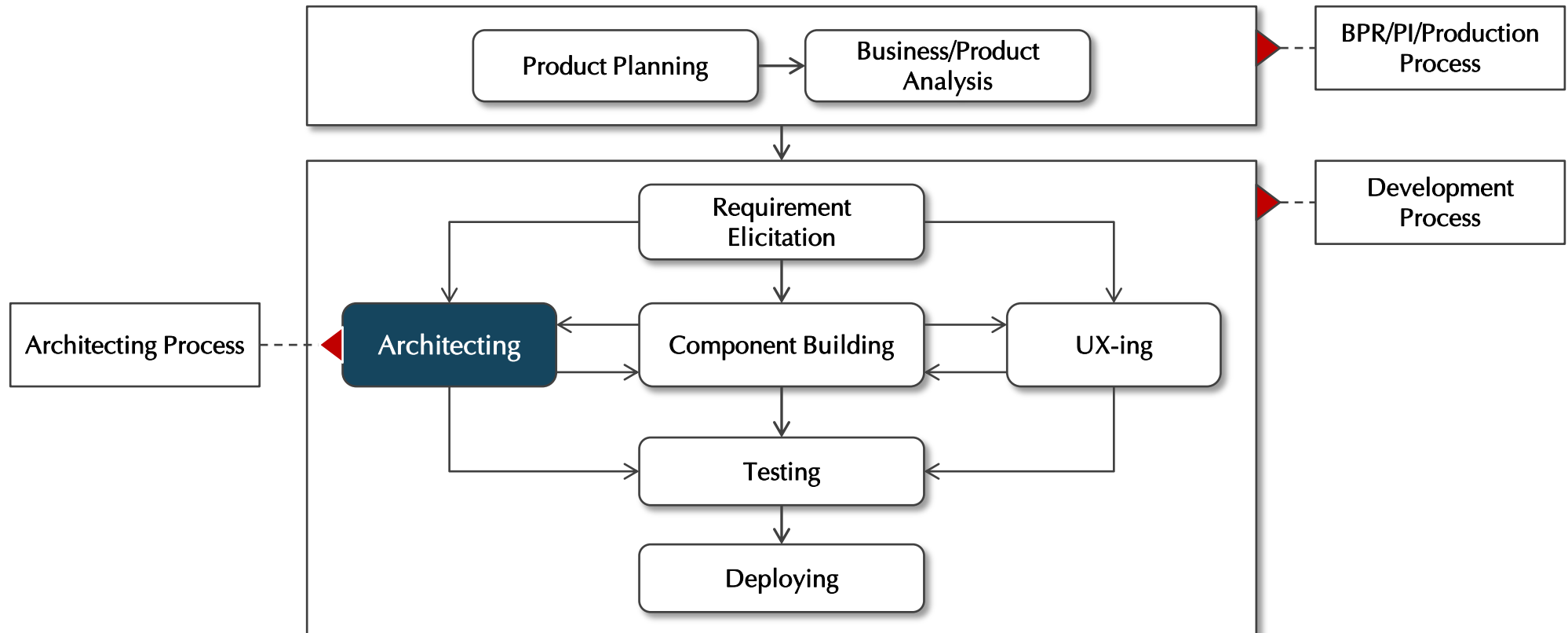


목차

1. Development process
2. Why Architecting Process ?
3. Architecting Process History
4. Reference Architecting Process - ADD
5. IEEE 1471 개념 모델 확장
6. New Architecting Process
7. 토의

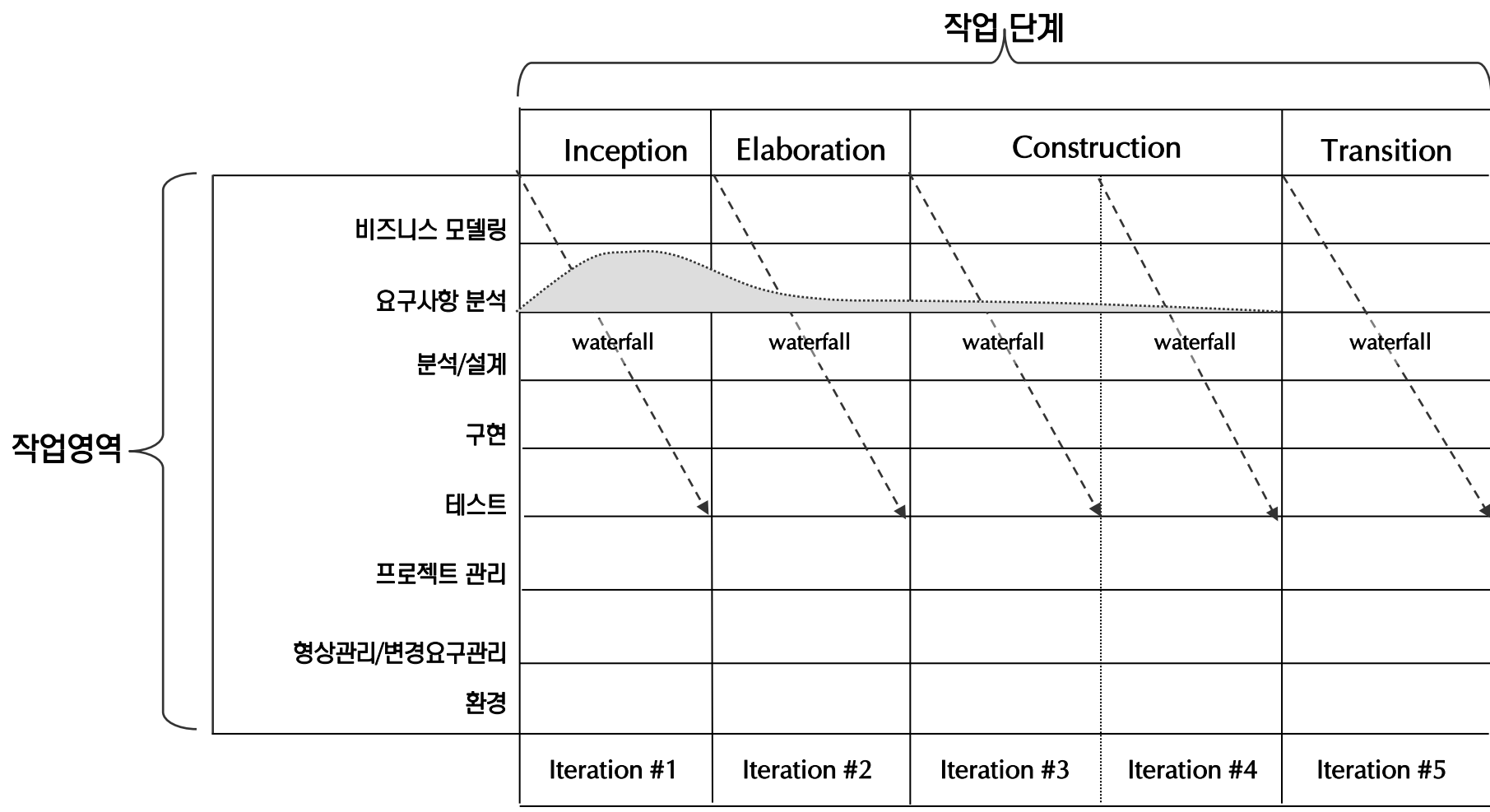
1. Development Process – Nextree 스타일

- ✓ SW를 기반으로 하는 제품이나 업무용 Application을 개발할 때, 다양한 프로세스가 필요합니다.
- ✓ 아키텍팅 활동의 목표 시스템의 틀과 원칙을 설계하는 중요한 활동입니다.
- ✓ 아키텍팅 프로세스는 개발 프로세스의 하위 프로세스이며, 아키텍처 설계, RI 등을 포함합니다.



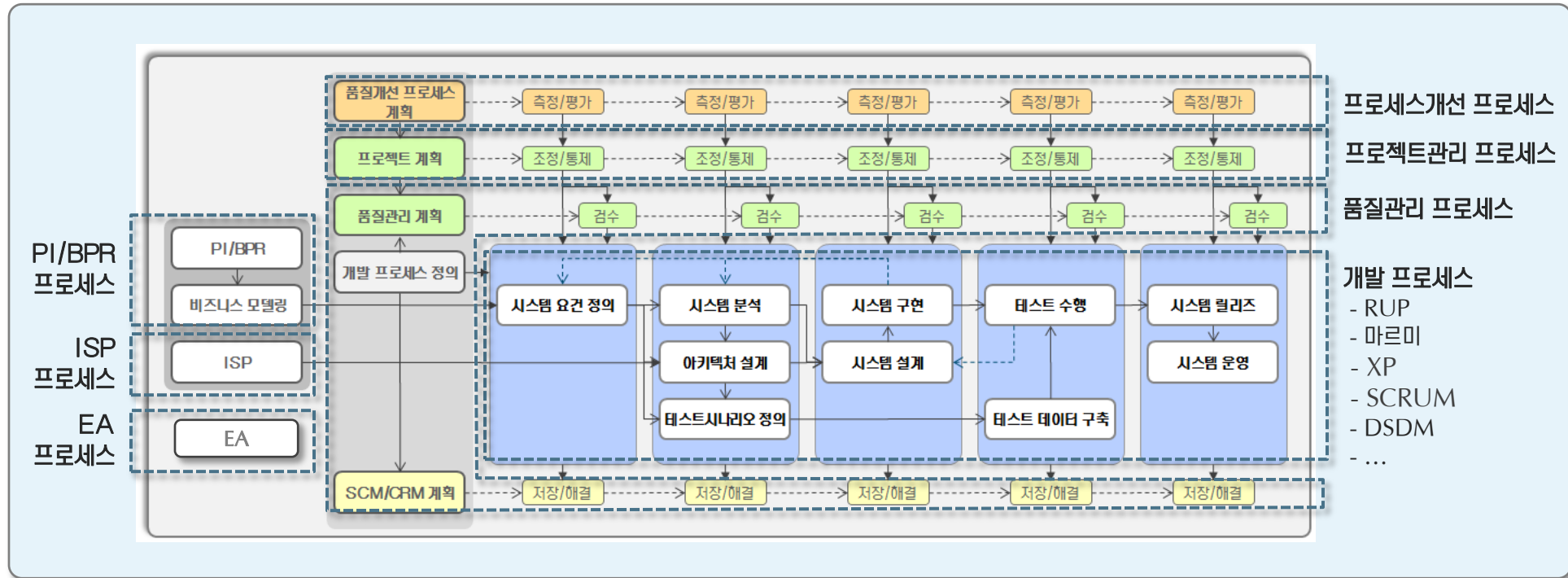
1. Development Process – RUP 스타일

- ✓ RUP는 Objectory를 모태로 하여 가장 완성된 형태의 개발 프로세스로 발전했습니다.
- ✓ Iterative & Incremental 사상을 기반으로 개발을 진행함으로써, Waterfall을 탈피하기 시작합니다.



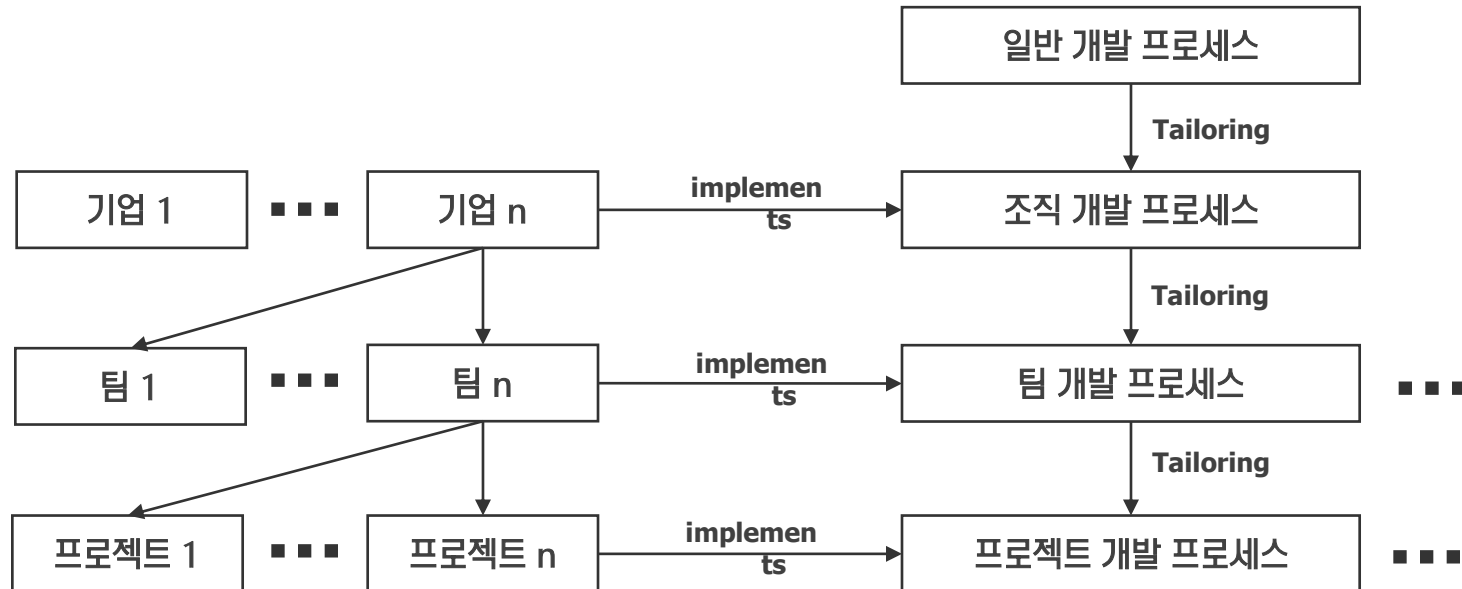
1. Development Process – 대규모 조직

- ✓ 대규모 조직은 다양한 종류의 프로세스를 가지고 있으며, 개발 프로세스는 그 중에 하나입니다.
- ✓ 프로젝트 유형에 따라 개발 프로세스를 선택하여 조정한 후에 사용하는 것이 일반적입니다.
- ✓ 프로세스는 역할 그룹(관리, 개발, 도메인, 솔루션, 시스템 엔지니어링 등) 별로 서로 다릅니다.



1. Development Process – 프로세스 조정

- ✓ 조직 개발 프로세스는 일반 개발 프로세스로부터 조정(tailoring)한 결과입니다.
- ✓ 팀 개발 프로세스는 조직의 표준 개발 프로세스를 팀의 개발작업 특성에 맞추어 놓은 것입니다.
- ✓ 프로젝트 개발 프로세스는 팀의 개발 프로세스를 해당 프로젝트의 특성에 맞추어 놓은 것입니다.



2. Why Architecting Process ?

- ✓ 프로세스는 태권도의 품새(Form)와 비슷합니다.
- ✓ 품새는 순서, 자세를 강조하지만, 실전에서는 스피드, 파워, 순간 판단력이 중요합니다.
- ✓ 실전은 품새가 조정(Tailoring)된 결과로 볼 수 있습니다.



2. Why Architecting Process ?

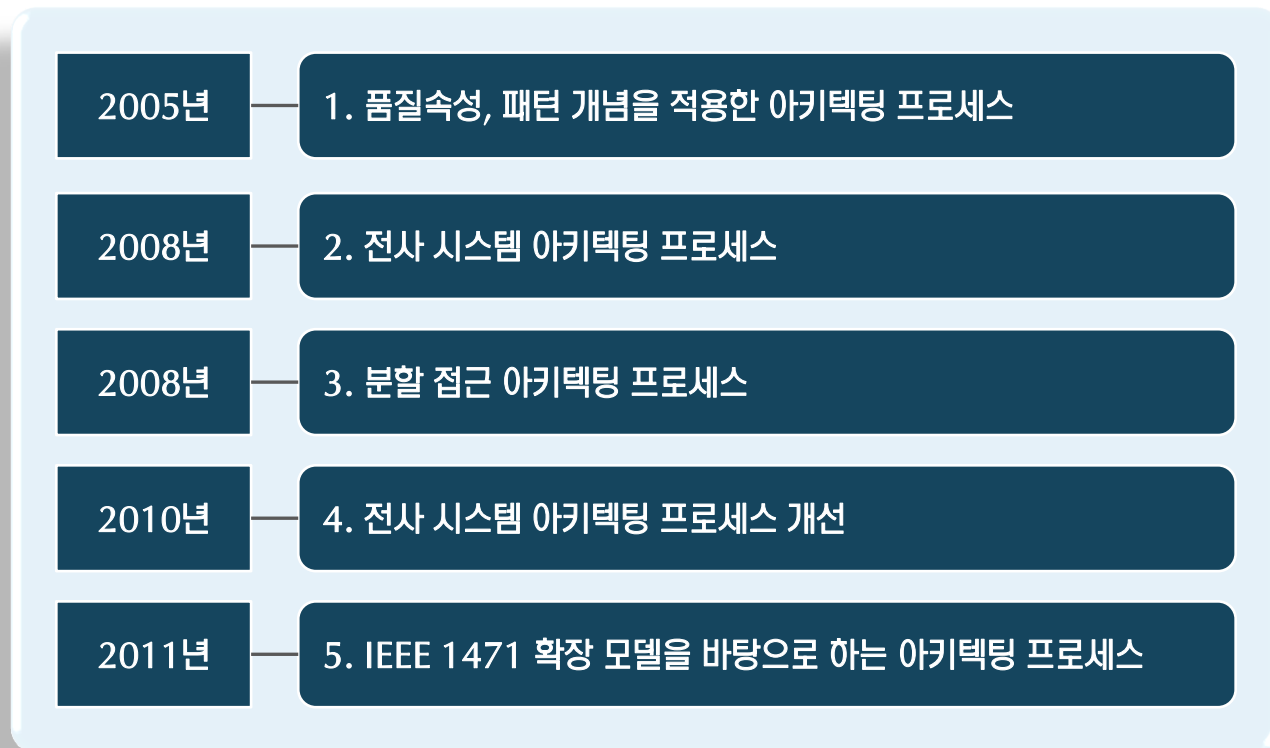
- ✓ 일단 품새가 만들어지면,
- ✓ 경험자의 노하우를 정형화된 형태로 품새에 담을 수 있으며,
- ✓ 수련자는 경험자의 기술을 품새를 통해서 전달받을 수 있습니다.
- ✓ 이런 과정을 통해서 경험과 지식 축적되고 전달되는 선순환 과정을 반복하면서 작업 성숙도가 높아집니다.



