

# 제39회 ITPE 실전 명품 모의고사 해설집

2025.12.21

## 제 39 회 ITPE 실전 명품 모의고사

일시 : 2025 년 12 월 21 일

제 1 교시(시험시간: 100 분)

분야	정보통신	자격종목	정보관리 컴퓨터 시스템 응용	수검 번호		성 명	
----	------	------	--------------------	----------	--	--------	--

※ 다음 문제 중 10 문제를 선택하여 설명 하십시오. (각 10 점)

1. 옛지 AI 와 온디바이스 AI 비교
2. 디지털 출처(Digital Provenance)
3. ISO 26262 기반 백투백 테스트(Back-to-Back test)의 목적과 절차
4. 데이터 스페이스(Data Space)
5. 생체인식정보 보호 6 대 원칙
6. SDDC(Software-Defined Data Center)의 기술요소 설명 및 Data Center 와 비교
7. 임베디드 파이낸스(Embedded Finance)
8. 에이전틱 코딩(Agentic Coding)
9. 데이터베이스 리플리케이션(Replication)과 쿼리 오프로딩(Query-off Loading)
10. 프롬프트 인젝션(Prompt Injection)
11. 멀티 에이전트 시스템(Multi Agent System)
12. 하이퍼스케일 데이터센터(Hyperscale Data Center)

[정보관리기술사 선택문제]

13. 데이터 중심 AI(Data-Centric AI)

[컴퓨터시스템응용기술사 선택문제]

13. 상위 인터리빙 (High-Order Interleaving) 과 하위 인터리빙 (Low-Order Interleaving) 설명

01	AI 경량화		
문제	엣지AI와 온디바이스AI 비교		
도메인	인공지능	난이도	중 (상/중/하)
키워드	경량 AI, NPU, AI 가속기, 프라이버시 보호, 실시간처리, 엣지 클러스터, 스마트 팩토리		
출제배경	대형 LLM 모델에서 경량화 AI 모델로 변화되는 기술 패러다임 전환에 따른 개념 확인		
참고문헌	ITPE 서브노트		
출제자	배미경 기술사(제 135회 정보관리기술사 / hjmom0727@daum.net)		

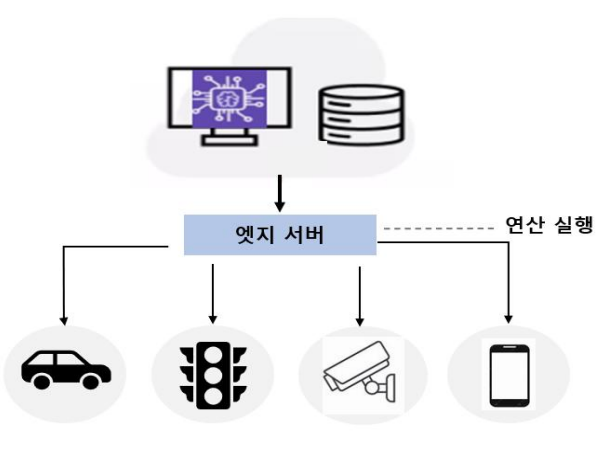
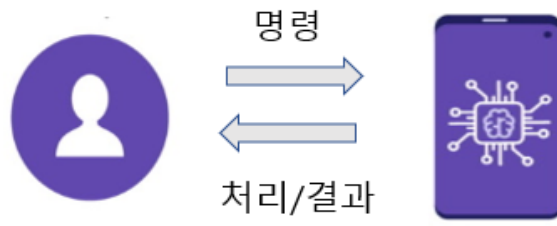
### I. 분산, 현장 중심의 경량화AI, 엣지AI와 온디바이스AI 개념 비교

엣지 AI	온디바이스 AI
- 데이터가 발생하는 지점 (엣지) 근처의 다양한 디바이스들 (엣지 서버, 센서, 사물인터넷 IoT 디바이스 등)에 AI 모델을 배포하여 데이터를 처리하는 AI	- 스마트폰, 노트북, 웨어러블 기기 등 개별 디바이스 자체에서 데이터를 수집하고, AI 엔진을 내장하여 연산을 처리하는 AI

