

제131회 정보관리기술사 해설집

2023.08.26

국가기술자격 기술사 시험문제

기술사 제 131 회

제 2 교시 (시험시간: 100 분)

분야	정보통신	자격 종목	정보관리기술사	수검 번호	성 명
----	------	----------	---------	----------	--------

※ 다음 문제 중 4 문제를 선택하여 설명하시오. (각 10 점)

1. ISP (Information Strategy Planning) 과 BPR (Business Process Reengineering) 의 개념과 수행절차를 비교 설명하고, 기업에서 이 두가지가 상화 보완적으로 활용하기 위한 방안을 설명하시오
2. 데이터 시각화(Data Visualization)와 관련하여 다음을 설명하시오.
 - 가. 데이터 시각화의 개요
 - 나. 데이터 시각화의 원리 및 절차
 - 다. 데이터 시각화 유형
 - 라. 효과적인 데이터 시각화를 위한 효율화 방안
3. 인공지능의 개발 및 적용과정에서 윤리적으로 다루어져야 할 주요 내용과 인공지능을 효과적으로 관리하고 규제하기 위한 거버넌스 모형에 대하여 설명하시오
4. 제로 트러스트 보안(Zero Trust Security)모델의 보안원리, 핵심원칙, 적용분야를 트러스트 보안(Trust Security) 모델과 비교하여 설명하시오.
5. 소켓(Socket) 통신과 관련하여 다음을 설명하시오.
 - 가. 소켓 통신 정의
 - 나. 소켓 통신 방식 개념도 및 유형
 - 다. TCP 소켓 및 Web 소켓 흐름 설명 라. 소켓 통신 방식과 HTTP 통신 방식 비교

6. 아키텍처 스타일과 디자인 패턴에 대하여 다음을 설명하시오.

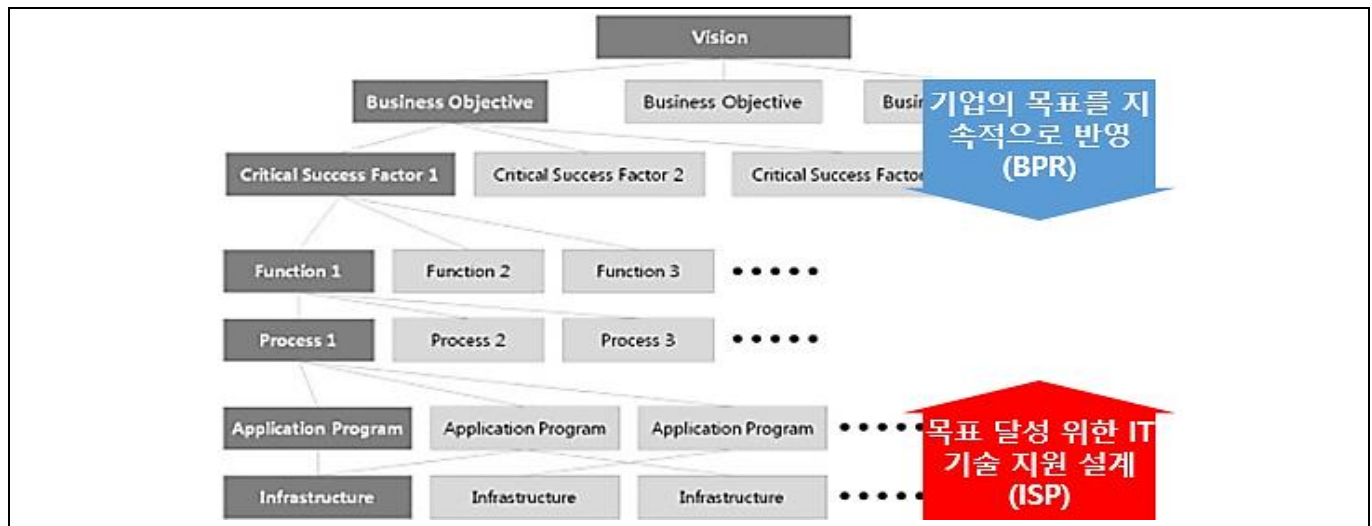
가. 아키텍처 스타일과 디자인 패턴의 차이점

나. 대표적인 아키텍처 스타일 3 가지

다. GoF(Gang of Four) 디자인 패턴의 유형을 구분하고, 유형별 대표적인 디자인 패턴
설명

01	ISP(Information Strategy Planning)과 BPR(Business Process Reengineering)		
문제	ISP(Information Strategy Planning)과 BPR(Business Process Reengineering)의 개념과 수행절차를 비교 설명하고, 기업에서 이 두가지가 상화 보완적으로 활용하기 위한 방안을 설명하시오		
도메인	경영전략	난이도	중(상/중/하)
키워드	환경분석, 환경분석, 목표모델 설계, 이행계획 수립		
출제배경	최근 기업과 공공에서 정보시스템 계획 전략 수립 시 BPR과 함께 진행하는 추세임		
참고문헌	ITPE 서브노트 https://m.blog.naver.com/Strategy&Consulting		
해설자	강남평일야간반 전일 기술사(제 114회 정보관리기술사 / nikki6@hanmail.net)		

I. 정보전략 컨설팅 방법론, ISP/BPR 의 최근 추이 분석



- 기업의 비전 달성을 위해 다양한 사업 목표가 있고 그것에 영향을 주는 CSF가 존재하므로 기능과 프로세스들은 정보시스템과 상호 연관되어 설계되어야 함
- 그러다 보니 최근 프로세스 컨설팅의 전통적인 영역인 BPR과 ISP가 하나의 컨설팅으로 합쳐지는 추세

II. ISP(Information Strategy Planning)과 BPR(Business Process Reengineering)의 개념 비교

ISP	BPR
<div> <div>정보시스템 우선순위</div> <div>구축 비용 산정</div> <div>관리체계의 최적화</div> </div> <div>→</div> <div>성공적인 정보시스템, 정보관리 체계</div>	<div> <div>정보시스템</div> <div>현재 프로세스 → 혁신적 재구축 → 혁신 프로세스</div> <div>시간 최소화</div> <div>비용 최소화</div> <div>품질향상</div> <div>서비스 향상</div> <div>고객만족 극대화</div> </div>
<ul style="list-style-type: none"> - 조직의 중장기 마스터 플랜을 지원하기 위한 정보시스템을 계획하고 전략을 수립하는 활동 - 정보화 사업 기획 단계의 내실화, 재정투자관리 효율화 방안 	<ul style="list-style-type: none"> - 비용, 품질, 서비스, 속도와 같은 핵심적인 경영성과 지표들의 비약적인 향상을 이룩하기 위해 사업활동 (Business Process)을 근본적으로 재 고려하고 핵심적인 업무를 급진적으로 재 설계

II. ISP와 BPR의 수행 절차 비교

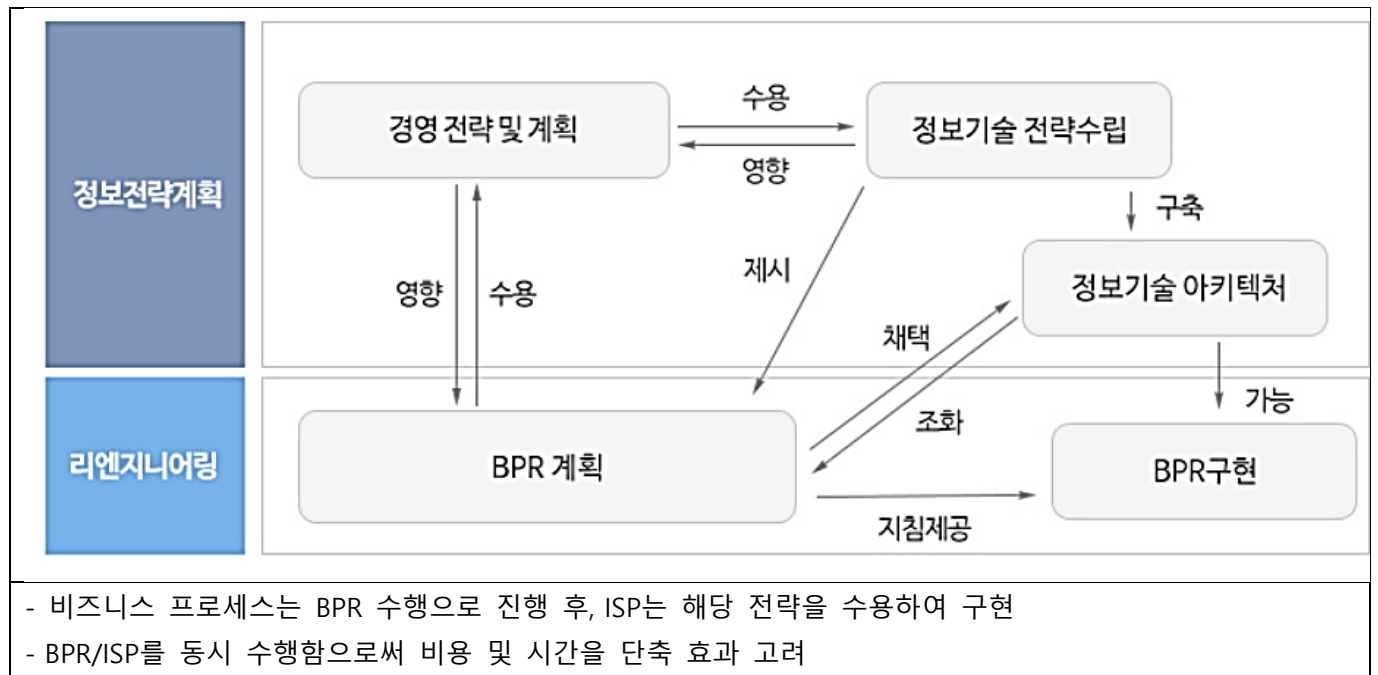
가. ISP와 BPR의 수행 절차 비교

ISP	BPR
<div>Phase 1 수행 기획</div> <div>Phase 2 현행 모델 분석(As-Is 단계)</div> <div>Phase 3 목표 모델 분석(To-Be 단계)</div> <div>Phase 4 통합 이행계획</div>	<div>Phase 1 환경 분석</div> <div>Phase 2 현황 분석</div> <div>Phase 3 목표모델 설계</div> <div>Phase 4 통합 이행계획</div>