▲ 세계태권도연맹(WTF)이 품새 경기에 프리스타일 종목 도입을 추진하고 있다. (자료사진)

3. Architecting Process History (1/5)

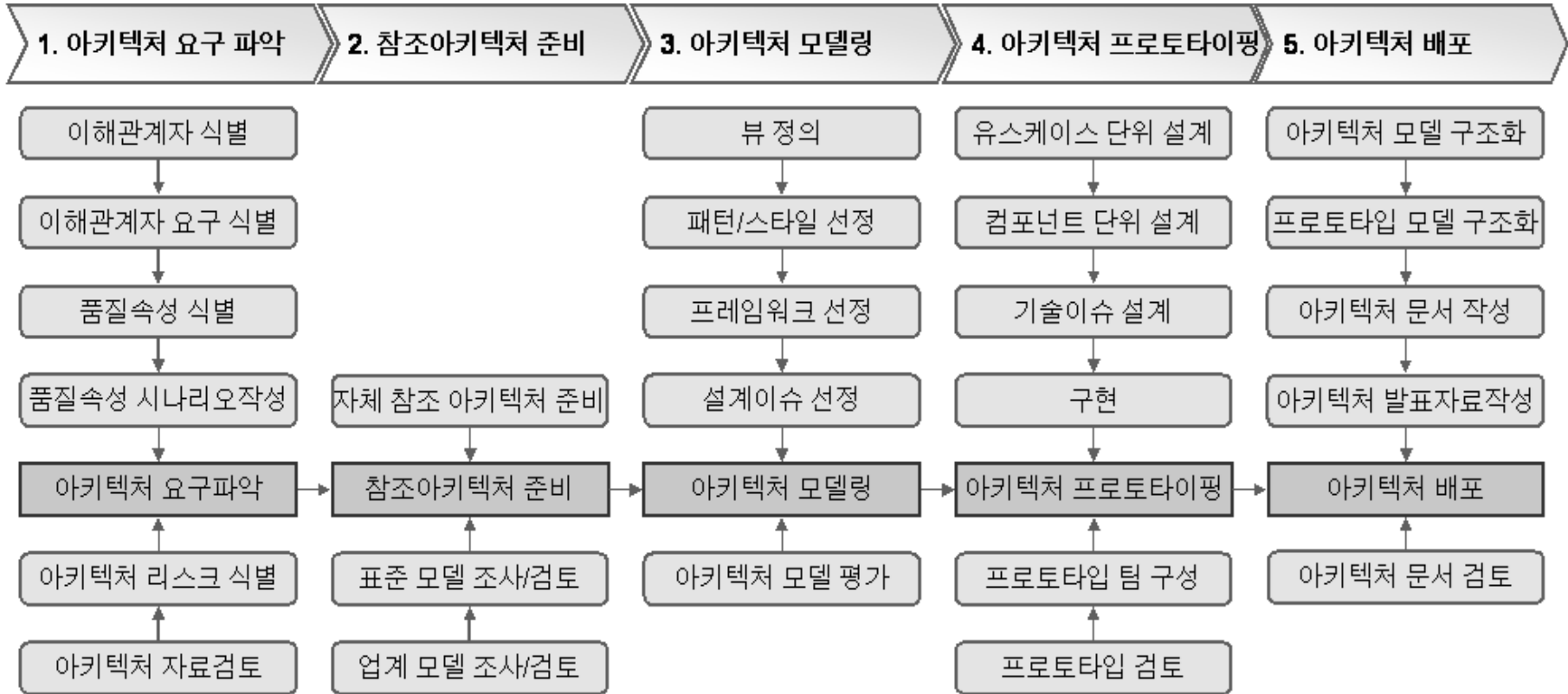
- ✓ 우리는 아키텍팅 프로세스가 필요하다. 경험을 축적하고 전달하는 수단으로.
- ✓ 2005년 이후 다섯 차례에 걸쳐 아키텍팅 프로세스를 발전시켜 왔습니다.
- ✓ 개선할 때 마다, 품질 속성, 전사 시스템, 분할 설계 등의 키워드가 있었습니다.
- ✓ 최근에는 IEEE 1471 – 확장모델을 기반으로 개발 프로세스를 개선하였습니다.



3. Architecting Process History (2/5)

- ✓ 2005년 금융회사의 차세대 프로젝트 과정에서 정의했습니다.
 - ✓ 품질속성, 패턴, 프로토타입을 특징으로 하는 프로세스입니다.
 - ✓ 전사 범위로 기획을 했으나 전사 시스템 아키텍팅 프로세스의 순환적인 특징을 반영하지 못합니다.
- 2005년

1. 품질속성, 패턴 개념을 적용한 아키텍팅 프로세스

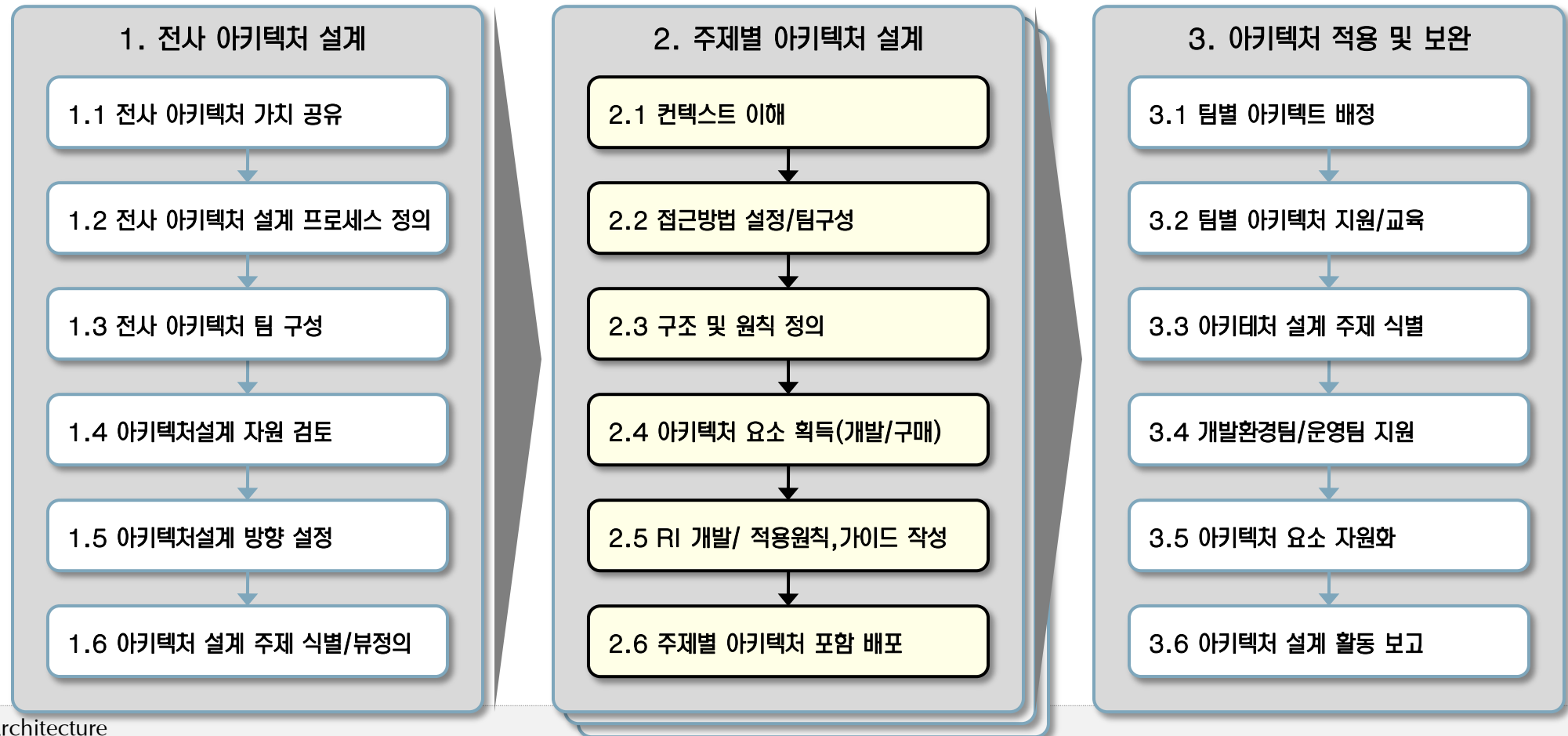


3. Architecting Process History (3/5)

- ✓ 전사 시스템 아키텍팅의 순환적인 특성을 반영했습니다.
- ✓ 독립형 시스템 아키텍팅 프로세스는 [아키텍처 요소 획득]으로 한정하였습니다.
- ✓ 아키텍처 적용(enforcement) 개념을 프로세스에 도입하였습니다.

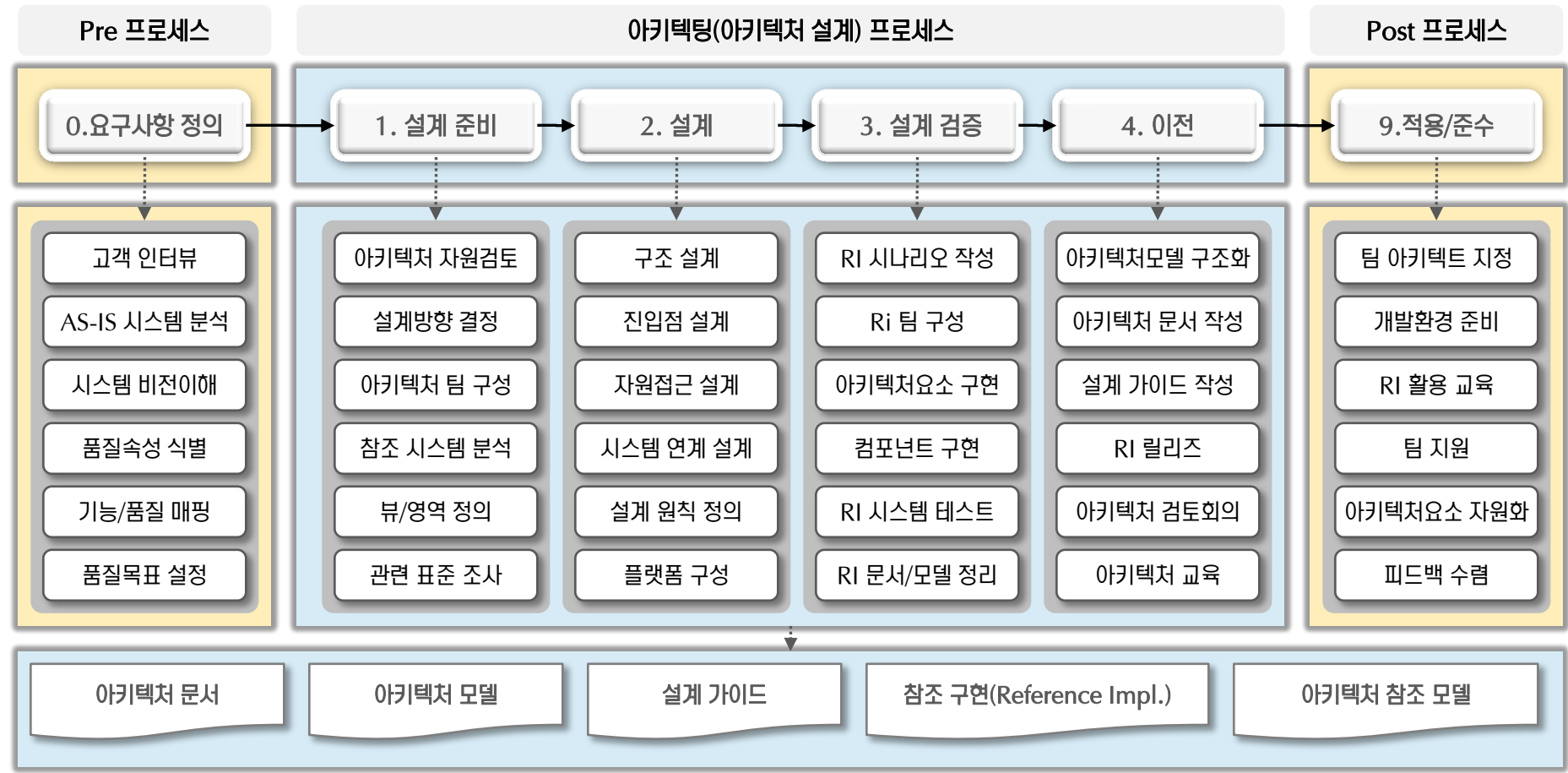
2008년

2. 전사 시스템 아키텍팅 프로세스



3. Architecting Process History (4/5)

- ✓ 독립형 애플리케이션 아키텍처 설계 시 분할-정복개념 적용하였습니다. 2008년 3. 분할 접근 아키텍팅 프로세스
- ✓ 구조, 진입점, 자원접근, 시스템 연계, 플랫폼 적용, 설계 원칙 등으로 나누어 설계하였습니다.
- ✓ 뷰는 4 + 1 로 고정하였습니다.



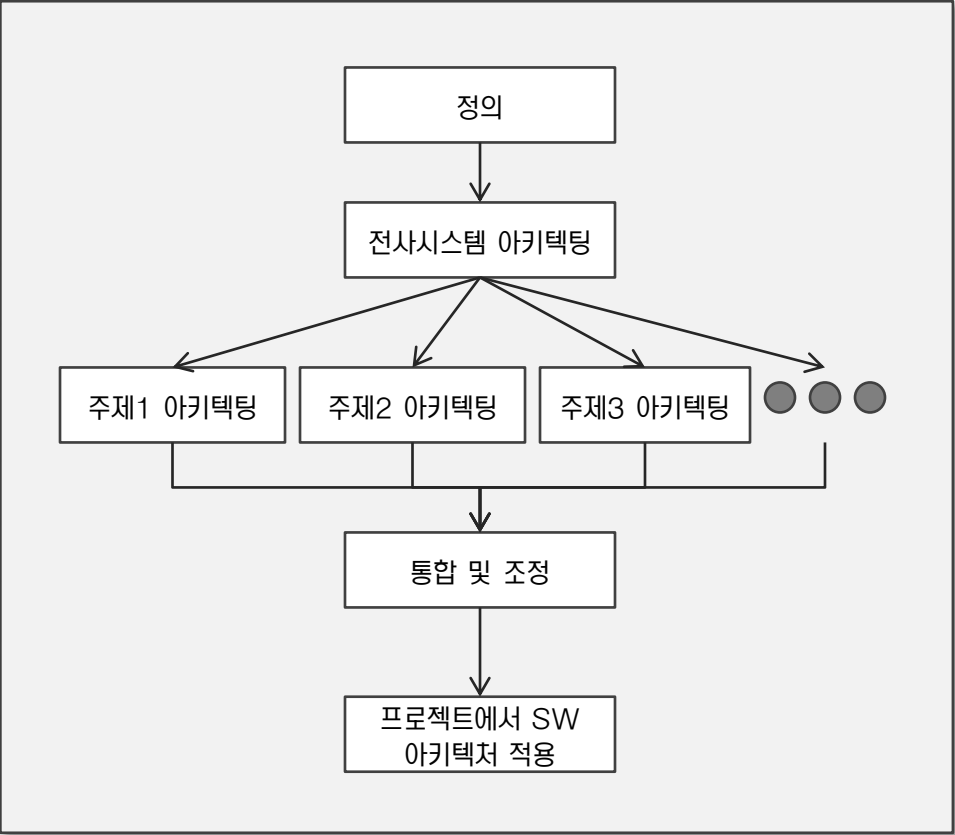
3. Architecting Process History (5/5)

- ✓ 기존의 전사 시스템 아키텍팅 프로세스를 개선하였습니다.
- ✓ 00사 차세대 프로젝트 준비를 위해 기존의 아키텍팅 프로세스를 개선하였습니다.
- ✓ 주제별 아키텍팅이라는 개념을 그대로 사용했습니다.