- 엣지AI와 온디바이스AI 모두 클라우드 의존도를 줄이고, 경량화 하여 실시간 대응력, 보안강화 및 에너지 효율을 증대하는 AI 모델

### II. 엣지AI와 온디바이스AI 상세 비교

#### 가. 엣지AI 와 온디바이스AI 개념도 비교

엣지 AI	온디바이스 AI
	
- 엣지 서버, 게이트웨이 등에 AI 가속기 탑재하여 연산 처리	- 단말 디바이스 내에 NPU 등 AI 가속기 탑재하여 연산처리

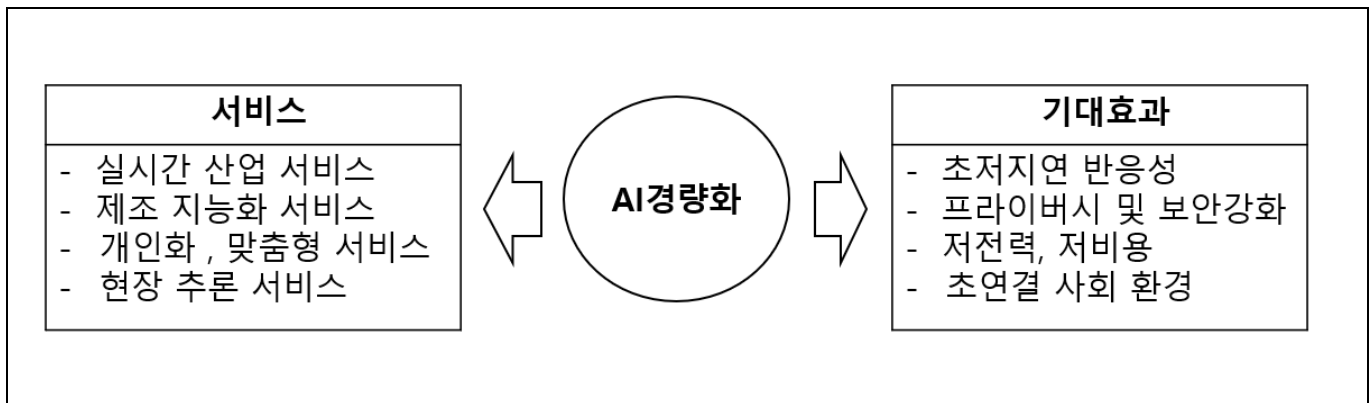
- 엣지AI는 더 넓은 범위의 장치와 산업응용에 초점을 맞추어 처리하여 현장에서 AI를 활용, 온디바이스AI는 개인용 기기에서의 AI 구현에 중점