나. ISP와 BPR의 수행 절차 상세 비교

ISP	BPR
<div>I. 환경 분석</div> <ul style="list-style-type: none"> - 내·외부 환경분석 - 법령·제도 분석 - 정보기술(IT) 환경분석 - 핵심성공요소(CSF) 도출 <div>II. 현황분석 (As-Is 분석)</div> <ul style="list-style-type: none"> - 업무현황 분석 - 정보기술(IT) 현황 분석 - 선진사례 분석 - 이해관계자 분석 - 요구사항 도출 - 이슈통합 및 개선과제 도출 <div>III. 목표모델 설계(To-Be Model)</div> <ul style="list-style-type: none"> - 정보화 비전 및 전략 수립 - 개선 과제 상세화 - 업무 프로세스 설계 - 정보시스템 구조 설계 - 데이터 구조 설계 - 기술 및 보안 구조 설계 - 연계 구조 설계 - IT거버넌스 설계 - 법령·제도 개선사항 도출 <div>IV. 통합 이행계획</div> <ul style="list-style-type: none"> - 이행과제 확정 - 통합 이행계획 수립 - 총 구축비 산출 - 효과 분석 - 제안요청서 작성 	<div>I. 수행 기획</div> <ul style="list-style-type: none"> - 수행계획 수립 - 수행활동 정의 및 조직 구성 - 수행계획 검토 및 승인 <div>II. 현행 모델 분석 (As-Is 단계)</div> <ul style="list-style-type: none"> - 문제 분석 - 핵심 프로세스 정의 - 문제대안 도출 <div>III. 개선 모델 개발 (To-Be 단계)</div> <ul style="list-style-type: none"> - 대안 도출 - 최적 대안의 설정 <div>IV. 통합 이행</div> <ul style="list-style-type: none"> - 대안 실행 - 대안 성과 평가

II. 기업에서 이 두가지가 상화 보완적으로 활용하기 위한 방안

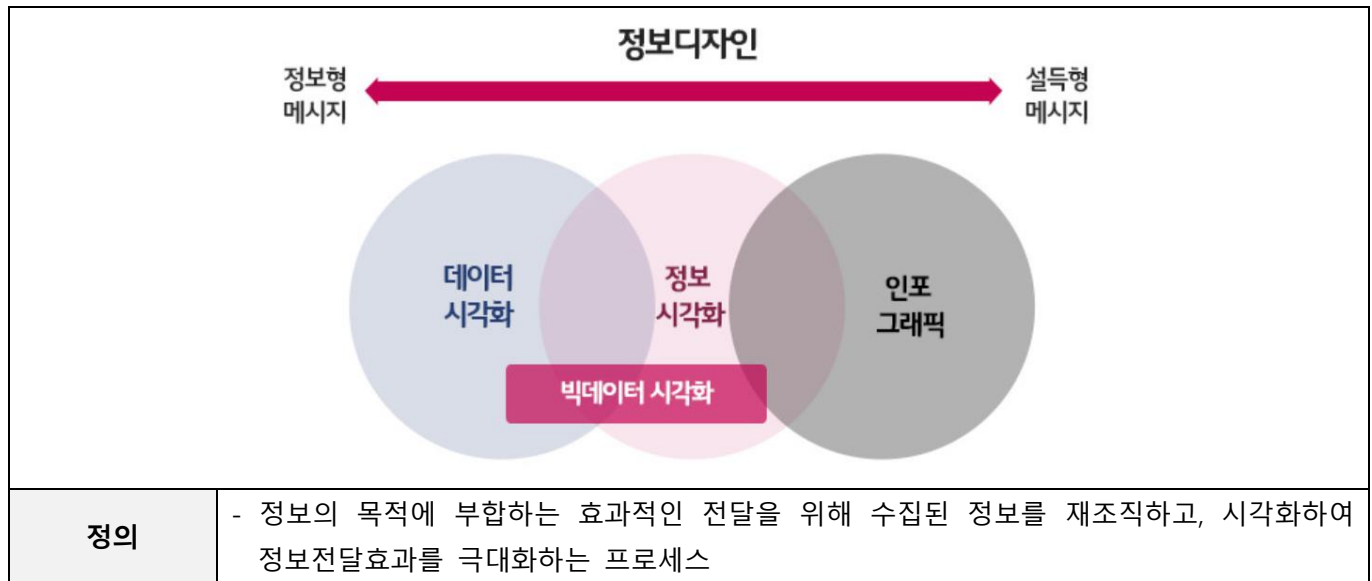


- 컨설팅 수행 시간 및 업무 특성을 고려하여 BPR 및 ISP 방법론의 수정(Tailoring) 적용 필요

“끝”

02	데이터 시각화(Data Visualization)		
문제	<p>데이터 시각화(Data Visualization)와 관련하여 다음을 설명하십시오.</p> <p>가. 데이터 시각화의 개요</p> <p>나. 데이터 시각화의 원리 및 절차</p> <p>다. 데이터 시각화 유형</p> <p>라. 효과적인 데이터 시각화를 위한 효율화 방안</p>		
도메인	데이터베이스	난이도	중(상/중/하)
키워드	<p>정보 구조화, 정보 시각화, 정보 시각화 표현</p> <p>시간 시각화, 분포 시각화, 관계 시각화, 비교 시각화, 공간 시각화</p>		
출제배경	빅데이터 분석 중 미 출제 영역인 데이터 시각화에 대한 이해 확인		
참고문헌	<p>ITPE 기술사회 자료</p> <p>https://kjwan4435.tistory.com/18</p>		
해설자	정상반멘토 이상헌 기술사(제 118회 정보관리기술사 / bluesanta97@naver.com)		

I. 데이터 분석의 마지막, 데이터 시각화의 개요



- 데이터 분석 결과를 한눈에 알아볼 수 있도록 데이터 시각화 원리 및 절차를 준수해야 함.

II. 데이터 시각화의 원리 및 절차

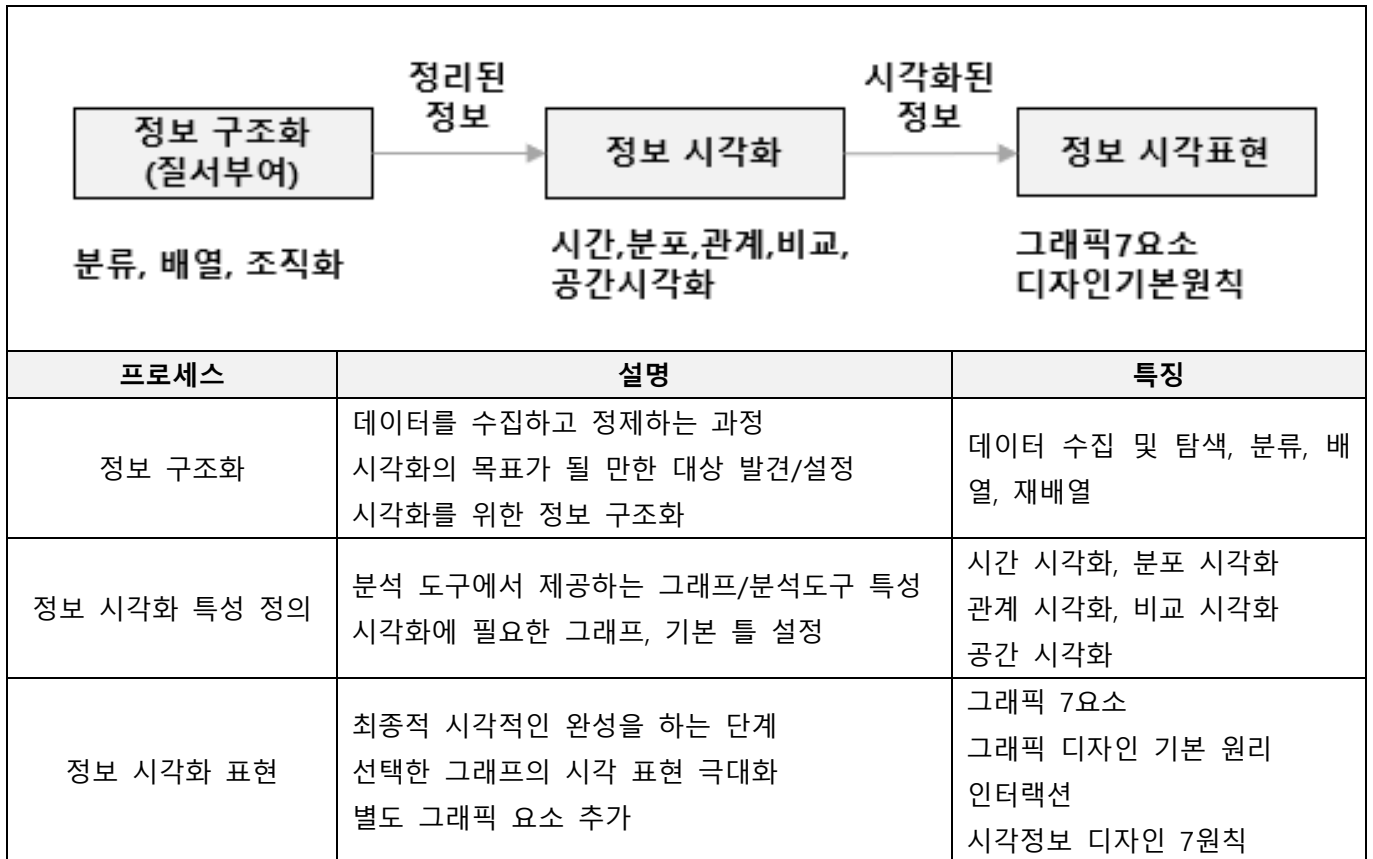
가. 데이터 시각화의 원리

원리	내용
Abstraction - Figuration	- 실제의 물체와 같은 것들일 수록 형상에 가깝고, 차트나 박스와 같이 단순화한 표현들은 추상에 가까움
Functionality - Decoration	- 장식에 집중하기보다 그 데이터자체의 표현에 집중하였는가, 꾸미는데 더 집중하였는가를 표현
Density - Lightness	- 자세히 살펴봄으로써 많은 데이터들을 얻을 수 있는지, 혹은 한눈에 보기에 쉽게 데이터들을 파악할 수 있는지를 확인

Multidimensional - Unidimensional	- 다른 다양한 종류의 데이터들을 표현하였는지, 혹은 한 두개의 데이터들을 표현하였는지를 나타냄
Originality - Familiarity	- 데이터 표현방식에 있어서 독창적으로 표현하였는지, 친숙하게(막대나 꺾은 선) 표현하였는지 나타냄
Novelty - Redundancy	- 데이터를 표현하는 데 있어서, 한 번만 표현할 것인지 아니면 여러 번 반복적으로 표현할 것인지를 나타냄

- 데이터 시각화 원리에 맞게 원하는 방향성을 설정하고 절차에 따라 시각화 작업을 수행함.