2010년

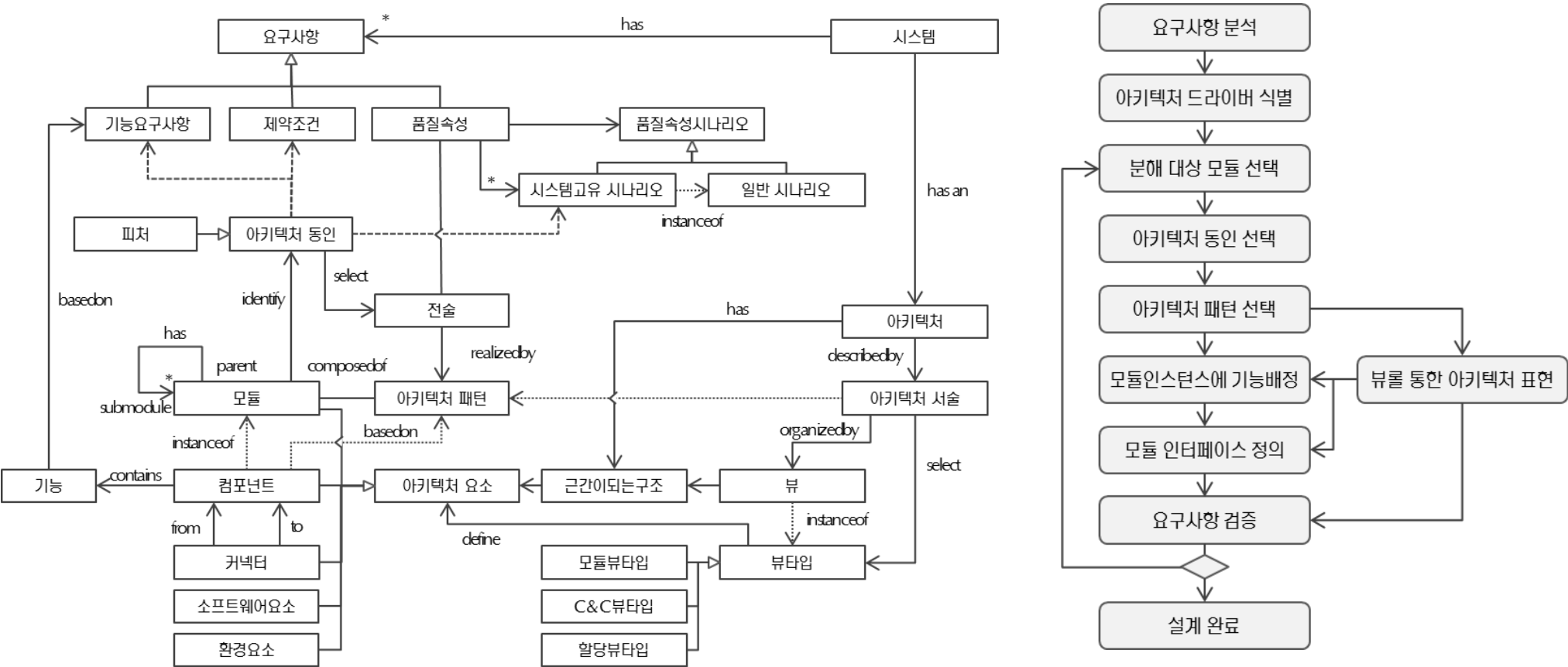
4. 전사 시스템 아키텍팅 프로세스 개선

1. 정의	1.1 다양한 아키텍처 개념 정의 및 조정
	1.2 전사 시스템 아키텍팅 프로세스 정의
	1.3 아키텍팅 접근방법 정의 - 뷰와 모델
2. 전사 시스템 아키텍팅	2.1 AS-IS 시스템 아키텍처 표현 및 분석
	2.2 전사 시스템 아키텍처 설계
	2.3 아키텍팅 주제 식별 및 우선순위 결정
3. 주제별 아키텍팅	3.1 주제 영역 연구 및 접근방법 결정
	3.2 SW 아키텍처 설계
	3.3 아키텍처 요소 획득
	3.4 RI를 통한 검증
4. 통합 및 조정	4.1 주제별 아키텍팅 결과 통합 및 조정
	4.2 전사 아키텍처 가이드 업데이트
	4.3 아키텍처 적용(enforcement) 계획 수립

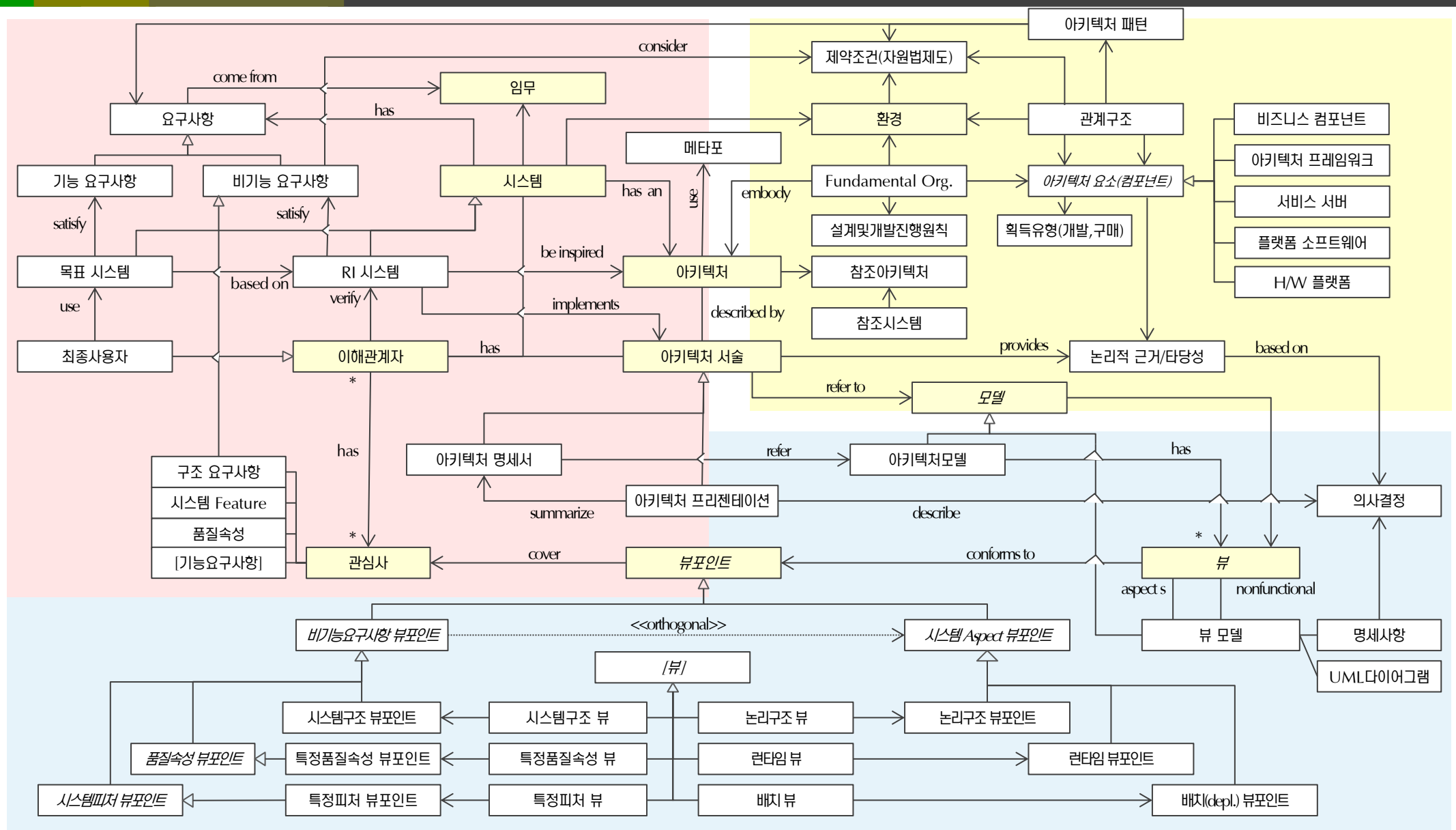


4. Reference Architecting Process – ADD

- ✓ ADD 프로세스는 현대의 다양한 아키텍팅 프로세스의 바탕이 되는 프로세스입니다.
- ✓ 소규모 시스템 또는 Embedded 시스템 개발에서는 아직도 활발하게 사용되는 프로세스입니다.
- ✓ 프로젝트에서 개발하려는 목표 시스템과 개발팀의 특성에 맞추어 조정(tailoring) 하여야 사용합니다.



5. IEEE 1471 – 개념 모델 확장

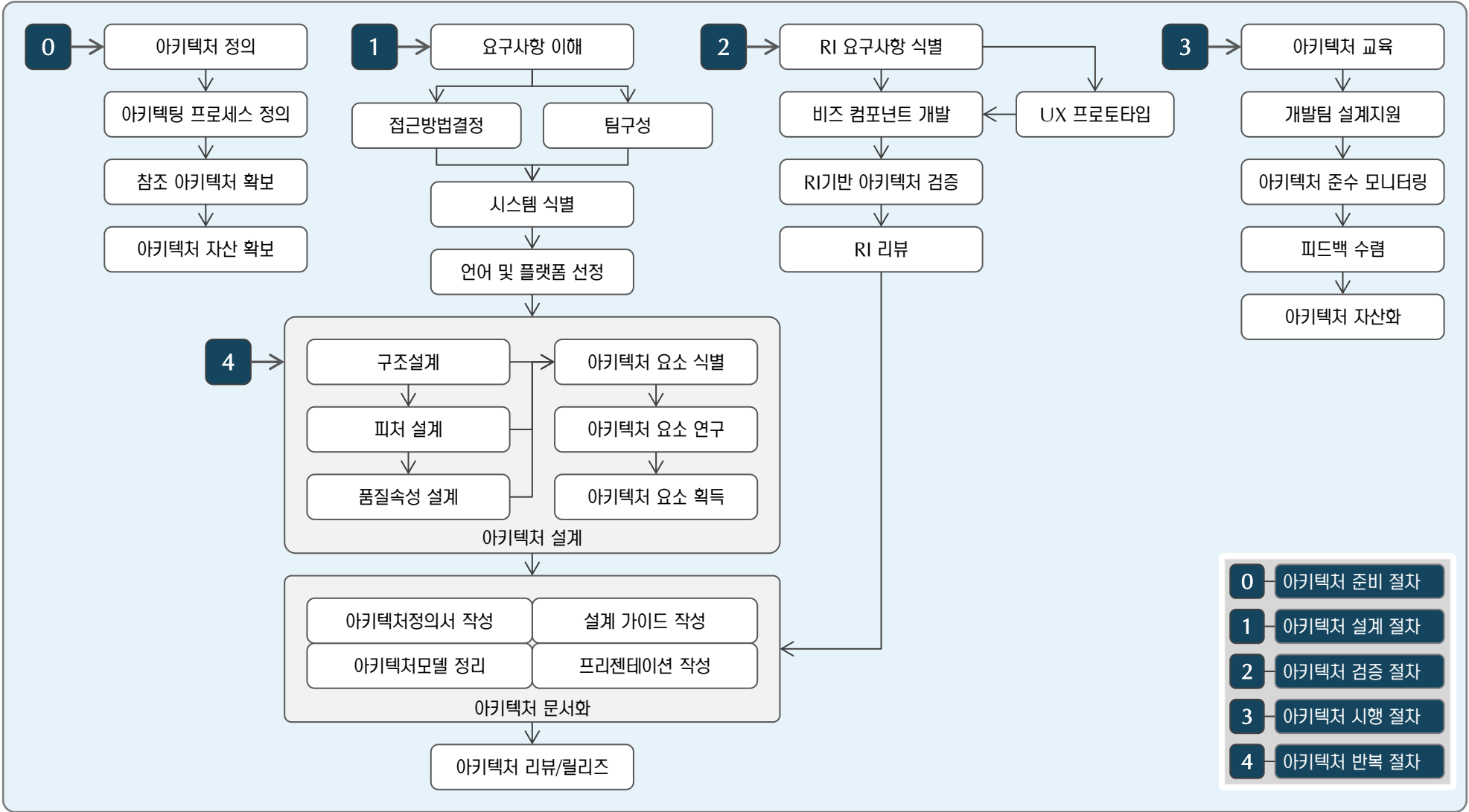


6. New Architecting Process

✓ 아키텍팅 프로세스는 IEEE 1471 확장 모델을 기반으로 합니다.

2011년

5. IEEE 1471 확장 모델 기반 아키텍팅 프로세스



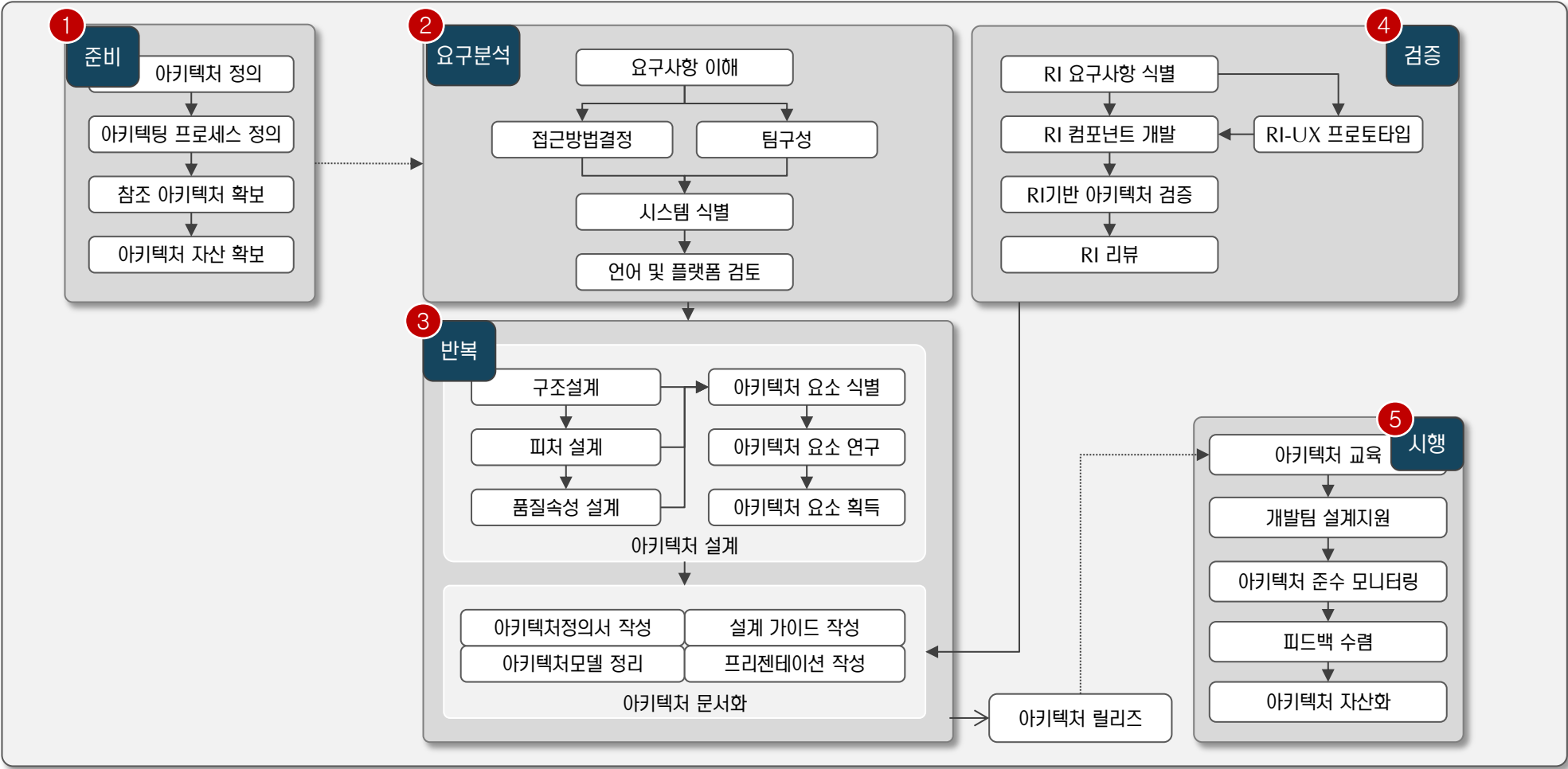
- 0 아키텍처 준비 절차
- 1 아키텍처 설계 절차
- 2 아키텍처 검증 절차
- 3 아키텍처 시행 절차
- 4 아키텍처 반복 절차

6. New Architecting Process(refined)

✓ 아키텍팅 프로세스는 IEEE 1471 확장 모델을 기반으로 합니다.