#### 나. 엣지AI 와 온디바이스AI 상세 비교

구분	엣지 AI	온디바이스 AI
목적	- 중앙(클라우드) 의존을 줄이고 실시간 응답·대역폭 절감·프라이버시 강화	- 디바이스 자체에서 독립적으로 AI 기능 수행. 낮은 지연, 오프라인 동작, 개인정보 보호
연산위치	- <b>디바이스 근처의 엣지 서버</b> (게이트웨이, 로컬 서버) 또는 엣지 노드 + 일부 연산은 디바이스에서 처리	- <b>단말</b> (스마트폰, 임베디드 MCU/SoC, IoT 센서) <b>내부</b> 에서 수행
모델크기	- 중형 모델	- 소형 ~ 초경량 모델
지연	- 낮은 지연 시간, <b>현장에서 처리</b>	- 지연 없음, <b>실시간 처리</b>
운영,관리	- 엣지 인프라 관리 필요 (배포, 업데이트, 모니터링)	- 디바이스별 배포, 버전관리 어려움
보안	- 데이터를 엣지에서 관리, 향상된 보안	- 데이터의 단말내 보관, <b>프라이버시 보호</b> 강화
기술요소	- <b>엣지 클러스터</b> / 오케스트레이션 - <b>경량화된 서버 HW</b> (AI 가속기) - <b>모델분할</b> (분산 추론), <b>컨테이너화</b>	- 모델 경량화 ( <b>Pruning</b> , 지식증류, 양자화 ) - 실시간처리 ( <b>Edge AI SDK</b> , 데이터 증강 ) - HW 가속기 ( <b>NPU</b> , 뉴로모픽칩, 뉴럴 엔진 ..)
적용분야	- <b>스마트 팩토리(현장 분석)</b> - 자율주행 보조(로컬 라우터) - 캠퍼스/빌딩 보안 - 통신사 MEC 서버	- 스마트폰의 음성인식/안면인식 등 <b>개인화된 AI서비스</b> - 웨어러블 헬스 측정 - IoT 센서 이벤트 필터링 - 카메라 임베디드 분석

- 엣지AI 와 온디바이스AI는 개별적으로 서비스 운영이 되기도 하지만, 융합하여 운영되어 지능화된 초연결 사회의 가속을 촉진

#### III. 엣지AI와 온디바이스AI의 융합으로 인한 서비스 및 기대효과



- 엣지AI 와 온디바이스AI는 분산 및 현장에서 융합된 서비스로 구현되어 실시간 AI 및 지능형 환경 실행이 가능

“끝”

02	디지털 출처		
문제	디지털 출처(Digital Provenance)		
도메인	디지털서비스	난이도	상 (상/중/하)
키워드	디지털 자산 진위 및 검증, 메타데이터 스키마, 데이터 카탈로그, SBOM, 디지털 워터마킹, Timestamp, 버전관리시스템, API Trace, WORM		
출제배경	AI생성 콘텐츠 확산과 디지털 데이터 위.변조 증가로 신뢰 진위 검증기술의 중요성 증가		
참고문헌	<a href="http://www.ibm.com/kr-ko/think/topics/data-provenance">/www.ibm.com/kr-ko/think/topics/data-provenance</a> <a href="https://graphable.ai/blog/what-is-data-lineage-data-provenance">https://graphable.ai/blog/what-is-data-lineage-data-provenance</a>		
출제자	배미경 기술사(제 135회 정보관리기술사 / hjmom0727@daum.net)		

## I. 디지털 자산의 진위 및 검증 수단, 디지털 출처의 개요

### 가. 데이터 출처(Digital Provenance)의 정의

- AI가 생성한 콘텐츠, 오픈소스 코드, 서드파티 소프트웨어 등 데이터가 다양한 프로세스와 변환을 거치며 이동함에 따라 그 메타데이터를 캡처하여 데이터의 출처를 자세히 설명하는 데이터의 과거 기록
- 가트너 2026년 주요 전략 기술 트렌드