나. 데이터 시각화의 절차




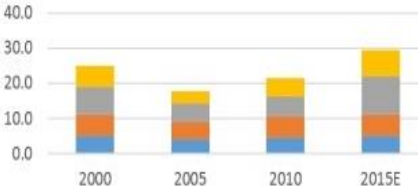
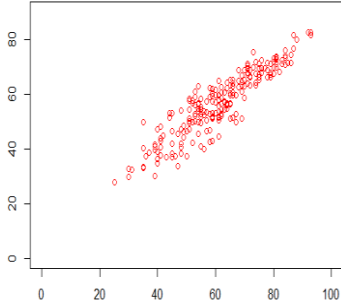
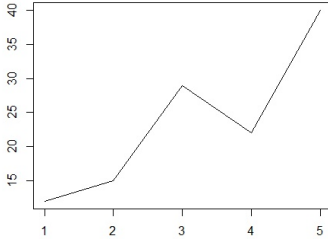
- 데이터 시각화는 다양한 형태의 유형이 존재하며 효과적으로 표현하기 위해 여러 기법을 활용할 수 있음

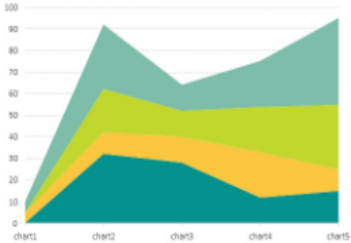
III. 데이터 시각화 유형

가. 데이터 시각화 유형

유형	설명	주요 시각화 방법
시간 시각화	- 시간에 따른 데이터의 변화 또는 트렌드 분석 목적	막대 그래프, 누적 막대 그래프, 점 그래프
분포 시각화	- 최대/최소/평균을 기준으로 각 항목 간의 비교 검토	파이 차트, 도우넛 차트, 트리맵, 누적 연속 그래프
관계 시각화	- 각 변수의 상관관계를 통해 하나의 변화를 통해 다른 변수의 변화를 추정하는 목적	스캐터플롯, 버블차트, 히스토그램
비교 시각화	- 여러 변수를 통해 원하는 데이터를 찾아 가는 목적	히트맵, 스타 차트, 평행 좌표계, 다차원 척도법
공간 시각화	- 지역별 특성 또는 현황을 나타내는 형태로 활용	지도 맵핑

나. 데이터 시각화 주요 기법

구분	개념도	설명
막대 그래프		<ul style="list-style-type: none"> - 통계를 낼 때 사물의 양을 막대 모양의 길이로 나타낸 그래프 - 막대를 세로로 할 수도 있고 가로로 할 수도 있으며, 가독성 면에선 항목이 적을수록 가로가 좋고 항목이 많을수록 세로가 좋음
누적 막대 그래프		<ul style="list-style-type: none"> - 일반적으로 차원별로 측정값의 비중을 보여주기 위한 목적으로 사용하는 막대 그래프의 한 유형 - 단순 누적 막대 그래프(Simple Stacked Bar Chart)와 100기준 누적 막대 그래프(100% Stacked Bar Chart)로 분류
산점도		<ul style="list-style-type: none"> - 직교 좌표계(도표)를 이용해 좌표상의 점들을 표시함으로써 두 개 변수 간의 관계를 나타내는 방법 - 도표 위에 두 변수 X와 Y값이 만나는 지점을 표시 - 두 변수 사이의 관계를 알 수 있는 그래프
선 그래프		<ul style="list-style-type: none"> - 수량을 점으로 표시하고 그 점들을 선분으로 이어 그린 그래프 - 시간에 따라 지속적으로 변화하는 것을 기록할 때 유용 - 조사하지 않은 중간의 값도 대략 예측할 수 있다는 장점이 있는 그래프

영역 차트		<ul style="list-style-type: none"> - 선 그래프와 막대형 그래프를 결합하여 시간 경과에 따른 수량 변화를 표시하는 그래프 - 시간에 따른 변화 규모를 강조하는 경우 유용 - 데이터 값의 비율을 알고 싶을 때 사용
-------	---	---

- 이외 여러 기법들을 활용하여 효과적으로 시각화 할 수 있는 방안을 수립하여야 함

IV. 효과적인 데이터 시각화를 위한 효율화 방안

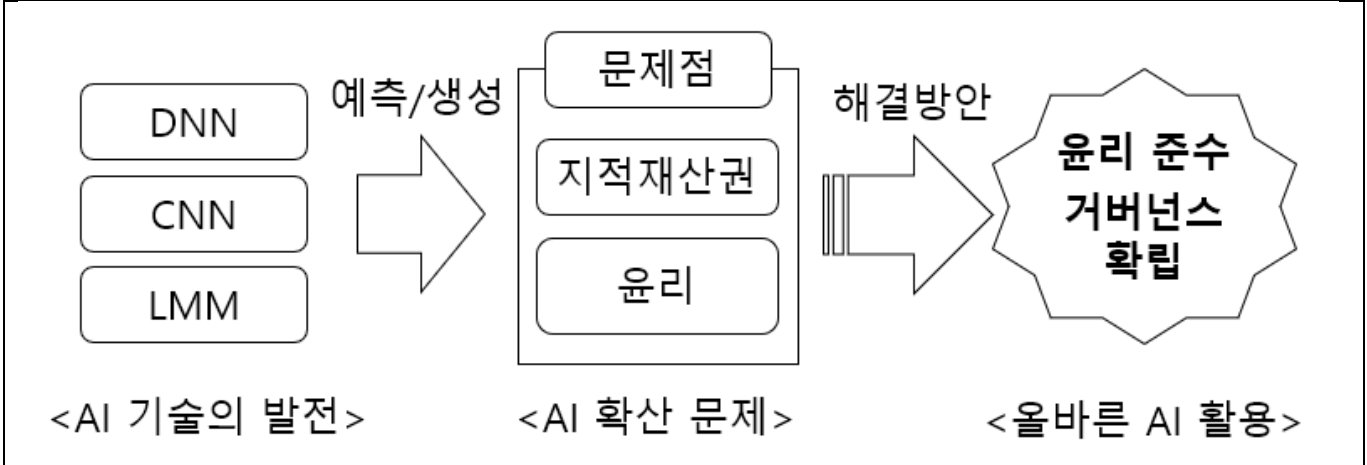
<div> <div>정보구조화</div> <div>정보시각화</div> <div>정보시각표현</div> </div>		
<div> <div> <ul style="list-style-type: none"> - 전달 스토리 및 대상 선정 - 데이터 크기 및 유형 선정 </div> <div> <ul style="list-style-type: none"> - 데이터 표현 기법 선정 - 단순화 </div> <div> <ul style="list-style-type: none"> - 적절한 레이아웃 검증 - 효과적 시각 자료 선택 </div> </div>		
단계	방안	설명
정보 구조화	데이터 크기/유형 선정	- 시각화를 위한 데이터 크기 및 유형을 명확한 기준에 따라 선정
	대상 선정	- 시각화 보고서를 전달할 대상을 선정
	주요 스토리 선정	- 시각화로 보여주고자 하는 스토리 선정
정보 시각화	데이터 표현 기법 선정	- 데이터에 맞는 표현기법 추출 및 선정
	단순화	- 다양한 데이터 표현기법 중 보고서의 스토리에 맞는 시각화 도구로 보고서 단순화 수행
정보 시각표현	레이아웃 검증	- 보고의 목적에 맞는 보고서 레이아웃 최종 검증
	효과적 시각 자료 선택	- 목적에 맞는 시각 자료 선택을 통한 효율성 증대

- 데이터 분석도 중요하지만 최종 결정을 위한 데이터 시각화를 통한 보고 및 전달의 간략화가 데이터 분석에서 매우 중요함

“끝”

03	AI윤리 및 AI 거버넌스		
문제	인공지능의 개발 및 적용과정에서 윤리적으로 다루어져야 할 주요 내용과 인공지능을 효과적으로 관리하고 규제하기 위한 거버넌스 모형에 대하여 설명하시오.		
도메인	인공지능	난이도	중(상/중/하)
키워드	존엄성, 공공선, 합목적성, 책임성, 프라이버시, 학습데이터, 규범, 규제		
출제배경	최근 인공지능 기술의 확산으로 인해 발생하는 윤리적인 문제와 해결방안에 대한 확인		
참고문헌	인공지능 거버넌스에 관한 탐색적 연구 (전북대학교 / 김길수) 정보정책론 (원광대학교 강의자료) ITPE 기술사회 자료		
해설자	모멘텀 안수현 기술사(제 119회 정보관리기술사 / itpe.momentum@gmail.com)		

I. 인공지능 기술의 확산에 따른 사회적 이슈



- 인공지능 기술의 발전 및 확산에 따른 윤리적인 문제를 해결해야 하며, 거버넌스 구축을 통한 관리 필요

II. 인공지능의 개발 및 적용과정에서 윤리적으로 다루어져야 할 주요 내용

가. 인공지능 윤리기준의 3대 기본원칙

3대 기본원칙	설명
인간 존엄성 원칙	<ul style="list-style-type: none"> - 인간은 인간을 위해 개발된 기계제품과는 교환 불가능한 가치가 있음 - 인간의 생명, 정신적 및 신체적 건강에 해가 되지 않는 범위에서 개발 및 활용 - 인공지능 개발/활용은 안전성과 견고성을 갖추어 인간에게 해가 되지 않도록 함.
사회의 공공선 원칙	<ul style="list-style-type: none"> - 공동체로서 사회는 가능한 한 많은 사람의 안녕과 행복이라는 가치를 추구 - 인공지능은 사회적 약자와 취약 계층의 접근성을 보장하도록 개발 및 활용 - 공익 증진을 위한 인공지능은 사회, 국가, 글로벌 관점에서 보편적 복지를 향상
기술의 합목적성 원칙	<ul style="list-style-type: none"> - 인공지능 기술은 인류의 삶에 필요한 도구라는 목적과 의도에 부합되게 개발 및 활용되어야 하며 그 과정도 윤리적이여야 함. - 인류의 삶과 번영을 위한 인공지능 개발 및 활용을 장려하여 진흥