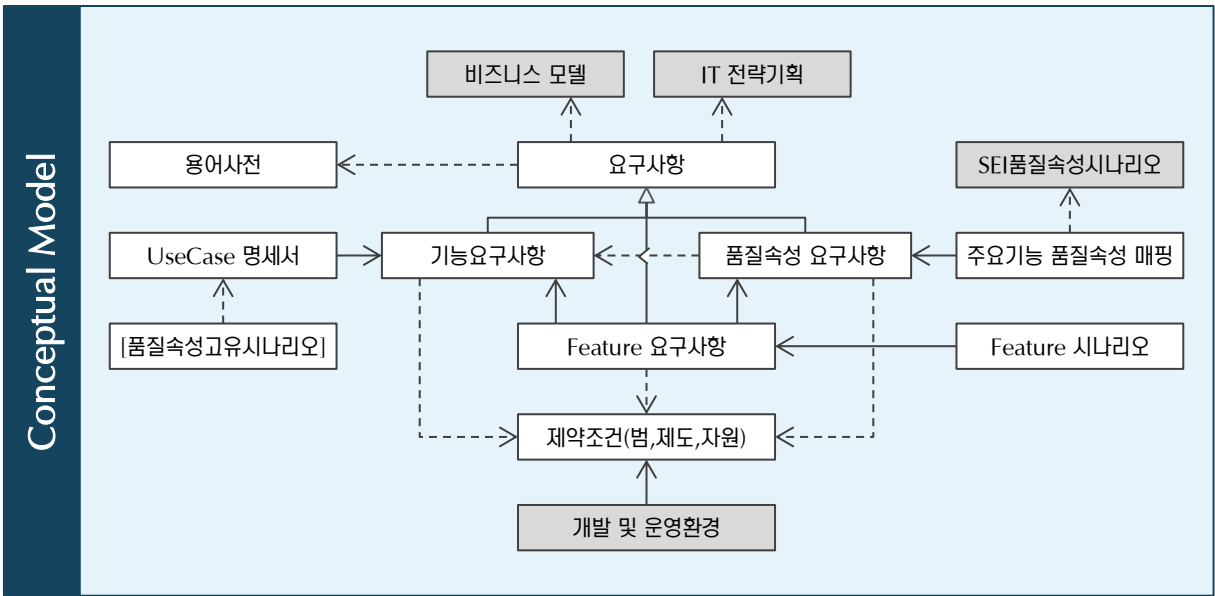
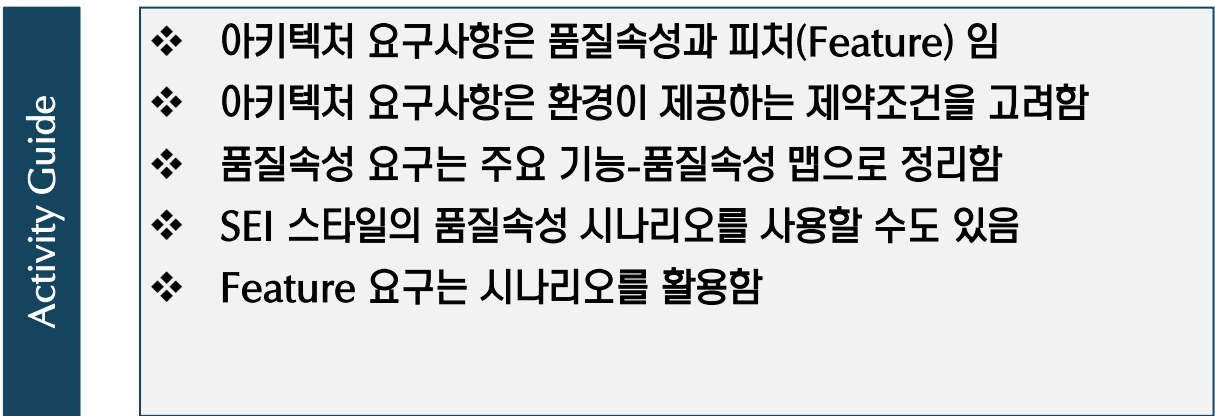
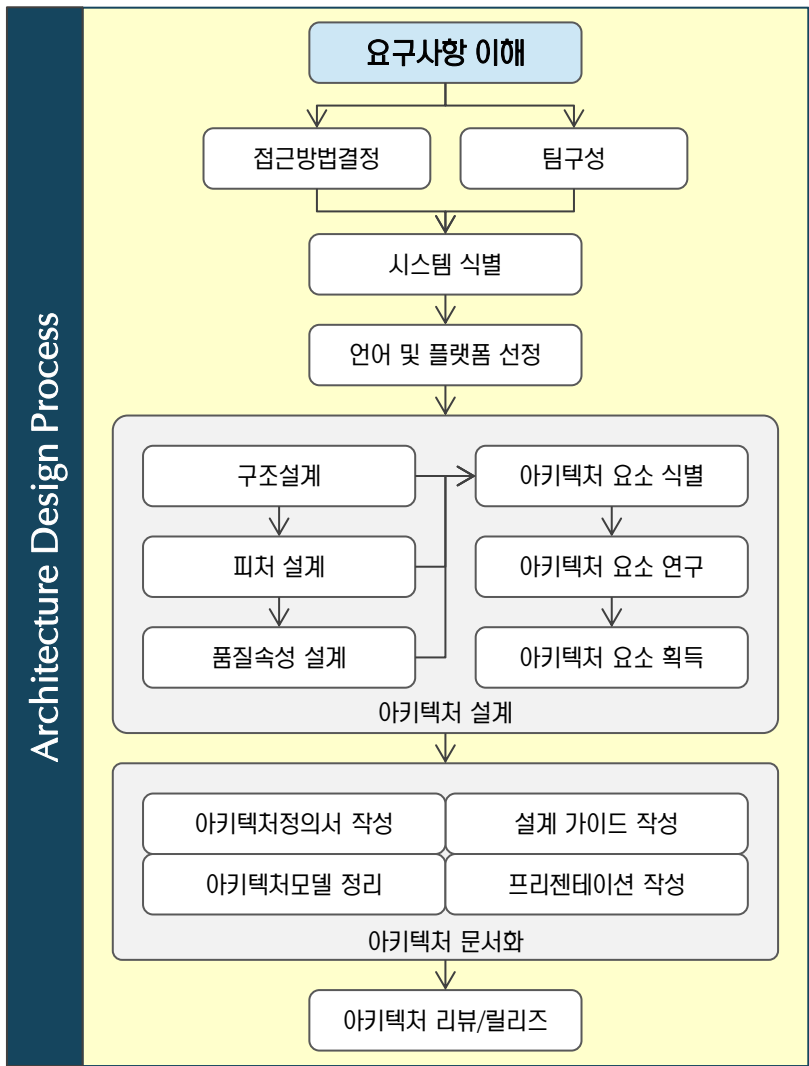
2011년

5. IEEE 1471 확장 모델 기반 아키텍팅 프로세스



6. New Architecting Process – 요구사항 이해

✓ 아키텍처 요구사항을 이해하고 기록합니다.



6. New Architecting Process – 요구사항 이해

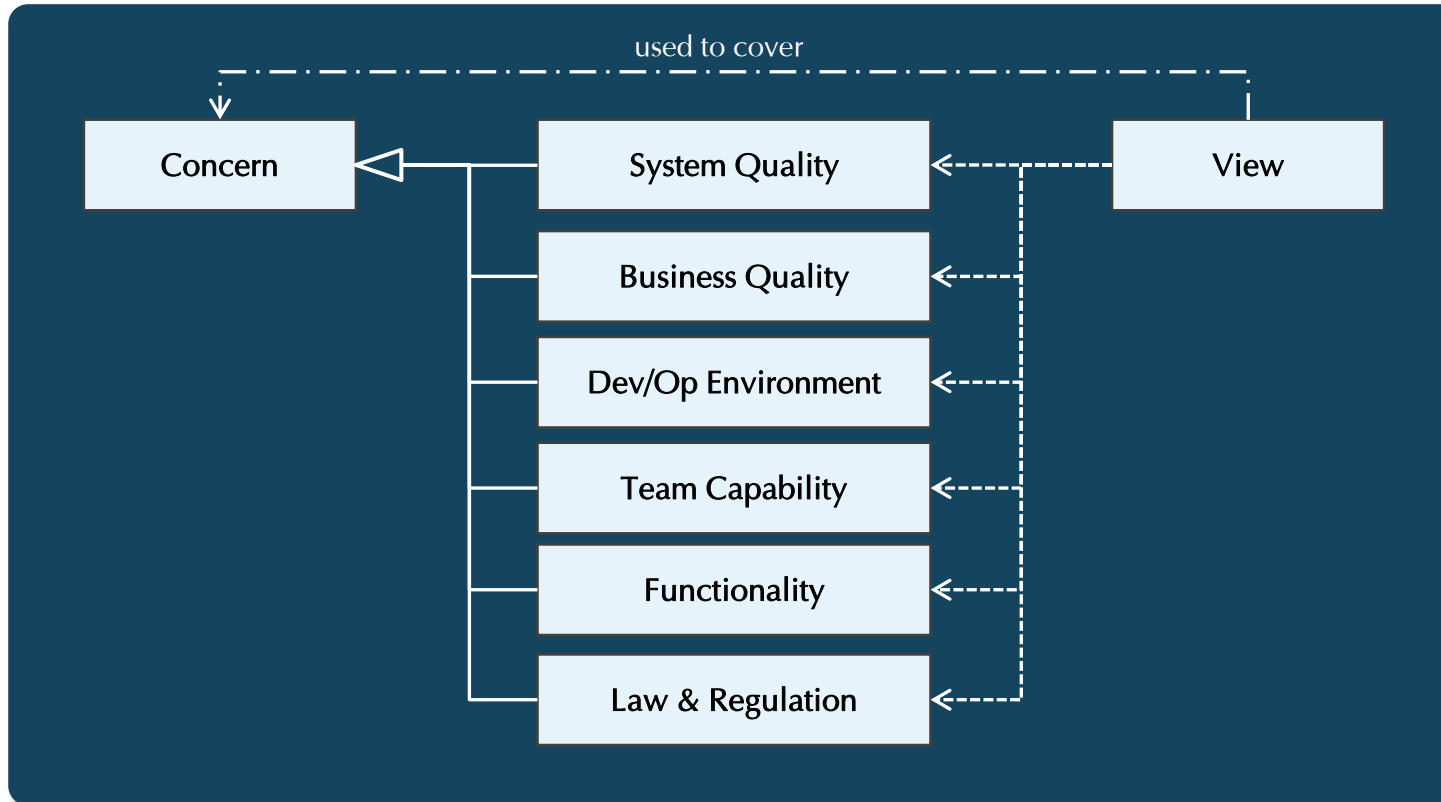
- ✓ 품질과 기능의 차이와 역할에 대한 이해는 아키텍처 설계의 출발점입니다.
- ✓ 기능과 품질은 직교(orthogonal) 함
 - 상판:기능
 - 교각:품질
- ✓ 품질은 주로 Cross-cutting 이슈입니다.
- ✓ 두 경계선이 흐릿한 경우도 있습니다.
- ✓ 피처(Feature)는 두 속성의 조합입니다.

[세계에서 가장 높은 다리인 미요교(Le Viaduc de Millau)]



6. New Architecting Process – 요구사항 이해

- ✓ 이해관계자의 관심사는 매우 다양하며, 시스템 영역에서 포함되는 것과 그렇지 않은 것들이 있습니다.
- ✓ 여러 관심사를 특정 기준으로 묶음짓고 이를 바라보는 관점을 정의하는데 그것을 뷰(view)라고 합니다.
- ✓ 관심사는 포괄적인 개념으로 시스템 품질이 주를 이루고 있지만, 기능성, 비즈니스 환경, 운영 환경, 법과 제도를 포함하고 있습니다.



6. New Architecting Process – 요구사항 이해

- ✓ 품질 속성은 조직 별, 컨텍스트 별로 서로 다른 정의를 가질 수 있습니다.
- ✓ 모든 품질 속성은 관심사가 될 수 있지만, 모든 관심사가 품질 속성은 아닙니다.
- ✓ 프로젝트 팀은 목표 시스템과 관련된 품질 속성을 정의함으로써 의사소통을 원활하게 할 수 있습니다.

시스템의 품질 속성(From SEI)

- 가용성(availability)
- 변경 용이성(modifiability)
- 수행성능(performance)
- 보안성(security)
- 테스트 용이성(testability)
- 사용성(usability)
- 상호운용성(interoperability)
- 이식성(Portability)
- 범위성(Scalability)

다양한 품질속성...

비즈니스의 품질속성 (From SEI)

- 시장 출하 시기(time to market)
- 비용과 장점(cost and benefit)
- 시스템의 예정 생명주기(projected lifetime of the system)
- 목표 시장(targeted market)
- 최초 공개 일정(rollout schedule)
- 기존 시스템과의 통합(integration with legacy systems)

시스템의 품질 속성(From TOGAF)

- Availability
 - manageability
 - serviceability
 - performance
 - reliability
 - recoverability
 - locatability

Assurance:

- security
- integrity
- credibility

Usability

- International operation

Adaptability

- interoperability
- scalability
- portability
- extensibility

6. New Architecting Process – 요구사항 이해

- ✓ 하나의 기능은 여러 품질 속성과 관련이 있습니다.
- ✓ 서비스(기능) 제공 목표를 달성하기 위해, 요구되는 품질 목표를 달성하여야 합니다.
- ✓ 때로는 품질 목표가 기능 설계 또는 업무 구조 또는 절차와 관련이 있을 수 있습니다.
- ✓ 따라서, 아키텍팅 활동을 위해 특정 업무 기능에 대한 이해가 필요할 수도 있습니다.

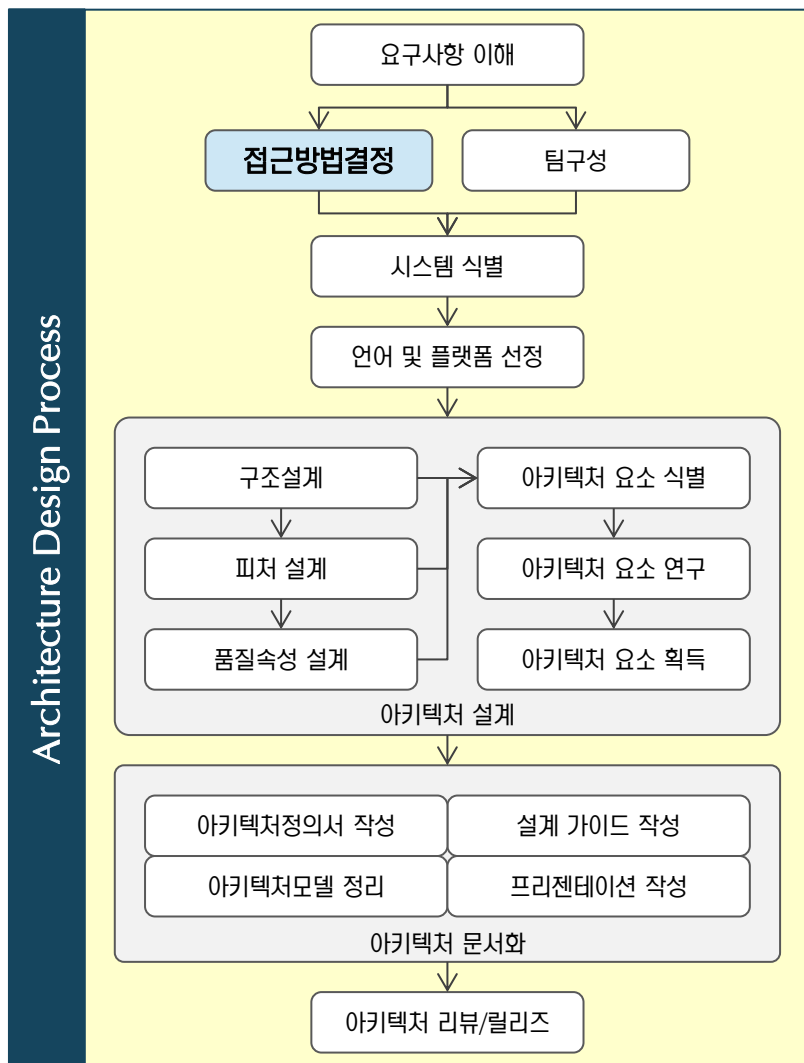
구분			경매등록	경매물품조회	입찰	낙찰금액지불	경매마감
품질속성	명세내용	단위					
가용성	중요도	상중하	중	상	상	중	상
	허용범위	다운시간/년	30H	10H	20H	30H	1H
성능	중요도	상중하	하	상	상	하	상
	허용범위	초(sec)	3	1	1	5	1
범위성	중요도	상중하	하	상	중	하	하
	허용범위	동시 사용자수	5천	10만명	1만명	5천명	5천명
보안	중요도	상중하	중	하	중	상	하
	허용범위	인증/인가/암호화	인증/인가	인증	인증/인가	인증/인가/암호화	해당사항없음
확장성	중요도	상중하	중	상	중	중	하
	허용범위	작업일	3	3	3	5	3
관리성	중요도		Cross-cutting 이슈				
	허용범위						
사용성	중요도	상중하	중	상	상	중	하
	허용범위	사용자만족도	80%	95%	90%	80%	해당사항없음

[주요 기능과 품질 속성 간의 연관관계]



6. New Architecting Process – 접근방법 결정

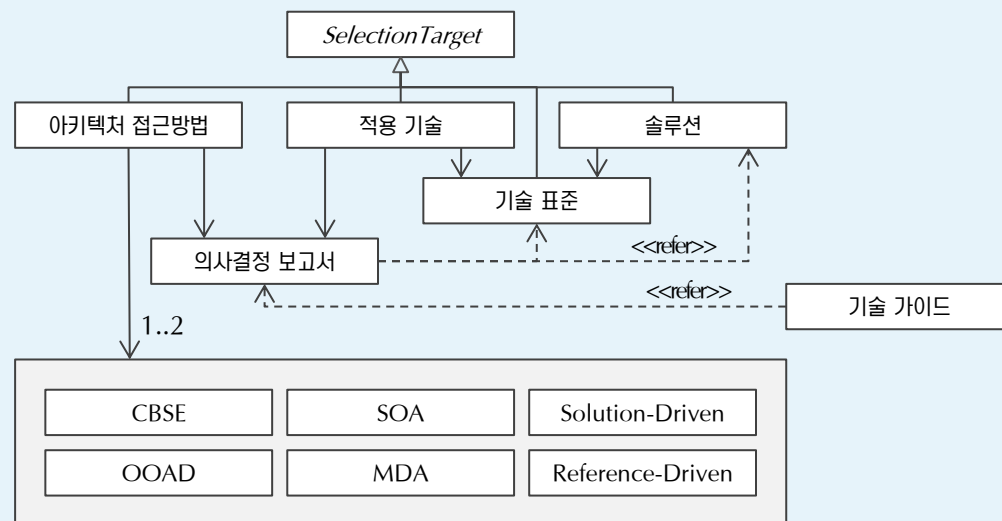
✓ 시스템 구축을 위한 접근 방법, 기술, 표준, 솔루션 등에 대한 방향을 결정합니다.