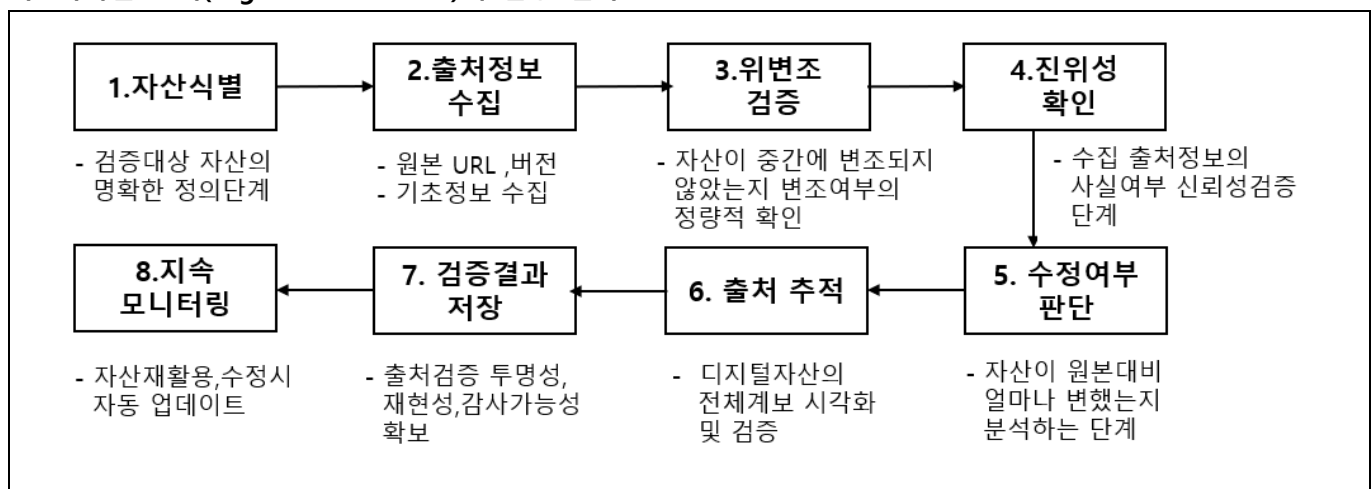
### 나. 디지털 출처(Digital Provenance)의 필요성

데이터 신뢰성 확보	- 데이터의 생성, 수정 등 추적을 통하여 데이터의 진위, 무결성을 보장
규제, 컴플라이언스 대응	- 데이터의 사용근거, 처리내역 기록, 감사시 데이터 처리경로 증빙
AI 모델 품질관리	- 어떤 버전의 데이터를 어떤 파이프라인을 통해 학습했는지 관리
데이터 조작.오염 탐지	- 데이터 계보를 통해 비정상 소스, 변조 이력, 오염된 처리 단계를 추적
데이터 라이프사이클 관리	- 데이터생성,처리,저장,분석,폐기까지 흐름 명확, 중복,불필요 데이터증가 방지

- 생성형 콘텐츠의 확산 시대에 데이터의 생성, 변환을 추적하여 데이터 신뢰유지를 위한 필수 기술

## II. 디지털 출처(Digital Provenance)의 검증 절차와 기술요소

### 가. 디지털 출처(Digital Provenance)의 검증 절차



- 데이터, AI 콘텐츠 등 디지털 자산의 출처, 변환, 사용 이력을 검증하는 표준 절차

## 나. 디지털 출처(Digital Provenance)의 기술요소

구분	핵심 기술	설명
자산식별	- 메타데이터 스키마	- 비정형 디지털 자산 기본정보 구조화
	- UUID / DID	- 고유 식별자 부여하여 자산 구분 및 추적 가능
	- Data Catalog	- 디지털 자산의 식별을 위한 자산 카탈로그
정보수집	- ETL Meta data	- 출처 증명을 위한 ETL 이력 추적
	- EXIF/XMP	- 이미지, 영상 메타 데이터 수집
	- SBOM / MLSBOM	- SW구성 컴포넌트, 메타데이터 등 체계적기록 자재명세서
위변조 검증	- 해시	- 임의길이 입력 데이터를 고정길이 해시값으로 변환하는 일방향 암호함수
	- 디지털 워터마킹	- 이미지, 영상 등 디지털 콘텐츠에 식별 정보 삽입
	- 전자서명	- 위변조 여부 보장위해 비대칭키 암호이용, 디지털형태 서명
진위성 확인	- PKI	- 공개키 암호활용 인증, 무결성 보장하는 디지털 신뢰기반
	- Timestamp	- 전자문서나 데이터에 신뢰할 수 있는 시각을 부여
	- 인증서 검증	- 브라우저/서버 루트 인증서 기반 경로 검증
수정여부 판단	- 포렌식	- 디지털 증거 수집, 보존, 분석, 제출하는 전 과정
	- 버전관리 시스템	- Git, SVN 등 디지털 파일의 변경 이력 저장 및 추적시스템
	- 데이터 드리프트 확인	- 데이터의 품질 변화 분석
출처 추적	- W3C PROV	- 디지털 출처를 기술하기 위한 표준 언어
	- 데이터/AI 라인리지 도구	- 디지털 자산이 어떤 경로로 사용되었는지 계보 재구성
	- API Trace	- API 기반 데이터 사용경로 추적
검증결과 저장	- WORM	- 수정 불가능한 저장 스토리지
	- Provenance Certificate	- 디지털 자산의 출처를 증명하는 인증서
	- Audit Log	- 감사 로그 저장