- 인공지능 윤리기준의 3대 기본원칙은 개발 및 적용과정에서 고려될 원칙

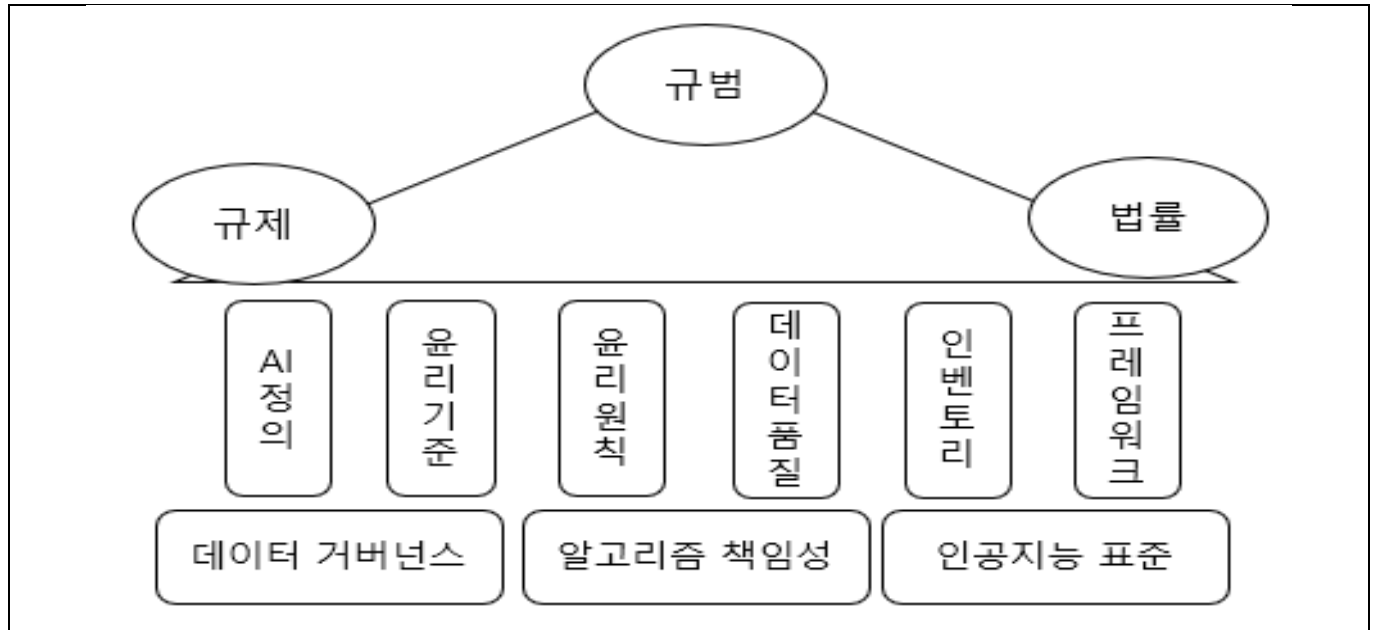
나. 인공지능 윤리기준의 10대 핵심요건

기본원칙	핵심요건	설명
인간 존엄성 원칙	인권보장	<ul style="list-style-type: none"> - 모든 인간의 권리를 존중, 민주적 가치와 국제인권법 등에 명시된 권리 보장 - 인공지능의 개발과 활용은 인간의 권리와 자유를 침해하지 않아야 함
	프라이버시 보호	<ul style="list-style-type: none"> - 인공지능을 개발/활용 전 과정에서 개인의 프라이버시를 보호 - 인공지능 전 생애주기에 걸쳐 개인 정보의 오용을 최소화
	다양성 존중	<ul style="list-style-type: none"> - 사용자의 다양성, 대표성 반영, 개인 특성에 따른 편향과 차별을 최소화하고, 모든 사람에게 공정하게 적용 - 사회적 약자, 취약 계층의 인공지능 기술 및 서비스에 대한 접근성 보장
사회의 공공성 원칙	침해금지	<ul style="list-style-type: none"> - 인공지능을 인간에게 직간접적인 해를 입히는 목적 활용 금지 - 인공지능의 위험과 부정적 결과에 대응 방안을 마련
	공공성	<ul style="list-style-type: none"> - 개인적 행복추구 뿐만 아니라 사회적 공공성 증진과 인류의 공동 이익을 위해 활용 - 인공지능은 긍정적 사회변화를 이끄는 방향으로 활용 - 인공지능의 순기능 극대화, 역기능 최소화위한 교육을 다방면으로 시행
	연대성	<ul style="list-style-type: none"> - 다양한 집단 간의 관계 연대성 유지, 미래세대를 충분히 배려하여 인공지능을 활용 - 인공지능 전 주기에 걸쳐 다양한 주체들의 공정한 참여 기회를 보장 - 윤리적 인공지능의 개발 및 활용에 국제사회 협력
기술의 합목적성 원칙	데이터 관리	<ul style="list-style-type: none"> - 개인정보 등 각각의 데이터를 목적 외 용도로 활용 금지 - 데이터 수집/활용 전 과정 데이터 편향성 최소화, 데이터 품질/위험 관리
	책임성	<ul style="list-style-type: none"> - 인공지능 개발/활용과정에서 책임주체 설정함으로써 발생할 수 있는 피해를 최소화 - 인공지능 설계 및 개발자, 서비스 제공자, 사용자 간의 책임소재를 명확화
	안전성	<ul style="list-style-type: none"> - 인공지능 개발/활용 전 과정에 걸쳐 잠재적 위험을 방지하고 안전을 보장 - 명백한 오류, 침해 발생 시 사용자가 작동을 제어할 수 있는 기능 마련
	투명성	<ul style="list-style-type: none"> - 타 원칙과의 상충관계 고려 적합한 수준의 투명성과 설명 가능성을 높임 - 제품/서비스 제공 시 인공지능의 위험 등의 유의사항을 사전에 고지

- 인간성을 위한 인공지능 (AI for Humanity)을 위해 인공지능 개발에서 활용에 이르는 전 과정에서 고려되어야 할 기준으로 제시

III. 인공지능을 효과적으로 관리하고 규제하기 위한 거버넌스 모형

가. 인공지능 거버넌스 모형



- 인공지능 표준을 준수하며, 인공지능 윤리를 기반으로 효율적으로 관리하는 다계층 AI 거버넌스

나. 인공지능 거버넌스 모형의 상세 설명

구분	구성요소	설명
기술 계층	알고리즘 책임성	- AI 알고리즘에 의한 데이터의 수집, 사용, 관리는 공정성을 촉진 - 인종, 국적, 지역, 성, 성적 취향, 장애여부 등에 의한 사회적 차별로부터 안전해야 한다는 원칙
	데이터 거버넌스	- 학습 데이터의 품질, 표준, 구조, 흐름, 보안등을 관리하기 위한 체계
	인공지능 표준	- 인공지능 모델링, 경량화 프레임워크, 추론 프레임워크 등 표준 제정 필요
윤리적 계층	윤리 기준	- AI 알고리즘에 의해 유도되는 행동은 윤리적인 기반을 준수
	윤리 원칙	- 인간의 존엄성, 사회의 공공선 등의 윤리원칙 준수
사회적/법적 계층	규범	- AI 프로세스 및 자율시스템 등의 책임감 있는 활용위한 방안 필요
	규제	- AI를 비롯한 자율 시스템을 규제하기 위한 제도 필요

- 단기적으로는 기술 계층의 요소, 장기적으로는 사회적/법적 계층의 요소 구축


“끝”

04	제로 트러스트 보안 모델		
문제	제로 트러스트 보안 모델의 보안원리, 핵심원칙, 적용분야를 트러스트 보안 모델과 비교 설명.		
도메인	보안	난이도	중(상/중/하)
키워드	선 인증 후 연결, 트러스트 보안 모델 한계, 무신뢰 기반, SDP, Micro-Segmentation, 강화된 인증 식별, 디바이스, 네트워크, 응용, 데이터, 성숙도 모델		
출제배경	과학기술정보통신부 제로 트러스트 가이드라인 1.0 공개 (2023년 7월)		
참고문헌	제로 트러스트 가이드라인 1.0		
해설자	이제원 기술사(제 130회 정보관리기술사 / bwmslove@naver.com)		

I. 트러스트 보안 모델의 한계, 제로트러스트 보안 모델 등장 배경

기존 보안으로 막기 어려운 공격 출현

2022년 랩서스(LAPSUS\$)의 글로벌 IT 기업 해킹을 통한 기밀정보 유출사례



1. 공격 대상 임직원 계정 정보 확보 후 내부 침투
2. 네트워크 경찰 및 주요 서버 접근, 권한 상승
3. 기밀 데이터 획득 및 외부 유출

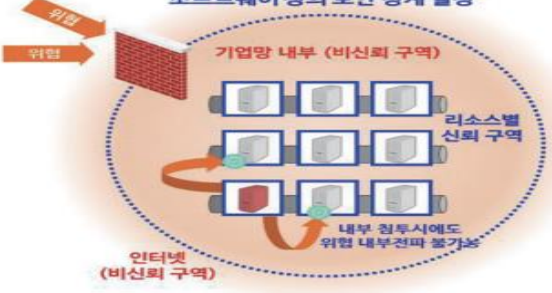
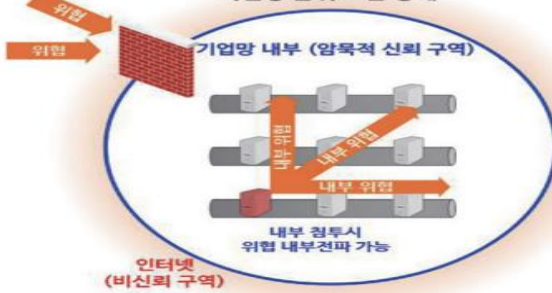
Why? 글로벌 대기업들은 높은 수준의 보안 기술을 사용중임에도 왜 해킹을 당했는가?

1. 탈취된 임직원 계정으로 로그인하여 기업망에 침투하는 해커를 막을 방법은 없는가?
2. 해커가 회사 관리 자산이 아닌 기기로 기업망에 접근할 때 어떤 정책을 적용해야 하는가?
3. 공격자가 내부 침투 후 기업망 내부 네트워크를 경찰하고 주요 서버에 접근하는 것을 막을 수 없는가?
4. 공격자가 대량의 기밀 데이터 획득 후 외부로 유출하는 행위를 막을 방법은 없는가?