Activity Guide

- ❖ 기준이 되는 접근방법을 의사결정보고서를 통해 결정함
- ❖ 적용기술, 관련 표준, 핵심 솔루션 등에 대한 Study를 진행함
- ❖ [주의] 초기에 모든 솔루션을 검토할 필요는 없음
- ❖ 이 활동이 종료되면 접근방법에 대한 공감대가 형성되고, 주요 기술, 표준, 솔루션 등에 대한 기본적인 이해를 공유함
- ❖ 마케팅용 Keyword를 걸러내고 팀을 올바른 방향으로 리드하는 활동

Conceptual Model



6. New Architecting Process – 접근방법 결정

- ✓ 프로젝트 초기 팀이 필요로 하는 것은 정확한 방향성과 접근방법입니다.
- ✓ 접근방법이 결정된 후, 주요 기술에 대한 조사 및 정리 후 공유하는 과정이 필요합니다.
- ✓ 기술은 관련 표준과 관련 솔루션을 동반하는 경향이 있으므로 너무 깊이 들어가지 않도록 주의합니다.
- ✓ 조직의 아키텍트 그룹에서 [주요 기술 동향 및 분석 보고서]를 정기적으로 발간하면 도움이 됩니다.

[접근방법] CBD가 아닌 CBSE 접근방법

1. 컴포넌트 개요
2. 컴포넌트 정의
3. 컴포넌트 식별
4. 컴포넌트 관계 규칙
5. 컴포넌트 내부 구조
6. 컴포넌트 개발 프로세스

사례참조

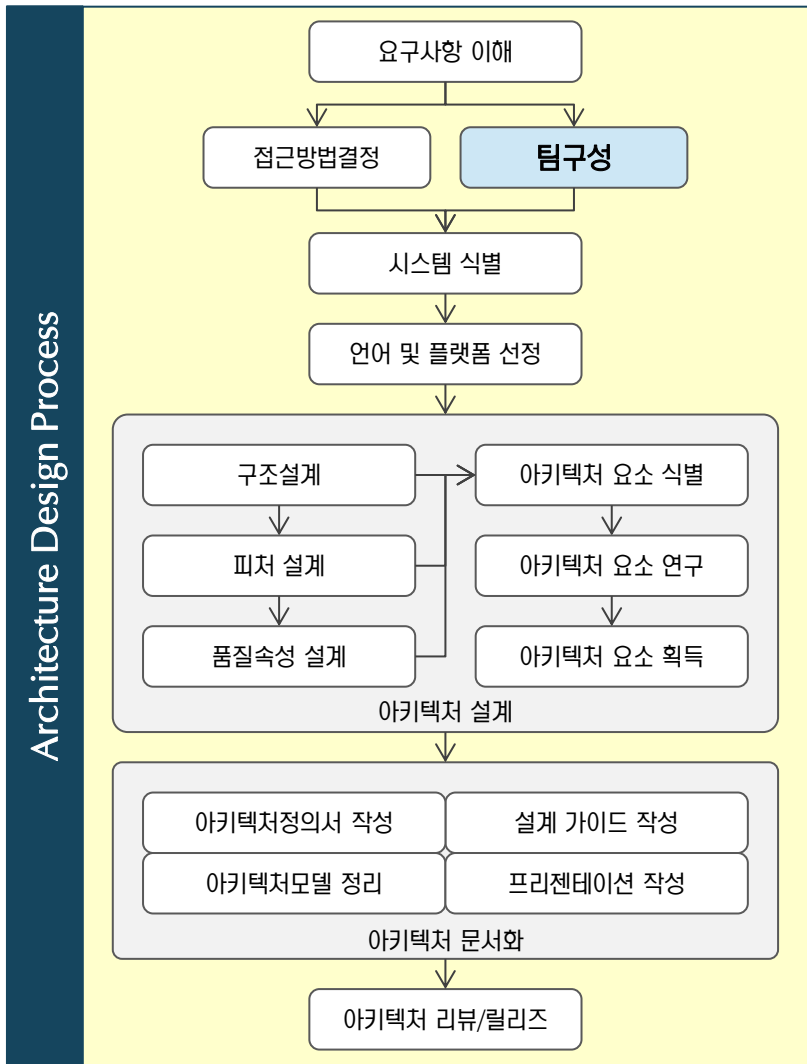
[기술] Identity & Access Management

1. Identity 관리 개념 정의
2. Identity 관리 프로젝트 접근방법
3. 정의 단계
4. Identity 관리
5. Access 관리
6. Identity 관리 이슈사항
7. 시나리오

사례참조

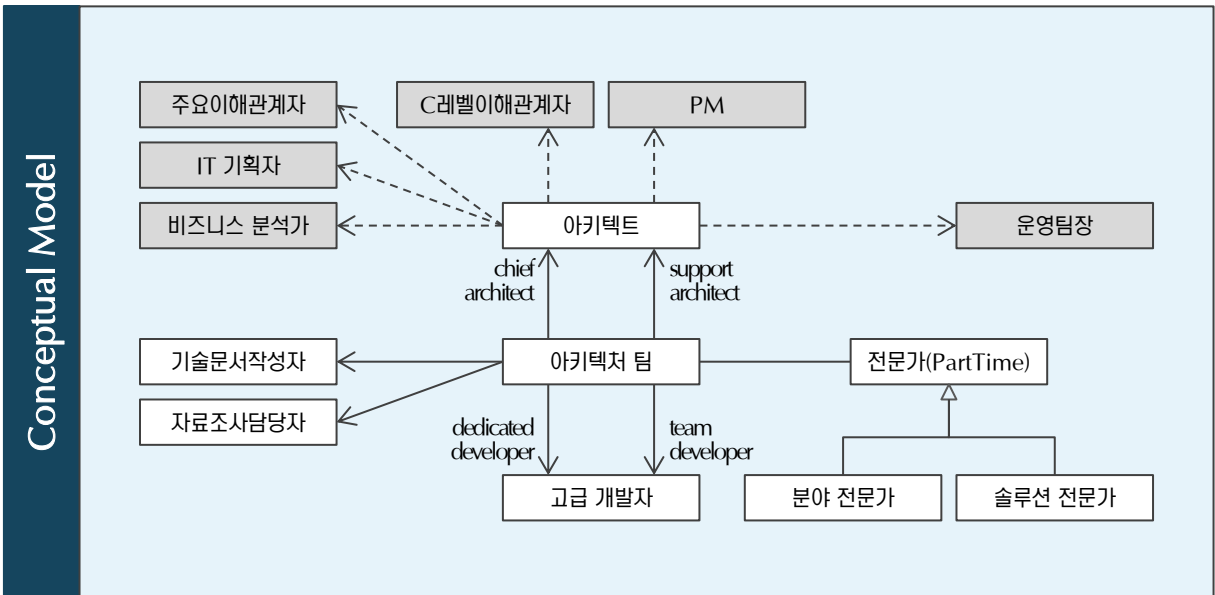
6. New Architecting Process – 팀구성

✓ 소프트웨어 아키텍처 팀을 구성합니다.



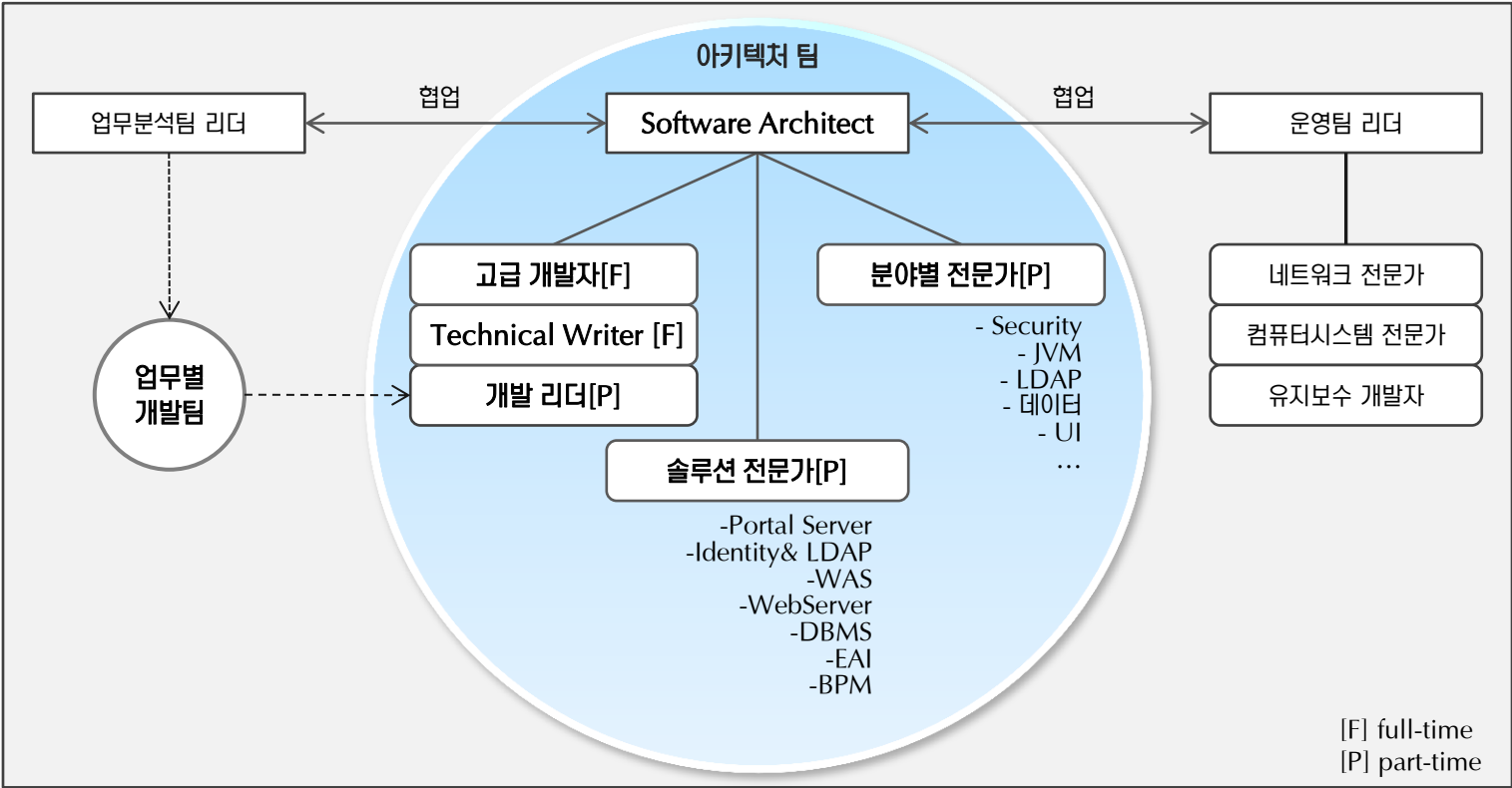
Activity Guide

- ❖ 팀은 개발 역량과 전문화 역량, 솔루션 역량을 중심으로 구성함
- ❖ 기술 문서작성자와 자료 조사 담당자를 통해 문서 수준을 높임
- ❖ 대체로 전문가는 필요한 기간에 파트타임으로 참여를 함
- ❖ 아키텍트는 다양한 역할 담당자, 이해관계자와 의사소통함
- ❖ 아키텍처 팀 규모는 조직의 업무특성과 경험에 의존함
- ❖ 고급개발자 그룹에 각 업무개발팀의 개발리더 참여를 권장함



6. New Architecting Process – 팀구성

- ✓ 아키텍처 팀 구성은 AA, DA, TA, BA 역할로 채워서는 안됩니다.
- ✓ 아키텍처 팀의 핵심은 고급 개발인력과 파트 타임으로 참여하는 솔루션 전문가, 또는 분야별 전문가입니다.
- ✓ 하드웨어, 네트워크 등은 아키텍처 팀의 범주를 넘어섭니다.
- ✓ 프로젝트와 사용 기술 특성에 따라 아키텍처 팀 규모를 결정합니다.



6. New Architecting Process – 팀 구성

- ✓ 어떤 유형의 개발이든 최소 1명의 아키텍트 역할을 지정해야 합니다.
- ✓ 애플리케이션 유형: 독립형 시스템, 엔터프라이즈 시스템(System of System)
- ✓ 개발 유형: 신규 개발, 확장 개발, 솔루션 맞춤(customization)
- ✓ 아키텍처 팀의 활동 범위에 따라 규모가 달라질 수 있습니다. ← 아키텍팅 프로세스 조정에 따라

시스템 유형 / 개발 유형		독립형 시스템		엔터프라이즈 시스템	
		소규모(10명내외)	중규모(50명내외)	중규모(100명내외)	대규모(200명이상)
신규 개발	신기술 (10%내외)	2명	5~7명	12명	20명이상
	기존 기술 (5%내외)	1명	3명	7명	12명이상
확장 개발	신기술 (5%이상)	1명	3명	6명	10명이상
	기존 기술 (5%이내)	1명	2명	4명	8명이상
솔루션 맞춤		1명	1명	2명	4명이상