- AI 생성 콘텐츠 증가에 따라 디지털 자산의 소유권, 무결성 검증을 위하여, 공급망 전반에서 디지털 자산의 진위와 이동경로를 검증하는 수단으로 활용

## III. 디지털 출처(Digital Provenance)와 디지털 리니지(Digital Lineage)의 비교

비교 항목	디지털 출처	디지털 리니지
개념	- 데이터가 언제, 어디서, 누구에 의해, 어떤 과정을 거쳐 생성, 변경 되었는지 추적	- 전사 데이터 신뢰성 확보위해 데이터 생명주기 전과정 추적,관리하여 시각적으로 제공하는 기술
목적	- 데이터 유효성 검증/ 법적 규제 대응	- 데이터흐름 변환과정 이해/데이터품질 관리
단위	- 레코드/속성	- 테이블/컬럼
중점요소	- 출처 및 신뢰성	- 흐름 및 변환 과정

- 데이터 출처와 데이터 리니지는 데이터의 추적관리 측면에서는 유사하나, 다른 용도로 사용됨

“끝”

<b>03</b>	<b>SW 테스트</b>		
<b>문제</b>	ISO 26262 기반 백투백 테스트(Back-to-Back test) 목적 과 절차		
<b>도메인</b>	소프트웨어공학	<b>난이도</b>	<b>중</b> (상/중/하)
<b>키워드</b>	높은 신뢰성, 안전성 , ASIL 요구사항, MIL, SIL, HIL, 병행 테스트		
<b>출제배경</b>	높은 신뢰성이 요구되는 자동차 소프트웨어 개발 시 백투백 테스트 권장으로 절차 이해 필요		
<b>참고문헌</b>	ITPE 서브노트 <a href="https://www.dbpia.co.kr/journal/detail?nodeId=T14567930">https://www.dbpia.co.kr/journal/detail?nodeId=T14567930</a>		
<b>출제자</b>	배미경 기술사(제 135회 정보관리기술사 / hjmom0727@daum.net)		

## I. 높은 신뢰성, 설계명세 기반의 백투백 테스트의 개요

### 가. 백투백 테스트의 개념

- 두 개 이상의 다양한 컴포넌트나 시스템을 동일한 입력 값으로 실행하여 동일결과 여부 확인하여, 불일치시 비교 분석하는 테스트 기법

### 나. 백투백 테스트의 특징

<b>높은 신뢰성</b>	- 신뢰성이 절대적으로 요구되는 소프트웨어의 결함 발생 가능성 최소화
<b>설계 기반 명세</b>	- 설계명세 기반으로 구현된 결과의 정확성 확인

- ISO 26262 요구사항으로 ASIL의 모든 레벨에서 백투백 테스트를 권장

## II. ISO 26262기반 백투백 테스트의 목적과 절차

### 가. ISO 26262기반 백투백 테스트의 목적

구분	목적	설명
기능 관점	- 기능적 불일치 방지	- 모델 설계 와 구현 코드간 기능적 등가성 검증
	- 출력 및 동작 검증	- 알고리즘, 제어로직, 상태전이 등 검증
안전 측면	- ASIL 요구사항 충족	- 차량안전관련 기능이 ASIL 요구사항 충족하도록 보장
	- Safety Critical 검증	- 브레이크, 에어백 등 안전에 예민한 부분의 검증
품질 측면	- 오류 탐지 신뢰성 보장	- SW 구현 시 발생오류 탐지 및 불일치 감지 개선
	- 변경 회귀 관리 적용	- 변경내용 적용시 기존 기능 안전성 유지