- 공격표면 증가, 공급망 공격, Zero-day attack 등 기존 트러스트 보안 모델의 보안 한계로 인해 무신뢰 기반 제로 트러스트 보안 모델 필요성 대두

II. 제로 트러스트 보안 모델과 트러스트 보안 모델 보안원리 비교

가. 제로 트러스트 보안 모델과 트러스트 보안 모델 보안원리 개념 비교

제로트러스트 보안 모델	트러스트 보안 모델
<p style="text-align: center;">제로트러스트도입 후: 소프트웨어 정의 보안 경계 설정</p>  <p style="text-align: center;">기업망 내부 (비신뢰 구역) 리소스별 신뢰 구역 내부 침투시에도 위협 내부전파 불가능 인터넷 (비신뢰 구역)</p>	<p style="text-align: center;">제로트러스트도입 전: 기업망 단위 보안 경계</p>  <p style="text-align: center;">기업망 내부 (암묵적 신뢰 구역) 리소스별 내부 위협 내부 침투시 위협 내부전파 가능 인터넷 (비신뢰 구역)</p>
<ul style="list-style-type: none"> - 선인증 후연결, 내·외부 사용자와 프로세스를 신뢰하지 않는 원칙하에 모든 사용자와 디바이스, 어플리케이션에 대해 확인을 요구하는 보안 모델 	<ul style="list-style-type: none"> - 선연결 후인증, 네트워크 내부 접속 요구 사용자 및 기기는 어느 정도 신뢰할 수 있다는 가정하에 구축하는 보안 모델

- 제로 트러스트 보안 모델은 선인증 후연결 기반 누구도 신뢰하지 않는 원칙의 보안 모델

나. 제로 트러스트 보안 모델과 트러스트 보안 모델 보안원리 상세 비교

구분	제로 트러스트 보안 모델	트러스트 보안 모델
원칙	- 선인증 후연결	- 선연결 후인증
특징	- 추상적, 논리적	- 구체적, 물리적
신뢰 수준	- 내·외부 사용자, 디바이스 무신뢰	- 내부 네트워크 사용자, 디바이스 신뢰
인증과 권한 부여	- 모든 신원 및 권한 확인	- 내부 네트워크 추가 인증 권한 미확인
접근 제어	- 엄격한 접근 제어	- 내부 네트워크 접근 제어 완화
내부/외부 구분	- 내·외부 네트워크 경계 의존성 감소	- 내·외부 네트워크 엄격한 구분
보안 경계	- 어플리케이션 중심 보안 적용	- 네트워크를 중심으로 보안 적용

- 제로 트러스트 보안 모델은 내부 사용자 및 디바이스에 대해서도 무신뢰 기반의 보안 정책 적용

III. 제로 트러스트 보안 모델과 트러스트 보안 모델 핵심원칙 및 적용분야 비교

가. 제로 트러스트 보안 모델과 트러스트 보안 모델 핵심원칙 비교

구분	제로 트러스트 보안 모델	트러스트 보안 모델
인증	- 인증 체계 강화	- 내부 네트워크 신뢰
	- 선인증 후연결	- 선연결 후인증
구역 관리	- Micro Segmentation	- 내부 네트워크 세그먼트 간 통신 허용
	- 개별 자원을 배치하고 지속적 검증	- 내부 자원간 횡적이동 가능
경계	- 소프트웨어 정의 경계(SDP)	- 물리적 네트워크 경계 중요
	- 네트워크 동적 구성	- 네트워크 정적 구성

- 제로 트러스트 보안 모델 구현 핵심원칙은 인증 체계 강화, 마이크로 세그멘테이션, SDP 3개로 구성

나. 제로 트러스트 보안 모델과 트러스트 보안 모델 적용분야 비교

구분	제로 트러스트 보안 모델	트러스트 보안 모델
지사 및 원격 접속	- SSO, MFA	- 전용선, MPLS
	- RBAC, 암호화 통신	- VPN, VDI
3rd-party 및 기업간 협업	- ID Federation, MFA	- 방화벽, ACL
	- 모니터링, 실시간 접근 관리	- 래터럴 무브먼트 위협
업무망 및 인터넷망 망분리	- SDP, PDP, PEP	- 물리적(2pc, 스위치) 망분리
	- SPA, RBAC, Micro Segmentation	- 망연계 솔루션
온프레미스 및 클라우드	- IAM, EAM, MFA	- FW, TMS, IPS, IDS
	- CWPP, CSPM, CASB	- Bastion Host, ACL, CL
OT/ICS 산업제어	- Micro Segmentation, 최소 권한 원칙	- IT/OT 물리적 분리
	- 컨텍스트 기반 접근제어	- 강력한 암호화 및 인증 한계
M2M 환경	- TPM, 기기간 인증 및 암호화	- API 호출, 데이터베이스 전송

- 제로 트러스트 보안 모델 도입 시 기술, 문화, 정책, 규제 등 기업 내·외부 환경 관점의 고려사항 존재

IV. 제로 트러스트 보안 모델 도입 시 고려사항

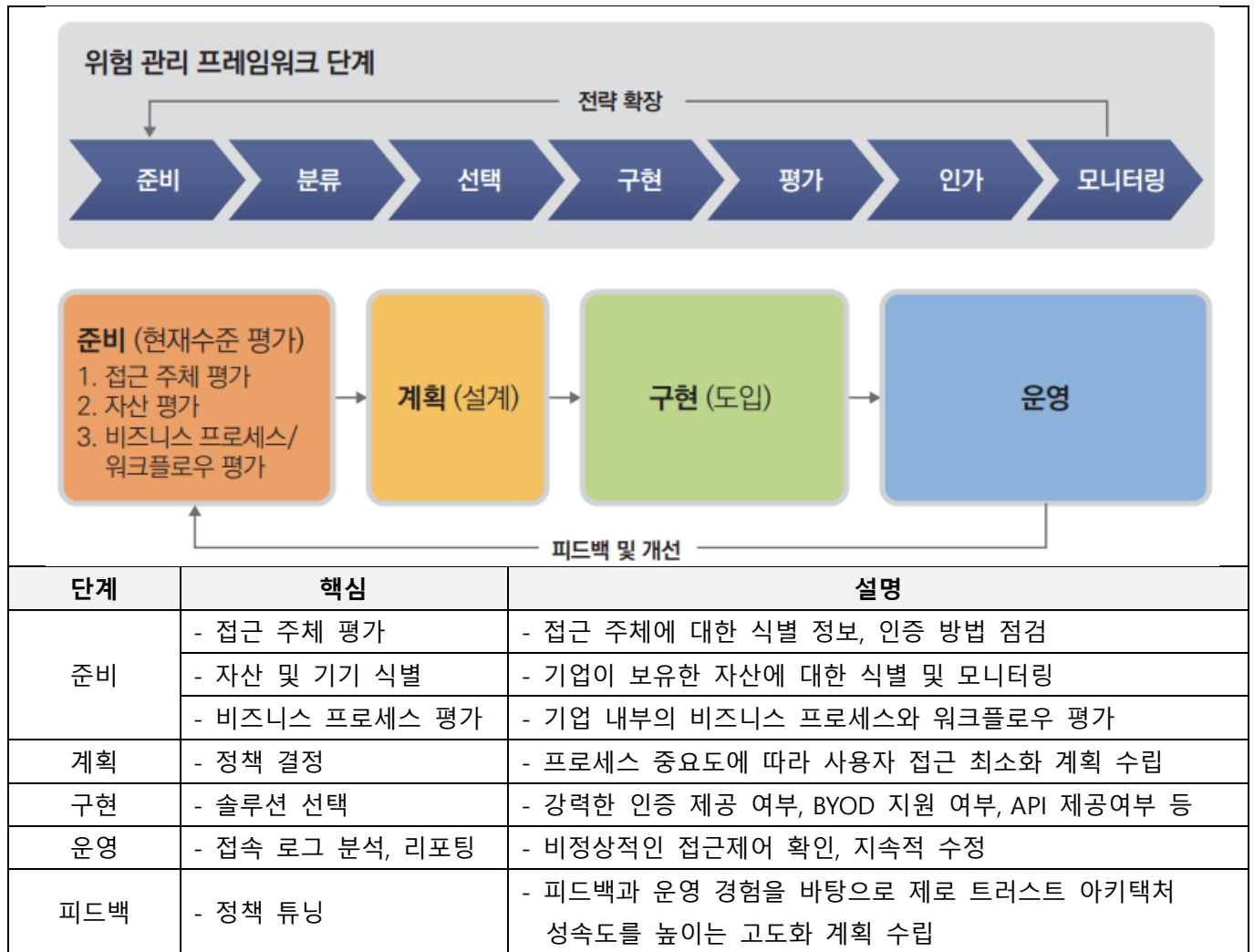
구분	고려사항	설명
기술	- 기존 레거시 기술 파악	- 기존 트러스트 기반 보안 모델에서 사용중인 기술 분석
	- 기존 기술 보완 방안	- 기존 기술을 보완하고 대체할 수 있는 방법 파악
문화	- 새로운 기술 적극 반영	- 변화에 대응하고 새로운 기술을 받아들이는 적극적인 문화
	- 경영진들의 의지	- 결정권자 및 이해관계자가 제로 트러스트 필요성 인지 필요
정책	- 접근 정책 수용성	- 인프라, 응용, 데이터 전체에 끼칠 영향에 대한 수용성 평가
	- 제로 트러스트 전환	- 2026년 공공 제로 트러스트 전환 정책 고려한 계획 수립
규제	- 국내 인증 제도 고려	- 제로 트러스트 보안 모델 도입 시 ISMS-P, 클라우드 서비스 보안인증 제도 등의 보호대책 준수 여부 고려

- 제로 트러스트 보안 모델 도입 시 기존 레거시 기술과의 보완 및 대체 방안을 고려하여 기업 특성에 맞는 전환정책 및 최적화 수준 결정

“끝”