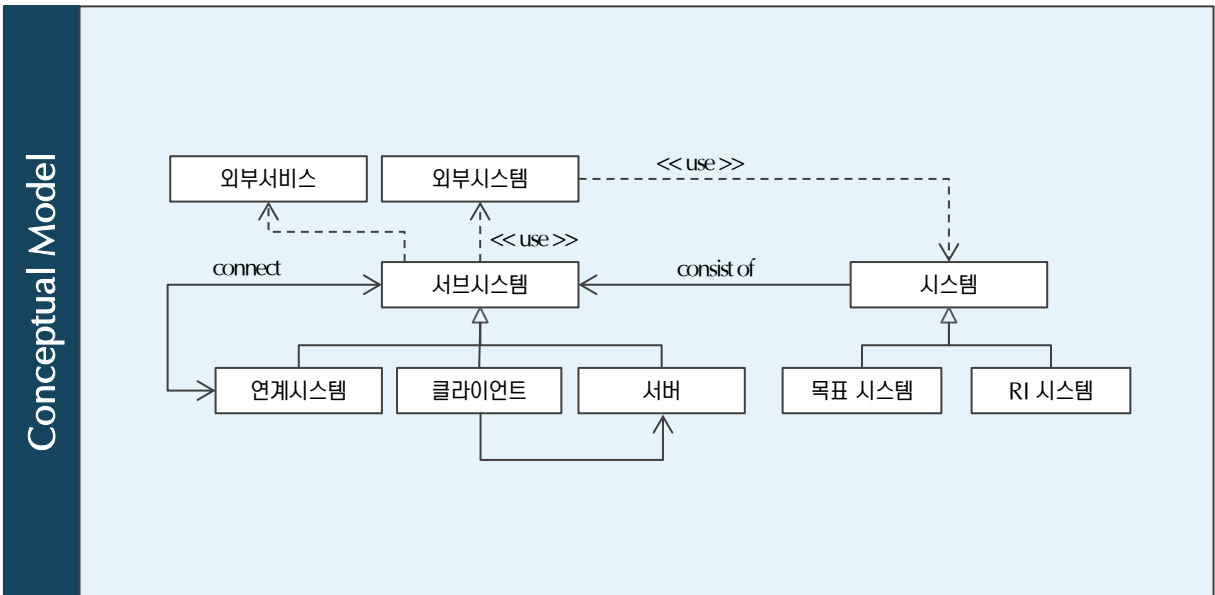
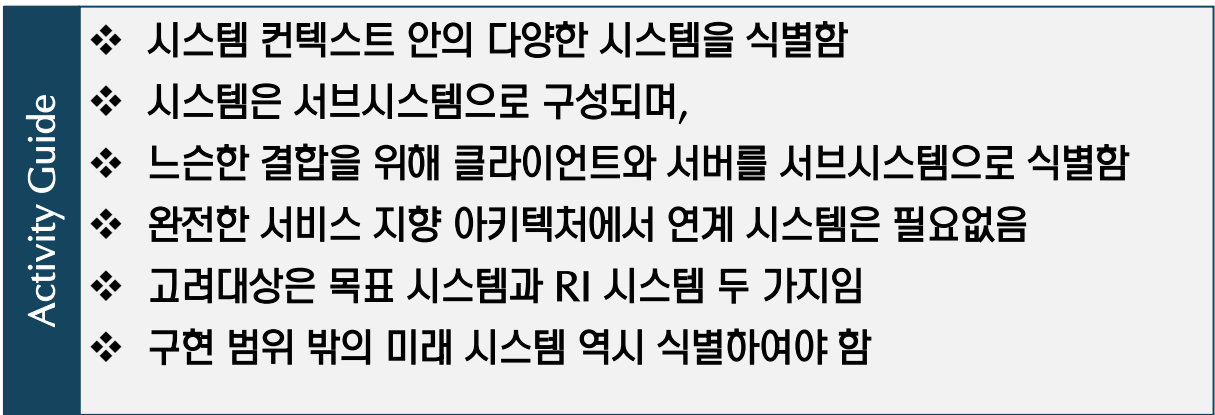
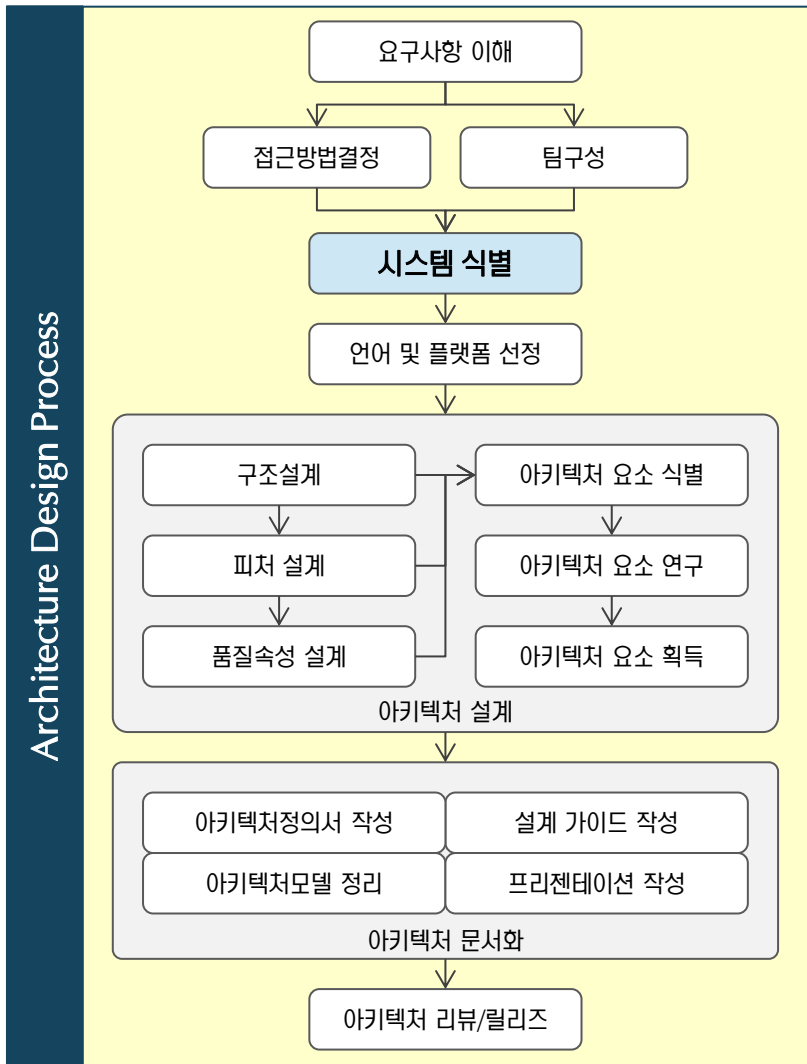
6. New Architecting Process – 개발 팀과 역할

- ✓ 소프트웨어 공학 기준, 표준 역할이 있으며, 개발 관련 프로세스는 역할 이름을 준수하여야 합니다.
- ✓ 조직의 특성과 프로젝트의 상황에 맞추어 역할을 정의합니다.
- ✓ 프로젝트에 사용한 기술에 따라 역할을 재 정의할 필요가 있습니다. , 예, UI integrator, 웹 퍼블리셔, 웹 기획자

RUP의 역할 개수	RUP 역할 이름	A사	B 프로젝트
5 가지 역할: 요구사항	프로젝트 관리자, 프로젝트 리더, 프로세스 엔지니어, 배치 관리자, 요구사항 검토담당자, 아키텍처 검토담당자	프로젝트 관리자	프로젝트 관리자 아키텍트 스크럼 마스터(PL) 서버 개발자 빌드 관리자 UI 개발자
6 가지 역할: 분석 설계	프로젝트 검토담당자, 이해관계자 요구사항 검토담당자	현업 관리자	
3 가지 역할: 구현	시스템 분석가, 요구사항 명세담당자 UI 디자이너, 소프트웨어 아키텍트 설계 검토담당자, 프로세스 엔지니어 도구 전문가, 형상 관리자 변경 관리자	선임 프로그래머	
2 가지 역할: 테스트	설계자, 개발자, 코드 검토담당자, 통합담당자 테스트 디자이너, 테스트담당자	프로그래머	웹 기반 시스템 프로젝트
4 가지 역할: 배치			
2 가지 역할: 형상 및 변경관리			
2 가지 역할: 프로젝트 관리			
3 가지 역할: 개발환경			

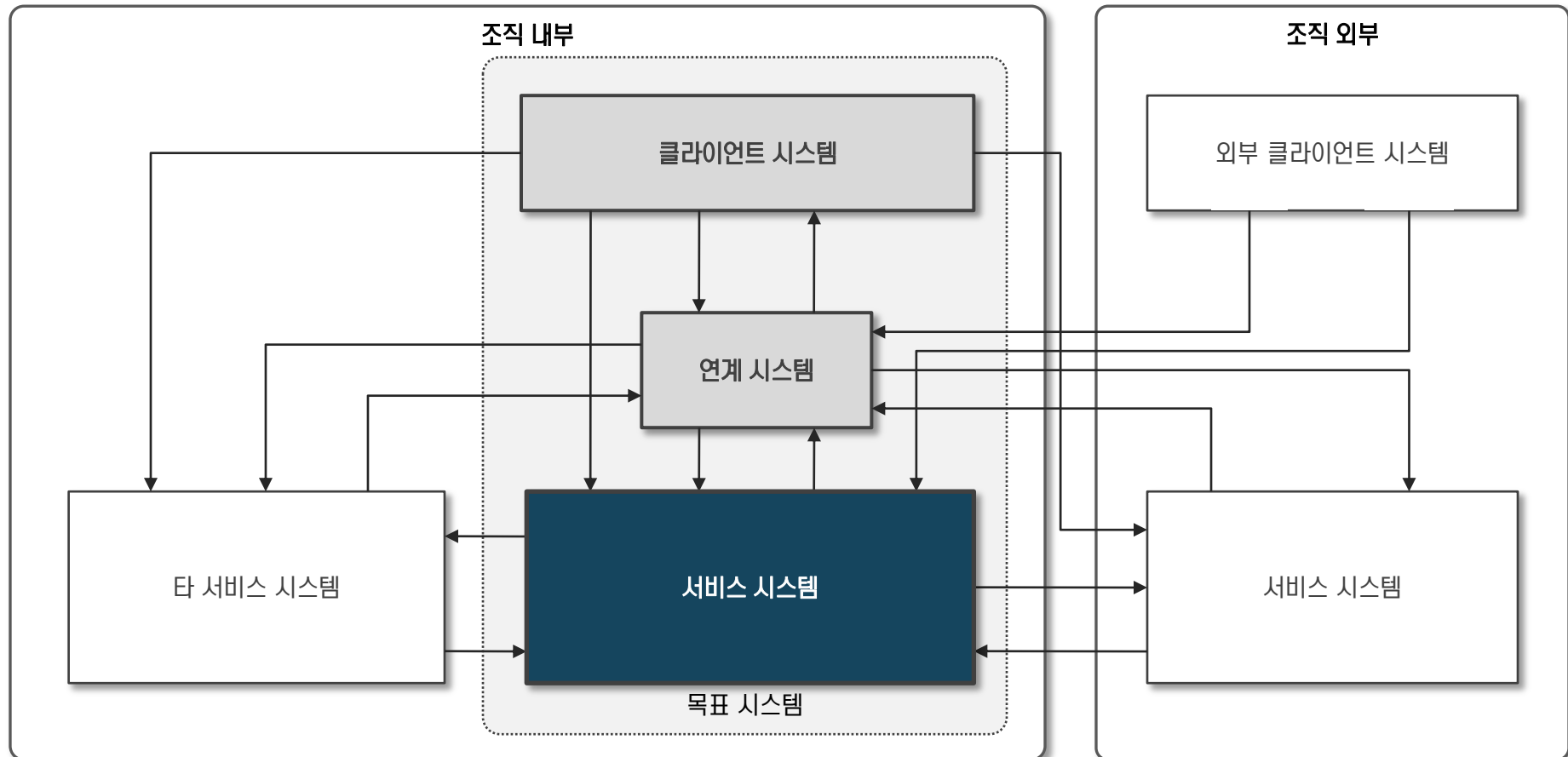
6. New Architecting Process – 시스템 식별

✓ 목표 시스템을 식별합니다.



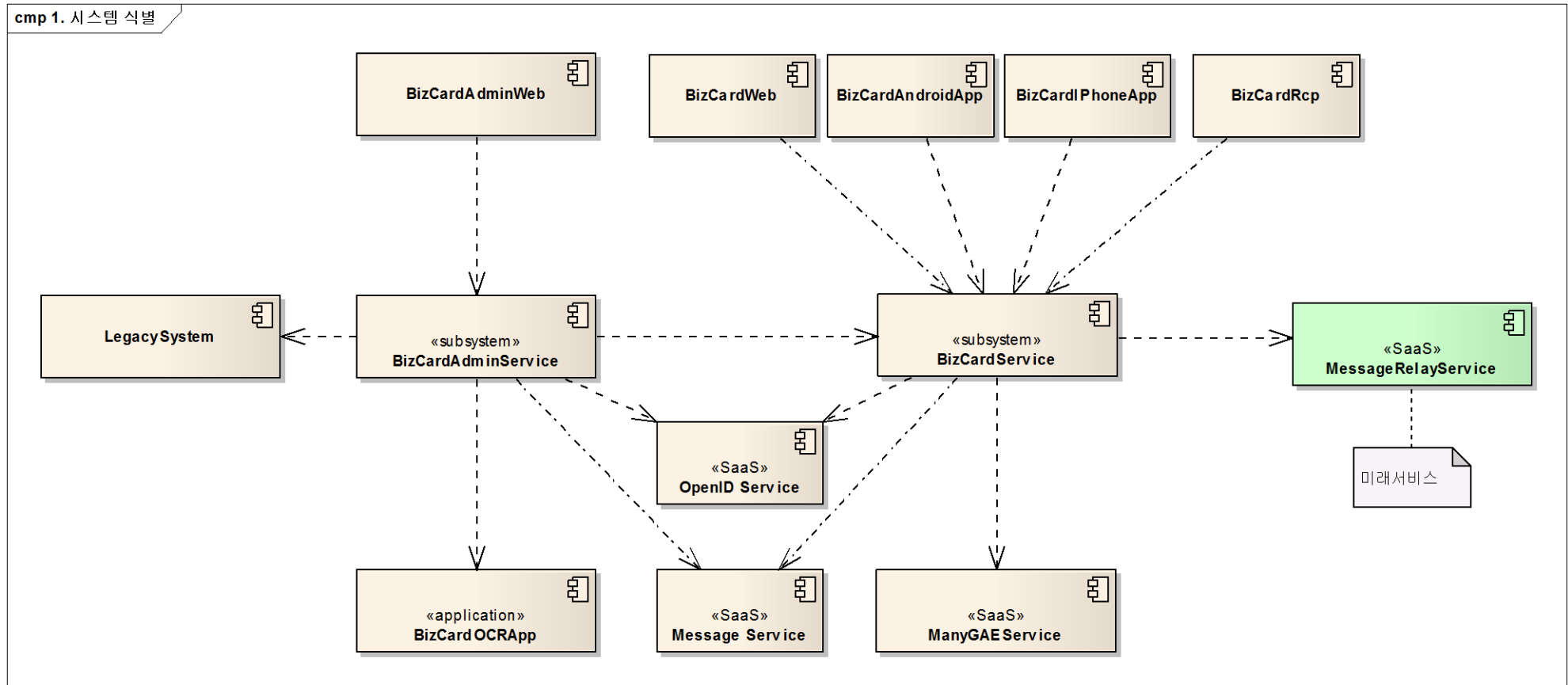
6. New Architecting Process – 시스템 식별

- ✓ 우선 시스템 유형을 정의하고, 정의된 기준에 근거하여 시스템을 식별합니다.
- ✓ 조직 내부의 시스템과 외부의 시스템을 나누어 식별하는 것이 바람직합니다.



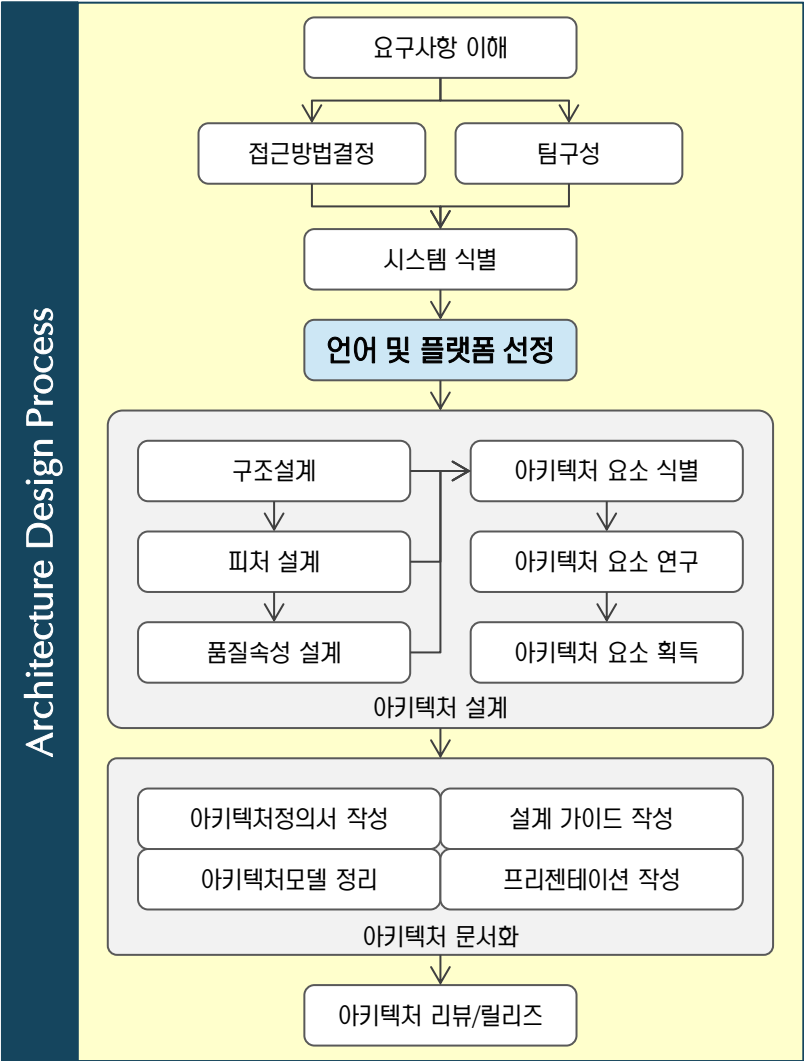
6. New Architecting Process – 시스템 식별

- ✓ 예제, 비즈카드 시스템에서 시스템 식별
- ✓ SEI에서 모듈 분해(Decomposition)와 유사한 작업입니다.



6. New Architecting Process – 언어 및 플랫폼 선정

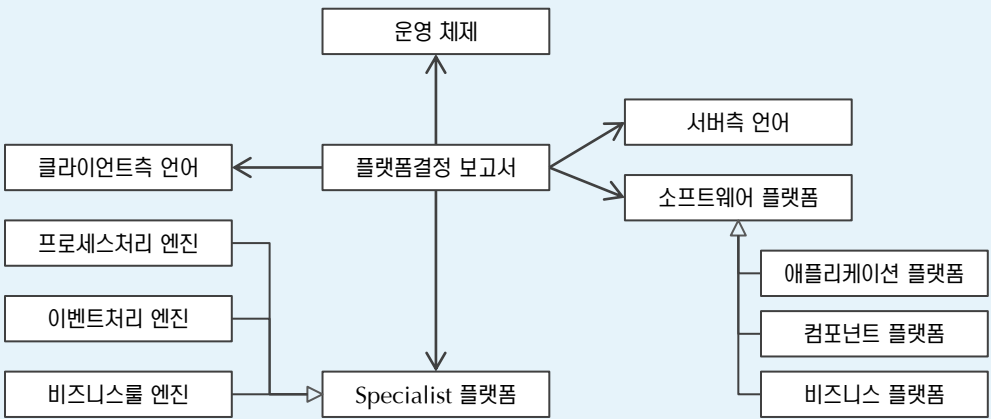
✓ 아키텍처 설계를 위한 프로그램 언어 및 플랫폼을 선정합니다.