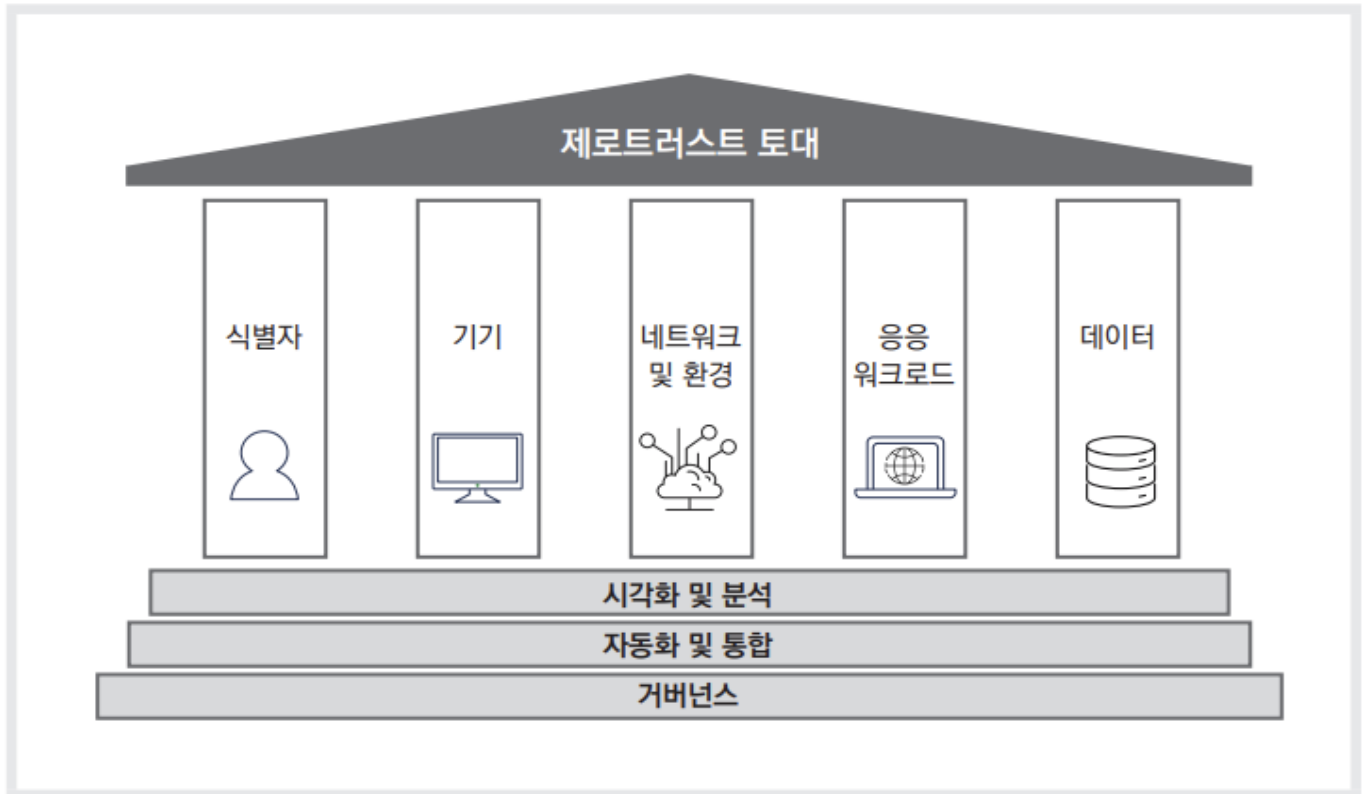
[참고자료]

1. 제로 트러스트 도입 단계



2. 제로 트러스트 프레임워크 개념도 및 핵심 요소

가. 제로 트러스트 프레임워크 개념도



(출처: CISA, "Zero Trust Maturity Model")

나. 제로 트러스트 프레임워크 핵심 요소

구분	고려사항	설명
5가지 핵심요소	- 식별자(Identity)	- 사람, 서비스, IoT 등 고유하게 식별 할 수 있는 집합 - RBAC, ABAC, 강력한 인증, 세밀한 접근제어 규칙 적용
	- 기기(Device)	- PC, 서버, IoT, BYOD 등 모든 하드웨어 장치
	- 네트워크 및 환경 (N/W & Enviornment)	- 유무선 네트워크, 클라우드, 인터넷 등 모든 통신 매체
	- 응용 워크로드 (App Workload)	- 응용 계층 및 컨테이너, 가상 머신 등의 보호 및 관리
	- 데이터(Data)	- 암호화 기법, 전송 데이터 보호, 데이터 유출 대응
3가지 교차 기능	- 시각화 및 분석	- 비정상 행위에 대한 탐지 및 가시성 제공으로 방어 기법 제공
	- 자동화 및 통합	- SOAR 통해 정의된 프로세스 내에서 자동화된 통합 보안 대응
	- 거버넌스	- 경영진의 관심, 조직적 문화, DMA 기반의 피드백

3. 국내 제로 트러스트 도입 시 발생 위험 및 대응 방안

구분	발생 위험	대응 방안
정책 측면	- 제로 트러스트 아키텍처 결정 과정 무력화 - 악의적인 내부자 위협	- 모든 설정 변경 시 기록 및 감사 - KMS, HSM, RBAC, ABAC
기밀성 측면	- 인증 수단 도용 - 시스템 및 네트워크 정보 저장소 보안	- 컨텍스트 기반 신뢰도 평가 알고리즘 - 중요 데이터 엄격한 접근제어 정책 적용
가용성 측면	- DoS 공격 또는 네트워크 장애 - 네트워크 가시성 위협	- FTS, HA 환경 구축 - 메타데이터, 머신 러닝 기반 트래픽 분석
신뢰성 측면	- 인공지능 또는 SW 기반 에이전트 인증 - 오탐과 미탐 가능성	- 설명가능한 AI, - 오탐/미탐에 대한 정기적인 분석 및 보완
상호운용성 측면	- 전용 데이터 규격 - 솔루션에 대한 의존	- 데이터 표준화, JSON, XML - 호환성, 이식성, 공급망 위험 등 종합 평가

4. 국내 제로 트러스트 성숙도 모델

성숙도 수준	해당 수준의 의미	해당 수준에서 요구하는 보안 기술 특징
기존 (Traditional)	아직 제로트러스트 아키텍처를 적용하지 않은 수준으로, 대체로 네트워크 방어에 초점을 맞춘 경계 기반 보안모델이 적용되어 있는 상태 (정교한 공격, 내부자 공격 등에 일부 취약성을 가짐)	<ul style="list-style-type: none"> 수동 설정 및 속성 부여, 정적 보안 정책 온프레미스 ID (때때로 SSO 및 다중인증방식 적용) 외부 시스템에 대해 정밀하지 않은 종속성을 가진 핵심 요소별 솔루션 프로비저닝에서 최소 기능 구축 정책 적용에서 독점적이고 유연하지 않은 핵심 요소 수동적인 사고 대응 및 완화 배포
향상 (Advanced)	제로트러스트 철학을 부분적으로 도입한 수준으로, 제로트러스트 원칙이 보안 아키텍처에서 핵심 기능이 되는 상태 (최소 권한 접근, 네트워크 분할, 로깅 및 모니터링 등이 부분적으로 적용되어 기본보다 높은 보안성 달성)	<ul style="list-style-type: none"> 세밀한 수준에서의 사용자 및 기기 접근제어 중앙 집중적 ID 제어 및 정책 적용, 상태 평가에 기반한 일부 최소 권한 변경 네트워크가 일부 세분화되어, 전체 환경에 영향을 미치는 공격의 위험이 줄어듦 일부 핵심 요소간 정합 중앙 집중적 가시성 부분 제공 사전 정의된 완화 기법을 통한 일부 사고 대응 외부 시스템과 종속성 측면에서 세부 정보 증가
최적화 (Optimal)	제로트러스트 철학이 전사적으로 적용된 상태 (자동화된 운영, 네트워크 세분화, 신원에 대한 지속적인 검증을 통한 최소 권한의 안전한 접근제어 등을 통하여 보안성이 크게 개선)	<ul style="list-style-type: none"> Federated ID 및 싱글사인온 등 ID 통합 관리 및 지속적 신원 및 신뢰도 검증, 자동화/실시간 분석을 통한 동적 접근제어 자산 및 리소스에 대한 완전 자동화된 속성 부여, 최소 권한 접근을 위한 종속성 부여 네트워크는 고유한 접근 규칙을 갖는 세분화된 영역으로 구분 핵심 요소간 상호운용성을 위한 개방형 표준을 통한 조정 특정 시점의 상태 기억을 위한 히스토리 기능을 가진 중앙 집중적 가시성 제공

05	소켓(Socket) 통신, HTTP 통신		
문제	소켓(Socket) 통신과 관련하여 다음을 설명하시오. 가. 소켓 통신 정의 나. 소켓 통신 방식 개념도 및 유형 다. TCP 소켓 및 Web 소켓 흐름 설명 라. 소켓 통신 방식과 HTTP 통신 방식 비교		
도메인	네트워크	난이도	중(상/중/하)
키워드	양방향성, 실시간, TCP, UDP, read, write, accept		
출제배경	소켓통신과 HTTP 통신방식 기본이해 확인		
참고문헌	ITPE 서브노트		
해설자	정유나 기술사(제 130회 정보관리기술사 / audfla89@naver.com)		

I. 서버와 클라이언트 양방향 통신방식, 소켓 통신 개념

가. 소켓 통신의 정의

- 두 프로그램이 서로 데이터를 주고 받기 위해 생성된 통신 단자(소켓)를 이용하여, 클라이언트와 서버 양쪽에서 서로에게 데이터를 전달하는 양방향 통신방식

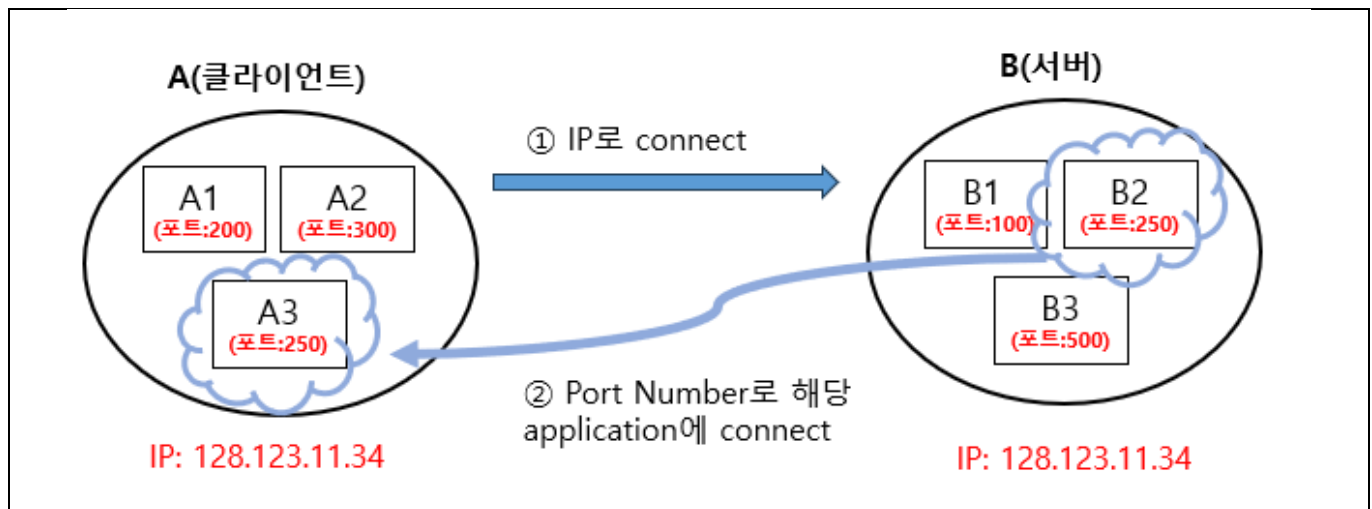
나. 소켓 통신의 특징

양방향 통신	클라이언트와 서버간 상호작용하는데 적합
다양한 데이터 처리	텍스트, 이진 데이터, 이미지 등 다양한 형식 지원
실시간 통신	웹환경에서 실시간 통신 구현