Activity Guide

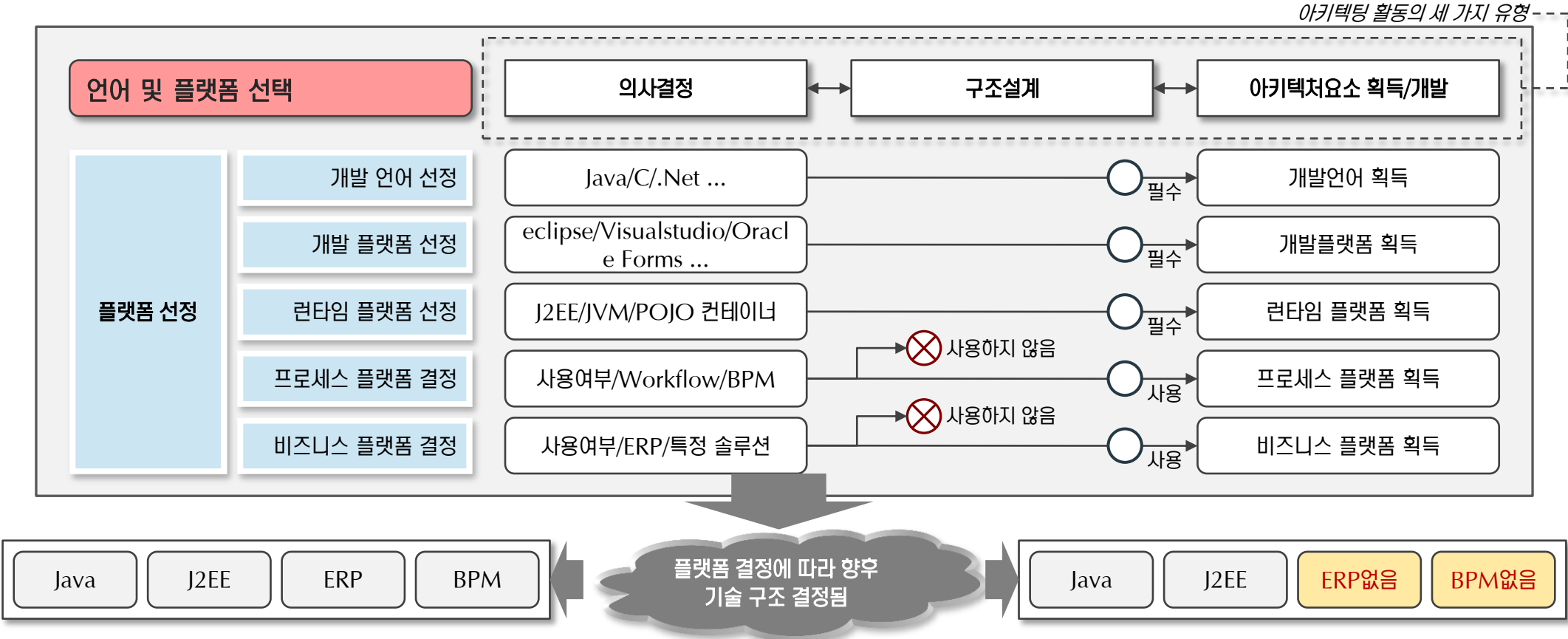
- ❖ 언어는 클라이언트측과 서버측 언어를 별도로 식별함
- ❖ 소프트웨어 플랫폼을 식별함
- ❖ 엔진으로 명명할 수 있는 특수 플랫폼을 식별함

Conceptual Model



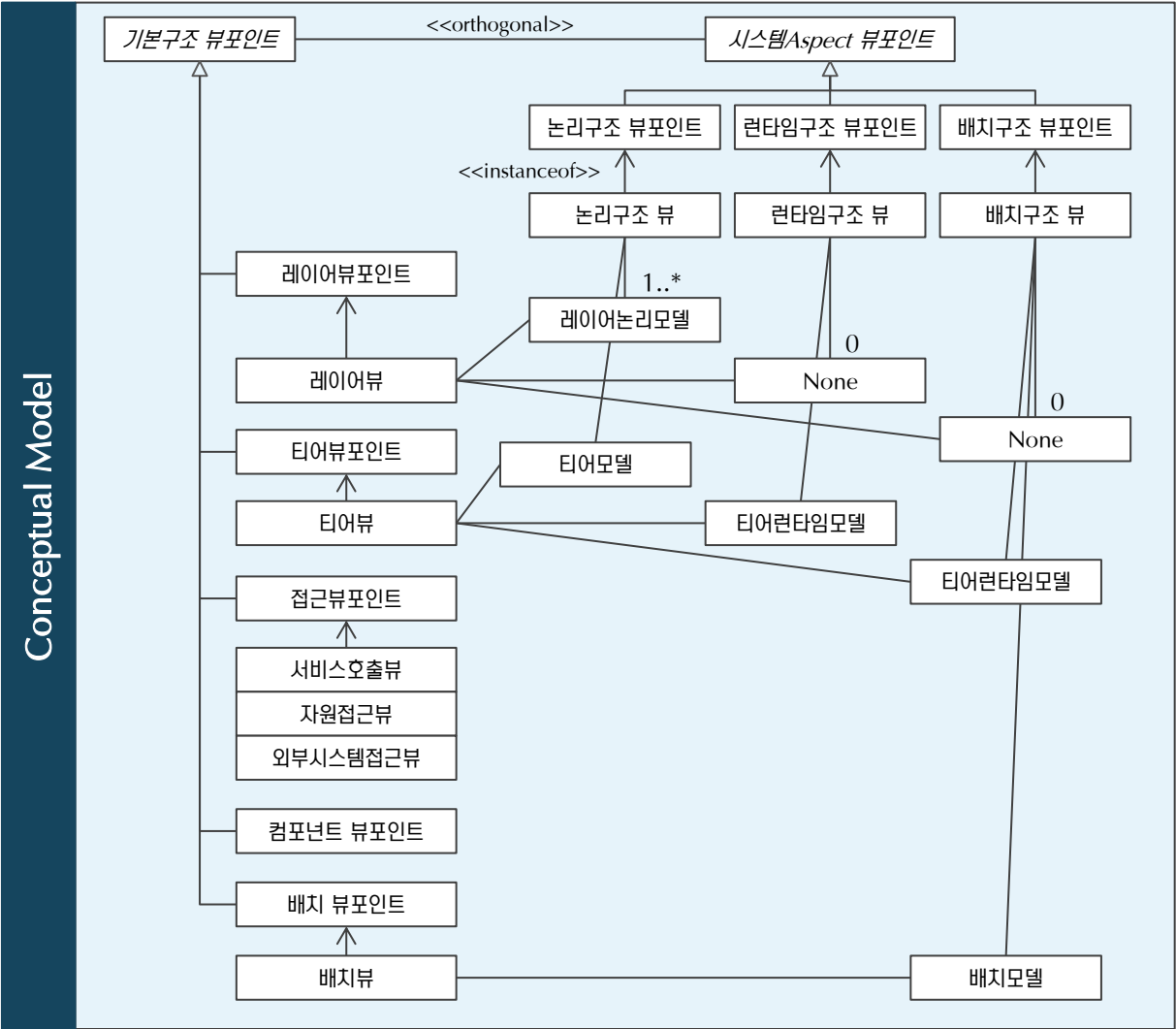
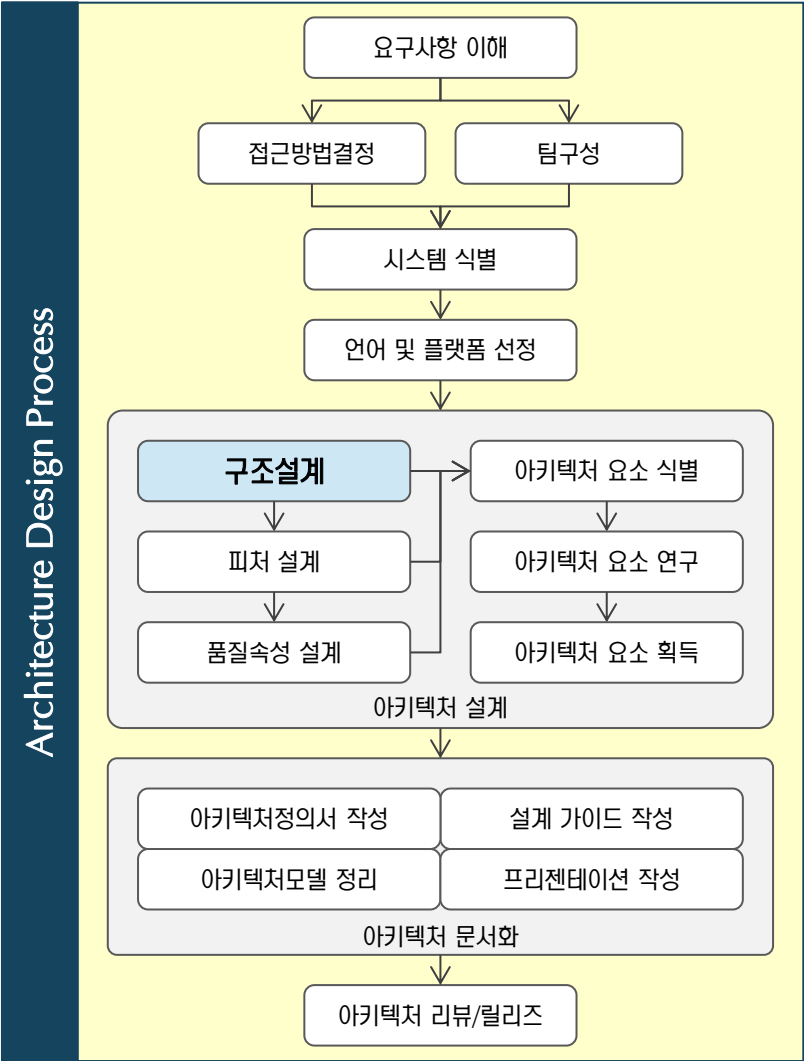
6. New Architecting Process – 언어 및 플랫폼 선정

- ✓ 잘못된 플랫폼 선택은 많은 문제의 근본 원인을 제공하고 있습니다.
 - 개발언어와 런타임 플랫폼은 필수 선택항목이며, 프로세스 플랫폼과 비즈니스 플랫폼은 옵션항목입니다.
 - 선택된 플랫폼은 기술구조, 개발인력, 사용 소프트웨어에 제약조건을 가하므로 매우 신중해야 합니다.