- 스트리밍이나 실시간 채팅 등 실시간으로 데이터를 주고받아야 하는 경우 활용

II. 소켓 통신 방식 개념도 및 유형

가. 소켓 통신 방식 개념도



- 소켓은 IP, 포트 넘버 기반으로 통신하며 특성과 용도에 따라 TCP, UDP방식으로 분류

나. 소켓 통신 방식의 유형

유형	핵심요소	설명
TCP 통신	연결 지향	클라이언트가 서버에 연결 요청을 보내면 서버는 이를 수락하여 양방향 통신 채널을 설정
	신뢰성	오류가 발생한 경우에도 재전송하여 데이터의 정확성을 유지
	순서 보장	전송된 데이터는 도착 순서대로 재조립되므로, 순서를 보장
	느린 전송	데이터의 신뢰성을 위해 추가적인 프로토콜 헤더 및 확인 절차가 필요하므로 상대적으로 느린 전송 속도 보유
UDP 통신	비연결 지향	클라이언트와 서버 간에 연결 설정 과정이 없으며, 데이터를 전송하기만 하고 응답을 기다리지 않음
	신속한 전송	추가적인 프로토콜 헤더나 확인 절차가 없어 TCP에 비해 더 빠른 전송 속도 보유
	신뢰성 감소	데이터 손실이나 손상 시 재전송을 하지 않음
	순서 미보장	데이터가 도착하는 순서가 보장되지 않음

- 소켓 통신 유형은 TCP, Web 2가지 유형으로 분류

III. TCP 소켓 및 Web 소켓 흐름 설명

가. TCP 소켓 흐름 개념

개념도	<pre> graph TD subgraph Server S_socket[socket()] --> S_bind[bind()] S_bind --> S_listen[listen()] S_listen --> S_accept[accept()] S_accept --> S_rw[read()/write()] S_rw --> S_close[close()] end subgraph Client C_socket[socket()] --> C_connect[connect()] C_connect --> C_rw[read()/write()] C_rw --> C_close[close()] end S_listen --> C_connect S_rw <--> C_rw </pre>	
개요	클라이언트와 서버 간 신뢰성 있는 연결을 설정하고 데이터를 주고받는 동작 방식	
서버 측면	1. 소켓 생성	socket() 함수를 이용하여 소켓을 생성
	2. IP, Port번호 설정	bind() 함수로 ip와 port 번호를 설정
	3. 클라이언트 대기 결정	listen() 함수로 클라이언트의 접근 요청에 수신 대기열을 구성함 (몇 개의 클라이언트를 대기시킬 지 결정)
	4. 클라이언트 연결 대기	accept() 함수를 사용하여 클라이언트와의 연결을 기다림
클라이언트	1. 소켓 Open	socket() 함수 이용하여 소켓을 Open

측면	2. IP, Port번호에 통신	connect() 함수를 이용하여 통신 할 서버의 설정된 ip와 port 번호에 통신을 시도
	3. 소켓 Descriptor 반환	통신 시도 시, 서버가 accept() 함수를 이용하여 클라이언트의 socket descriptor를 반환
	4. 서버와 통신	클라이언트와 서버가 서로 read(), write() 를 하며 통신

- 서버가 먼저 소켓을 열고 클라이언트와 서로 read, write를 수행하며 통신 과정이 반복됨

나. 웹소켓 흐름 개념

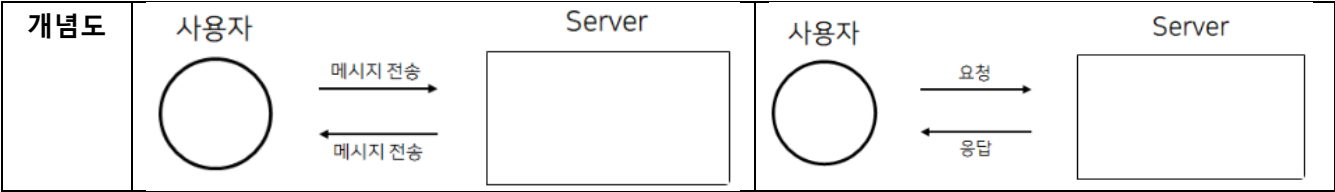
개념도		
개념	HTML5 표준 기술로, HTTP 환경에서 클라이언트와 서버 사이에 하나의 TCP 연결을 통해 실시간으로 전이중 통신을 가능하게 하는 프로토콜	
클라이언트 측면	1. 클라이언트 소켓 생성	연결 대상에 대한 정보가 들어있지 않은 소켓을 생성
	2. 연결 요청	IP, 포트번호로 연결하고 싶은 대상을 특정
	3. 데이터의 송수신	요청에 대한 결과를 수신(Send, Receive)
	4. 소켓 Close	데이터 송수신 필요 없어지게 되면 소켓 Close
서버 측면	1. 서버 소켓 생성	연결 대상에 대한 정보가 들어있지 않은 소켓 생성
	2. 서버 소켓 바인딩	소켓과 포트 번호를 바인딩
	3. 클라이언트 연결요청 대기	포트번호와 바인딩 작업 마친 후 클라이언트 요청 대기
	4. 클라이언트 연결 수립	최종 요청을 받아들이면서(accept) 다른 연결 처리 위해 listen이나 close

- 채팅앱, SNS, 구글 Docs, LOL 같은 멀티플레이 게임, 화상회의 등 다양한 분야에서 사용

IV. 소켓 통신 방식과 HTTP 통신 방식 비교

가. 소켓 통신 방식과 HTTP 통신 방식 개념 비교

구분	소켓 통신	HTTP 통신
개념	클라이언트와 서버 양쪽에서 서로에게 데이터 전달을 하는 방식의 양방향 통신	클라이언트의 요청이 있을 때 서버가 응답하며, 상태를 저장하지 않는(Stateless) 단방향 통신



- 소켓은 양방향, HTTP 통신은 단방향으로 구성되며 각각 다른 통신 용도로 사용

나. 소켓 통신 방식과 HTTP 통신 방식 상세 비교

구분	소켓 통신	HTTP 통신
연결 상태	실시간 데이터 송수신에 적합	일괄적인 요청과 응답에 적합
사용 용도	바이트 단위의 데이터를 처리	웹 리소스를 요청하는 용도로 사용
속도	상대적 빠름	상대적 저하
방화벽 및 보안	방화벽 등 추가적 보안 설정 필요	HTTPS 사용하여 암호화된 통신 지원
단점	Stateful한 연결상태 유지하므로 부하발생	실시간성 데이터 부적합

- 실시간 통신이나 상태 유지가 필요한 경우는 소켓 통신 방식이 적합

“끝”

06	아키텍처 스타일과 디자인 패턴		
문제	<p>아키텍처 스타일과 디자인 패턴에 대하여 다음을 설명하시오.</p> <p>가. 아키텍처 스타일과 디자인 패턴의 차이점</p> <p>나. 대표적인 아키텍처 스타일 3가지</p> <p>다. GoF 디자인 패턴의 유형을 구분하고, 유형별 대표적인 디자인 패턴 설명</p>		
도메인	소프트웨어 공학	난이도	하(상/중/하)
키워드	Layered 패턴, 블랙보드 패턴, MVC, 생성, 구조, 행위		
출제배경	소프트웨어 공학 기본 토픽		
참고문헌	ITPE 기술사 자료		
해설자	이제원 기술사(제 130회 정보관리기술사 / bwmslove@naver.com)		

I. 아키텍처 스타일과 디자인 패턴의 개념 및 상세 차이점

가. 아키텍처 스타일과 디자인 패턴의 개념 차이점

아키텍처 스타일	디자인 패턴
<ul style="list-style-type: none"> - 소프트웨어를 구성하는 요소 및 관계 타입과 더불어 어떻게 사용되는지에 대한 제약사항 집합을 기술하여 반복되는 아키텍처 컨셉을 서술한 소프트웨어 아키텍처를 위한 설계 유형 모음 	<ul style="list-style-type: none"> - 소프트웨어 상세 설계에서 자주 발생하는 문제에 대한 일반적이고 반복적인 해결책을 제안하며, 특정한 상황에서 구조적인 문제를 해결하는 모음

- 아키텍처 스타일과 디자인 패턴은 소프트웨어 설계에 있어 적용 시점 및 대상에 있어 차이점 존재

나. 소프트웨어 아키텍처 스타일과 디자인패턴 상세 비교

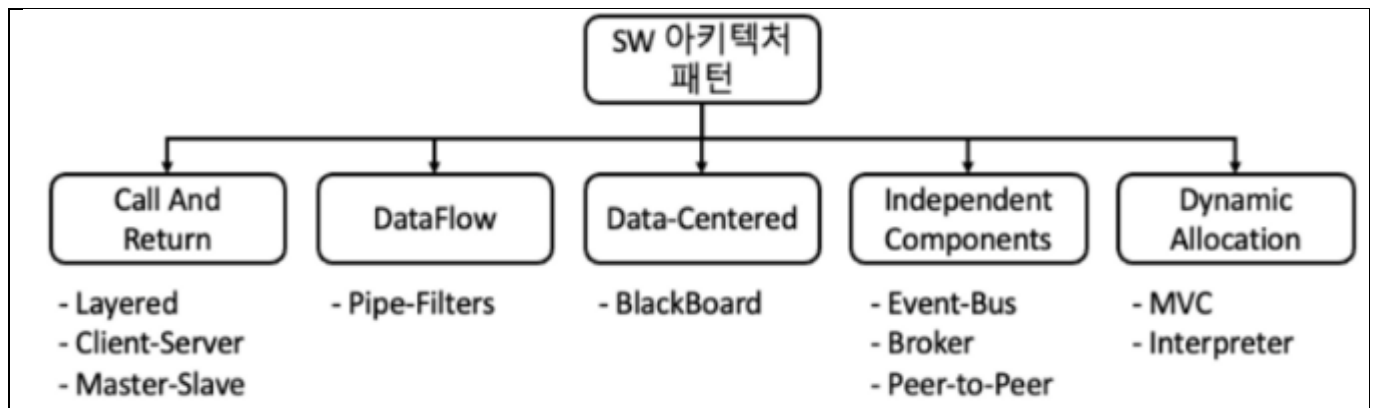
구분	소프트웨어 아키텍처 스타일	디자인 패턴
적용수준		

적용시점	- 소프트웨어 기본 설계 단계	- 소프트웨어 상세 설계 단계
관점	- 조직적 구성 관점 - 경험 기반의 아키텍처 구성	- 개발자의 구현 관점 - 경험 기반의 모델링
특징	- 아키텍처의 품질요소를 개선 - 기능적 컴포넌트를 변경하고 확장에 초점 - 동일한 기능을 수행하는 또다른 아키텍처 구성 패턴 사용의 제약	- 단순화, 이해 용이성 - 분석과 설계에 대한 추상적인 관점 제공 - 개발자가 이해하여 적용할 수 있는 형태로 제공 - 확장성, 재사용성, 이해용이성 등 제공
주요 유형	- 분산스타일 - 파이프와 필터 스타일 - 데이터 중심 스타일 - 규칙기반 스타일 - 블랙보드 스타일 - 레이어 스타일	- Adaptor Pattern - Factory Pattern - Command Pattern - Interpreter Pattern - Façade Pattern - Builder Pattern
문제해결측면	- 접근 방법에 집중 - 경량화 된 가이드라인	- 특정 컨텍스트에서의 검증된 구조적 해결 방안 제시