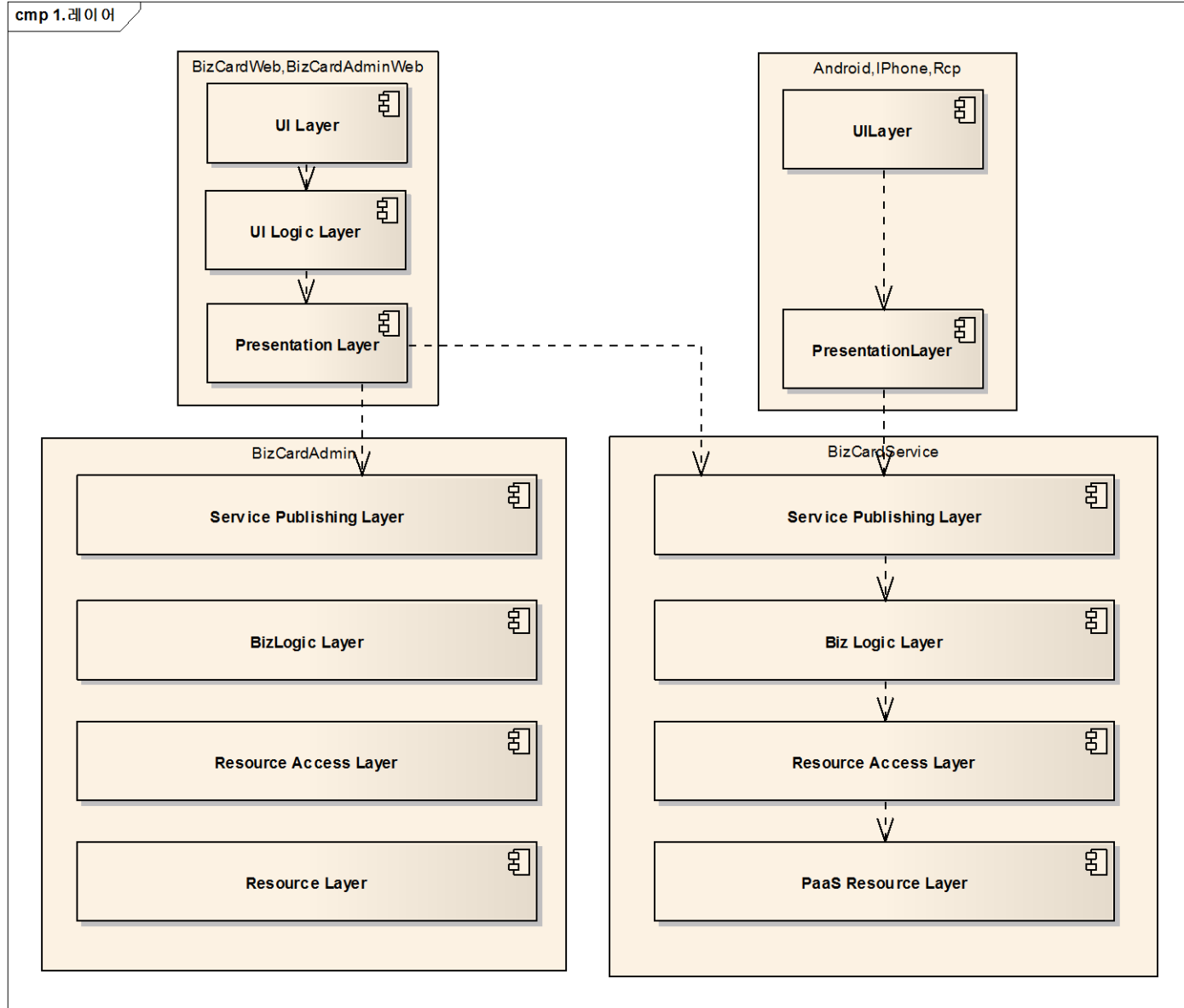
6. New Architecting Process – 구조설계

✓ 구조설계는 품질요구사항을 고려하지 않더라도 기본적으로 설계되어야 함



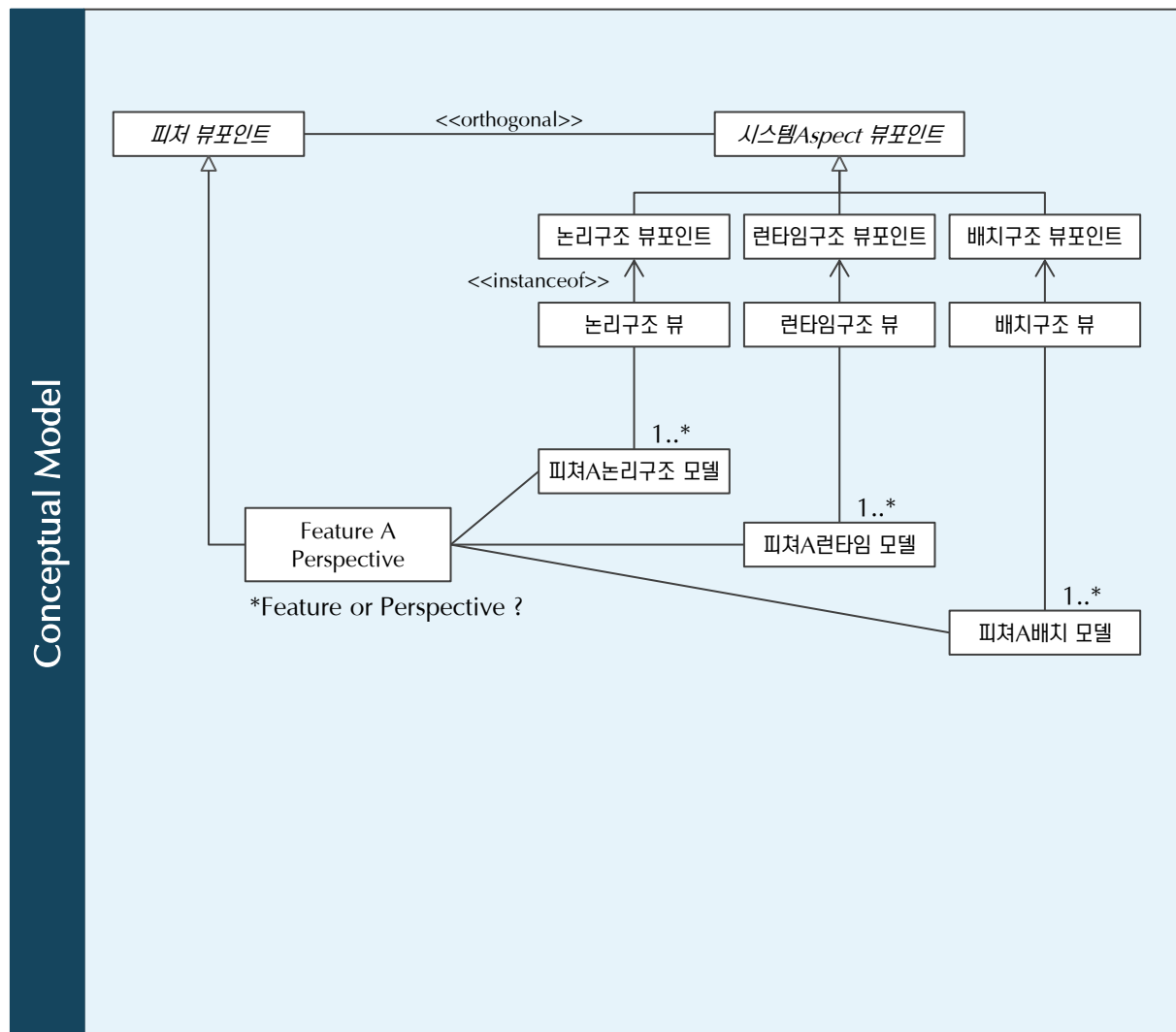
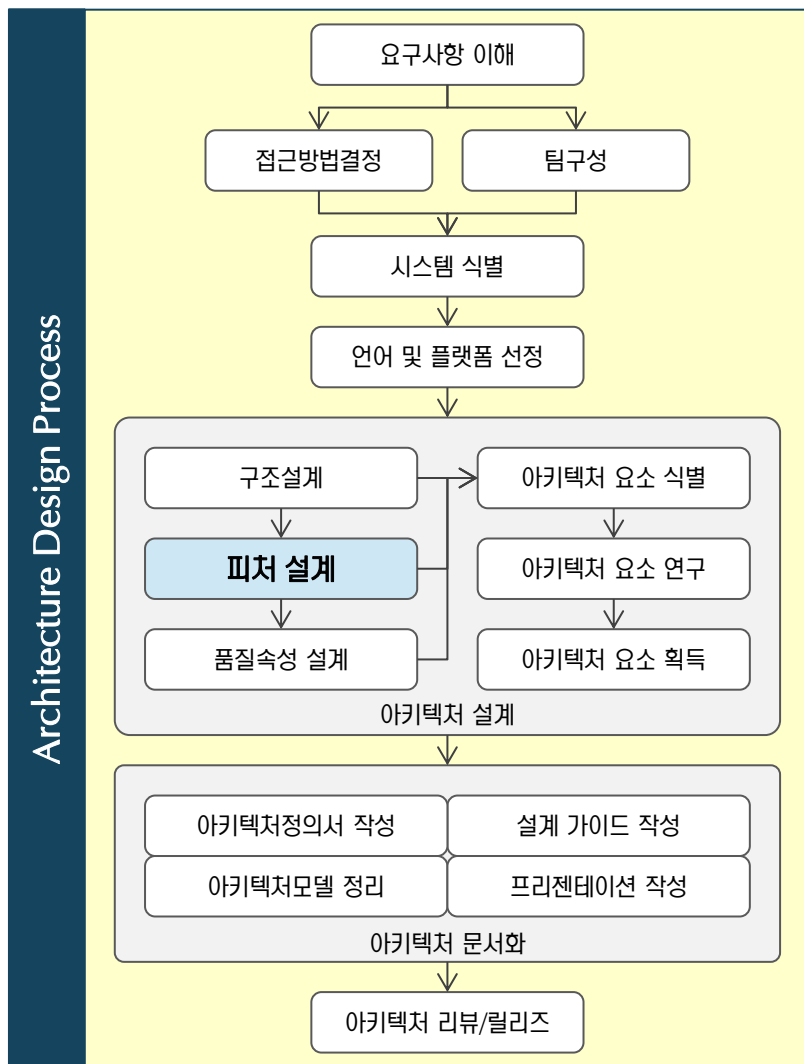
6. New Architecting Process – 구조 설계

✓ 구조설계의 첫 번째는 레이어 설계입니다.



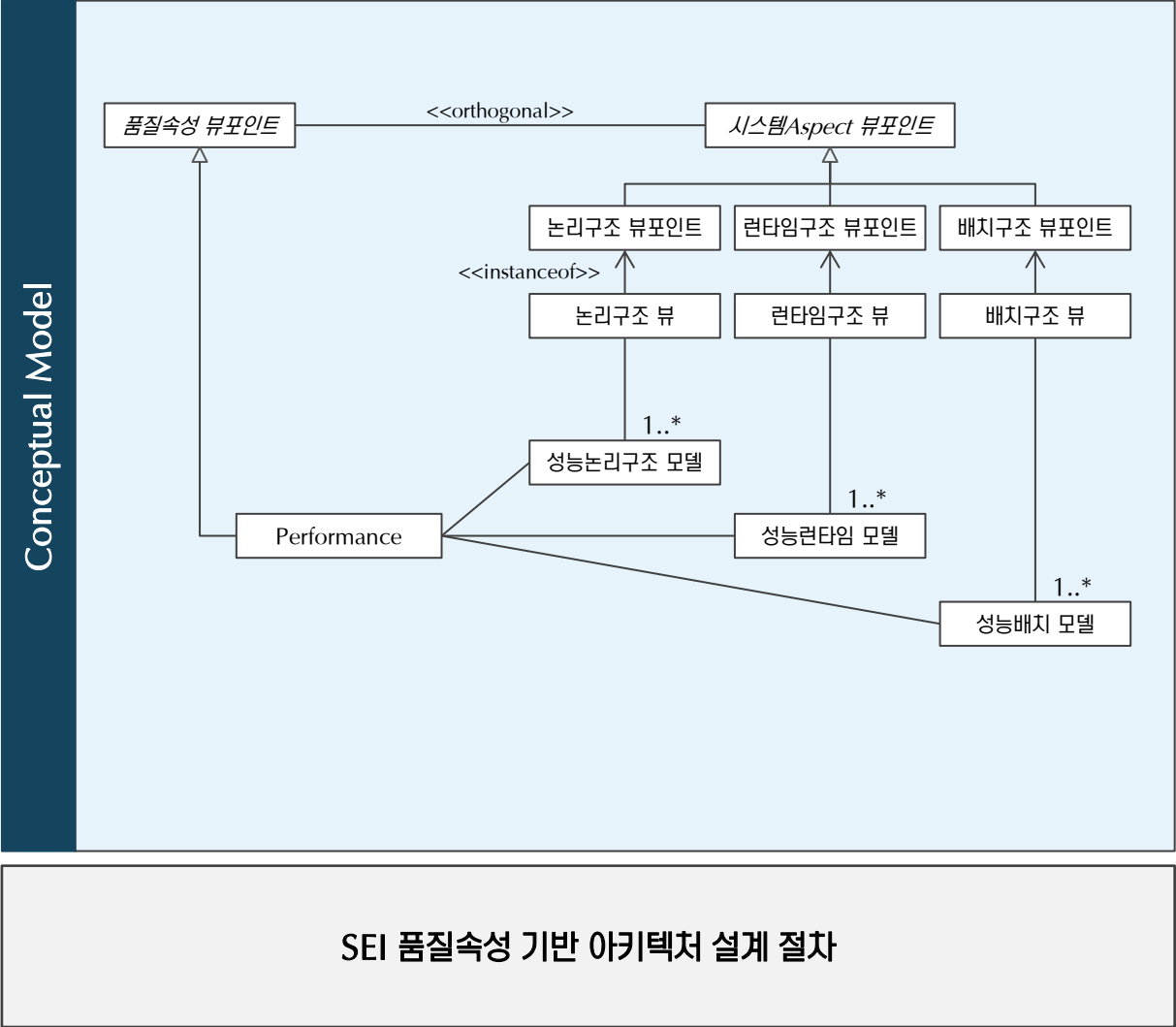
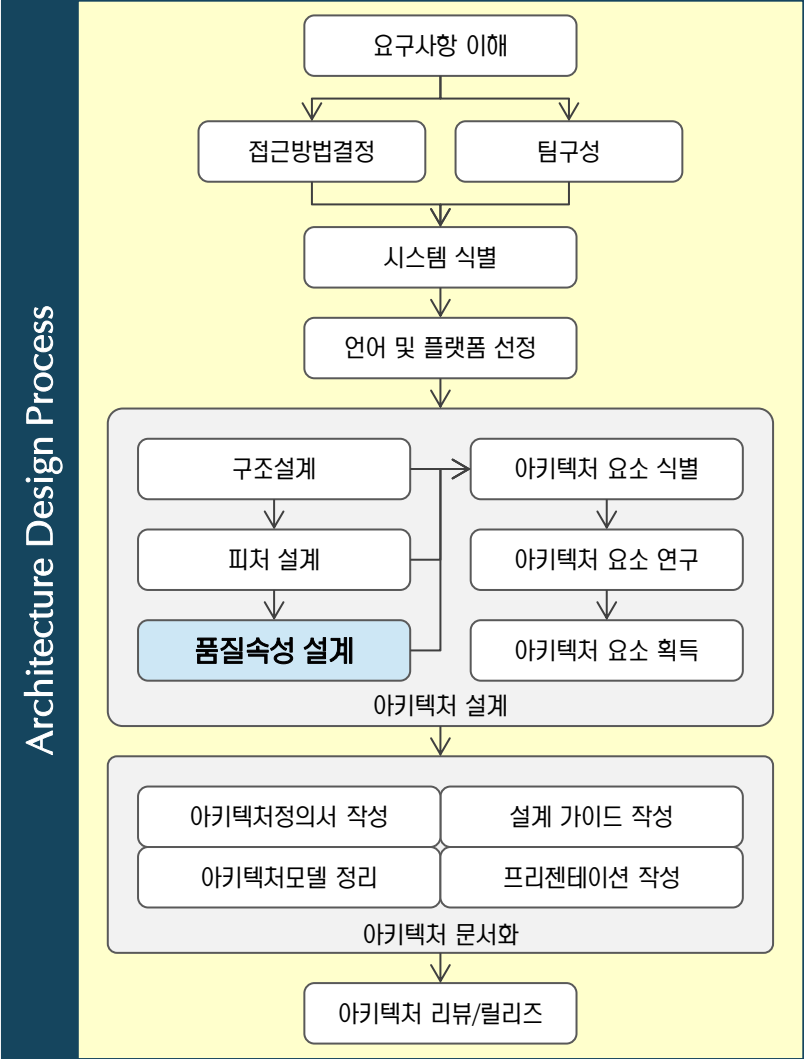
6. New Architecting Process – 피쳐 설계

- ✓ 피처는 기능과 품질이 다양한 비율로 결합된 형태의 요구사항입니다.



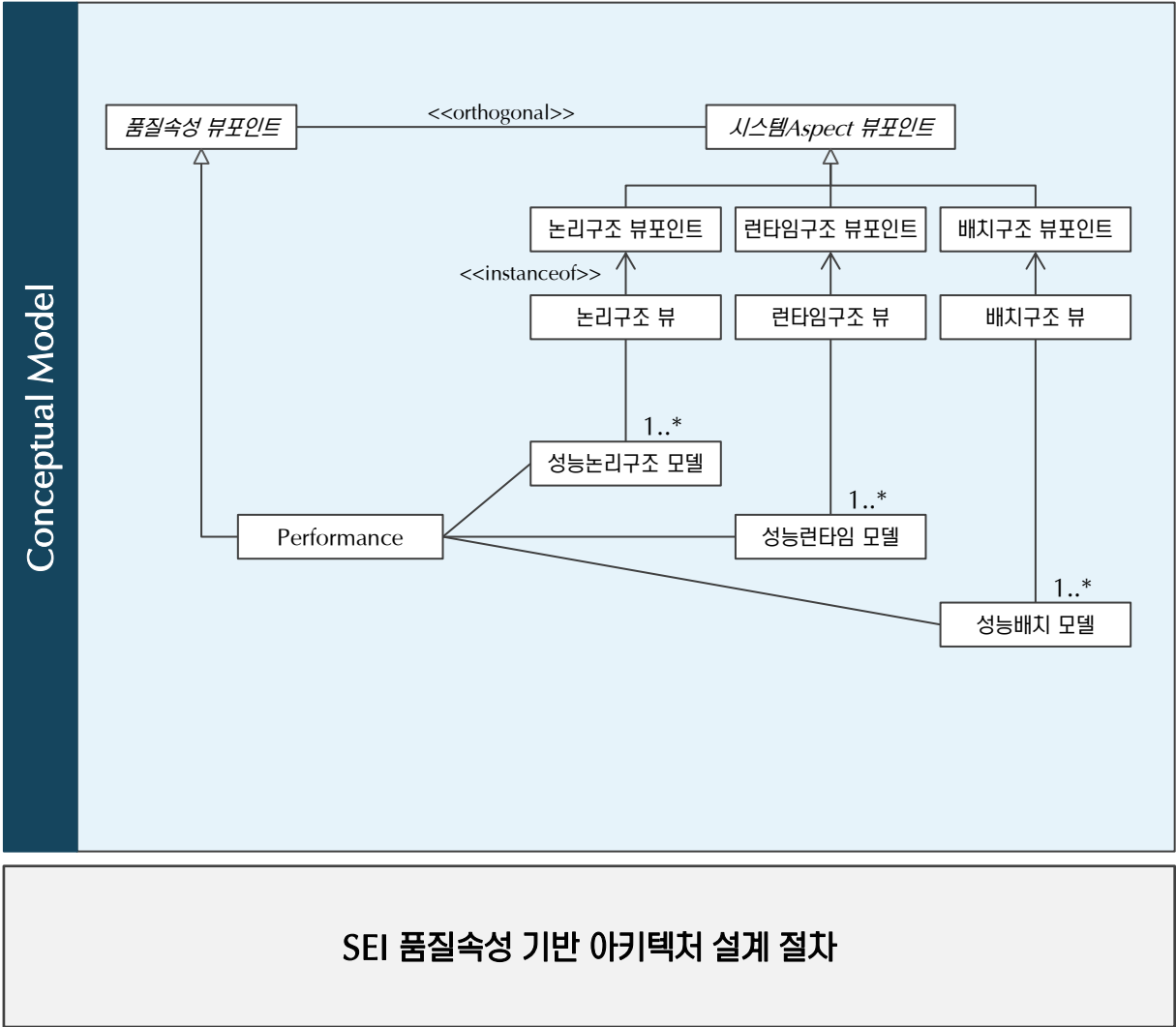
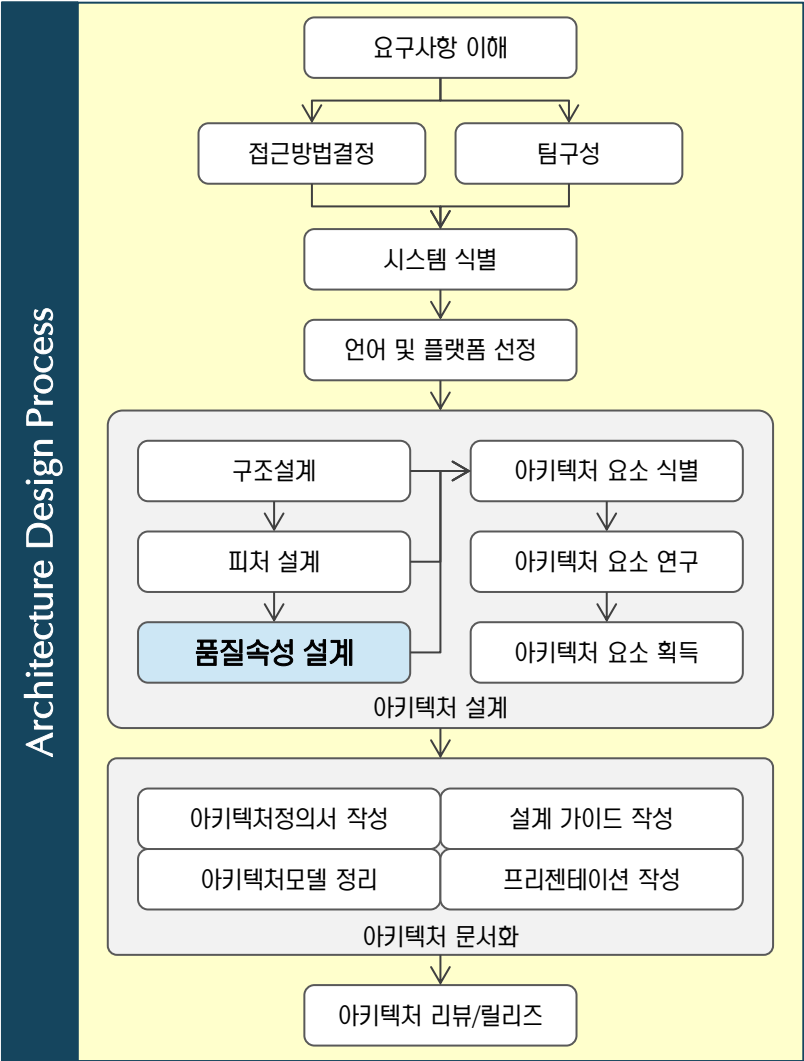
6. New Architecting Process – 품질속성 설계

✓ 품질속성은 조직이나 프로젝트 별로 서로 다른 세트를 선호하는 경향이 있음



6. New Architecting Process – 품질속성 설계

✓ 품질속성은 조직이나 프로젝트 별로 서로 다른 세트를 선호하는 경향이 있음



7. 토의

- ✓ 질의 응답
- ✓ 토론

감사합니다...

- ❖ 넥스트리컨설팅(주)
- ❖ CEO 송태국 / 대표 컨설턴트
- ❖ tsong@nextree.co.kr