- 소프트웨어 아키텍처 스타일을 이용해 소프트웨어 전반의 아키텍처 기본 설계를 마친 후 상세설계 시점에서 디자인 패턴을 활용하여 패키지 내부의 클래스 간의 관계를 설정

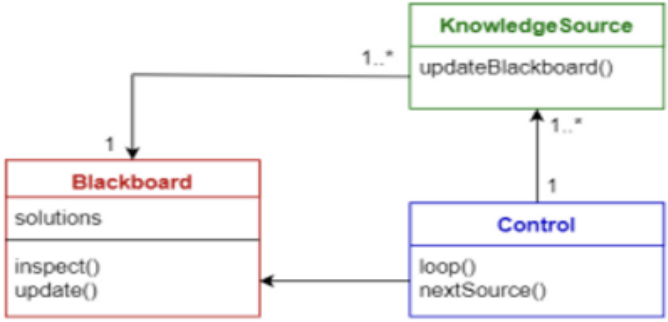
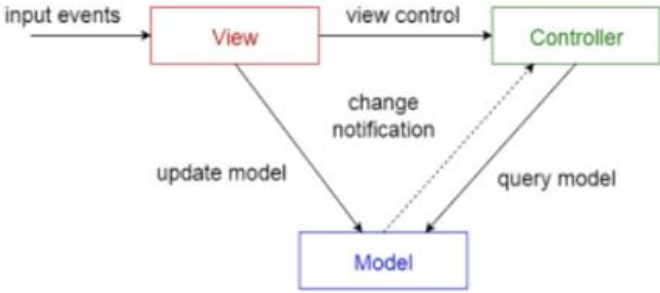
II. 대표적인 아키텍처 스타일 유형 및 3가지 상세

가 . 아키텍처 스타일 유형



나. 대표적인 아키텍처 스타일 3가지 상세 설명

구분	소프트웨어 아키텍처 스타일	설명
Client-Server	<p>Client와 Server가 TCP/IP를 통해 연결된 구조.</p>	<ul style="list-style-type: none"> - 클라이언트는 서버에게 서비스 요청하고 서버는 관련된 서비스를 클라이언트에게 제공

BlackBoard	 <pre> classDiagram class Blackboard { solutions inspect() update() } class KnowledgeSource { updateBlackboard() } class Control { loop() nextSource() } Blackboard "1" -- "1..*" KnowledgeSource Control "1" -- "1..*" KnowledgeSource Control --> Blackboard </pre>	<ul style="list-style-type: none"> - Blackboard: 솔루션에서 가져온 객체를 보유한 구조화된 전역 메모리 - Knowledge source: 각각의 표현 기술 - Control: 선택, 설정, 실행모듈
MVC	 <pre> classDiagram class View class Controller class Model View --> Controller : view control Controller --> View : change notification Controller --> Model : query model Model --> View : update model </pre>	<ul style="list-style-type: none"> - 사용자 인터페이스로부터 비즈니스 로직을 분리하여 상호 영향 최소화

- 이외에도 Micro Service Architecture, Event-bus, Pub-Sub 등 다양한 아키텍처 스타일 존재

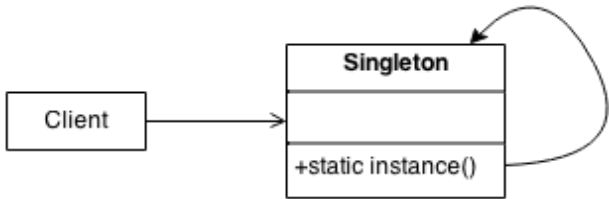
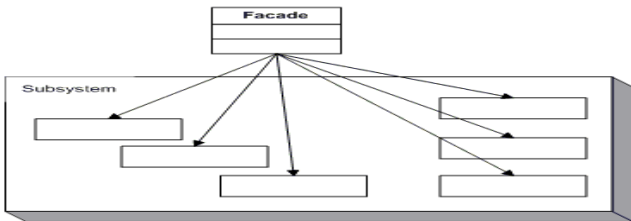
III. GOF 디자인 패턴 유형 및 유형별 대표 디자인 패턴 3가지

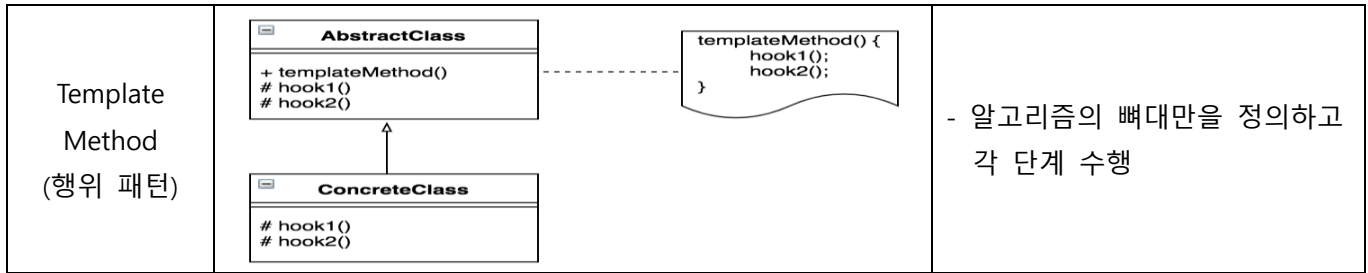
가. GOF 디자인 패턴 유형

유형	핵심	디자인 패턴
생성 패턴	- 객체의 생성방식을 결정하는 패턴	- Factory Method, Abstract Factory, Builder, Prototype, Singleton
구조 패턴	- Object 조직화하는데 유용한 패턴	- Adapter, Bridge, Composite, Decorator, Façade, Flyweight, Proxy
행위 패턴	- Object의 행위를 조직,관리,결합하는데 사용되는 패턴	- Interpreter, Template Method, Command, Iterator, Mediator, Memento, Observer, Strategy

- GOF의 디자인 패턴은 생성 패턴, 구조 패턴, 행위 패턴 3가지 유형으로 분류

나. GOF 유형별 대표 디자인 패턴 3가지

구분	디자인 패턴	설명
Singleton (생성 패턴)	 <pre> classDiagram class Client class Singleton { +static instance() } Client --> Singleton Singleton --> Singleton </pre>	<ul style="list-style-type: none"> - 인스턴스에 접근할 수 있는 전역적인 접촉점 제공
Façade (구조 패턴)	 <pre> classDiagram class Façade class Subsystem { * } Façade --> Subsystem </pre>	<ul style="list-style-type: none"> - 상위 수준의 통합된 하나의 인터페이스 제공



- 아키텍처 스타일과 디자인 패턴 적용 시 패턴 중독을 피하고 효과적으로 적용하기 위한 고려사항 존재

IV. 아키텍처 스타일과 디자인 패턴 적용 시 고려사항

구분	고려사항	설명
수용성 측면	- 과도한 복잡성 피하기	- 필요 이상의 많은 패턴은 유지보수와 이해용이성 저하
	- 일관성과 표준 준수	- 일관성이 부족한 코드는 이해하기 어려우며 빈번한 버그 유발
환경 측면	- 팀의 스킬과 경험 고려	- 팀, 이해관계자가 익숙한 패턴 사용
	- 문서화와 지식 공유	- 문서화와 지식 공유를 통해 이해도 향상 및 학습 기회 제공
목적 측면	- 목적과 맥락 고려	- 인프라, 응용, 데이터 전체에 끼칠 영향에 대한 수용성 평가
	- 문제에 따라 선택	- 문제에 맞는 가장 간단하고 효과적인 패턴 적용
설계 측면	- 성능 고려	- 대규모 시스템 적용 시 성능을 고려하여 패턴 선택 및 최적화
	- 테스트와 디버깅 고려	- 코드의 테스트 용이성에 대한 고려 필요

- 아키텍처 스타일과 디자인 패턴을 신중하게 선택하고 적용함으로써 소프트웨어의 품질과 유지보수성 향상

“끝”



ITPE 기술사회

제131회 정보처리기술사 기출문제 해설집

대 상	정보관리기술사, 컴퓨터시스템응용기술사, 정보통신기술사, 정보시스템감리사 시험
발행일	2023년 08월 26일
집 필	강정배PE, 전일PE, 이상현PE, 안수현PE, 이제원PE, 정명림PE
출 판	ITPE(Information Technology Professional Engineer)
주 소	ITPE 대치점 서울시 강남구 선릉로 86길 17 선릉엠티빌딩 7층 ITPE 선릉점 서울시 강남구 선릉로 86길 15, 3층 IT교육센터 아이티피이 ITPE 강남점 서울시 강남구 테헤란로 52길 21 파라다이스벤처타워 3층 303호 ITPE 영등포점 서울시 영등포구 당산동2가 하나비즈타워 7층 ITPE
연락처	070-4077-1267 / itpe@itpe.co.kr

본 저작물은 [ITPE\(아이티피이\)](#)에 저작권이 있습니다.

저작권자의 허락없이 **본 저작물을 불법적인 복제 및 유통, 배포**하는 경우
법적인 처벌을 받을 수 있습니다.