- ISO 26262기반 백투백 테스트는 단순 기능 검증을 넘어 안전, 품질, 프로세스, 기능적 측면에서 자동차 소프트웨어 신뢰성과 안전성을 확보하기 위한 필수 검증 활동



나. ISO 26262기반 백투백 테스트의 절차

단계	절차	설명
준비	- 대상 및 환경 설정	- 테스트 대상 및 시뮬레이션 환경 준비
설계	- 테스트 케이스 설계	- 테스트 입력 정의 및 출력 기준 정의
	- 자동화 계획 수립	- 테스트 스크립트, 시뮬레이션 자동화 설계
테스트 (병행진행)	- MIL(Model in the Loop)실행	- SW 모델 자체를 시뮬레이션 환경에서 테스트
	- SIL(S/W in the Loop) 실행	- 코드생성 테스트, 코드와 모델 일치 검증
	- HIL(H/W in the Loop) 실행	- HW 연결하여 테스트, SIL 결과와 비교
검토.보고	- 테스트 결과 불일치 분석	- 동일한입력에 대한 MIL, SIL, HIL결과 비교, 일치성검증
	- 결과 보고 및 문서화	- 요구사항 준수 및 안전결함 발견 시 개선조치 기록

- ASIL 등급에 따라 테스트 범위와 심층도를 결정하고, 모델, 코드, 하드웨어 간의 동작일치 여부를 검증

III. 백투백 테스트와 A/B 테스트의 비교

비교 항목	백투백 테스트	A/B 테스트
목적	- 기능 일관성 비교, 검증	- 사용자 반응, 비즈니스 성과 비교
성격	- 엔지니어링, 기능 안전 목적 테스트	- 마케팅, 제품 실험 목적 테스트
적용	- 자동차, Safety Critical SW	- 웹, 모바일, 앱, 전자상거래 시스템 등
비교내용	- 기능, 로직	- 사용성, 효과 비교

- 백투백 테스트와 A/B 테스트는 두개 대상에 대한 테스트라는 점은 동일하나, 테스트 목적, 성격은 상이함

“끝”

04	데이터스페이스(Data Space)		
문제	데이터 스페이스(Data Space)		
도메인	디지털서비스	난이도	중 (상/중/하)
키워드	데이터 주권, 신뢰, 상호 운용성, 분산형 구조		
출제배경	과학기술정보통신부의 K-Data Space 정책 추진		
참고문헌	- 한국형 데이터 스페이스(K-Data Space) : AX 시대 데이터 공유·활용 패러다임 전환 전략(NIA, 2025.10.20) - <a href="http://www.ewestoday.co.kr/news/articleView.html?idxno=2331078">http://www.ewestoday.co.kr/news/articleView.html?idxno=2331078</a>		
출제자	박서현 기술사(제 131 회 정보관리기술사 / mondaysss@naver.com)		

## I. 신뢰 기반 데이터 교환 생태계, 데이터 스페이스(Data Space)의 정의

### 가. 데이터 스페이스의 개념

- 데이터 공유·활용 생태계로 기업, 기관, 개인이 만든 데이터를 공유 및 교환하면서 보안·주권을 유지할 수 있도록 만든 체계

### 나. 데이터 스페이스의 특징

분산 구조	- 연합(Federation) 기반의 분산 구조
정책 기반	- 사용 조건 등 정책 기반 설정 및 자동 이행

## II. 데이터 스페이스(Data Space)의 개념도와 구성요소

### 가. 데이터 스페이스의 개념도



- 한국형 데이터 스페이스 참조모델에서는 거버넌스, 비즈니스 계층, 기술 계층으로 구성

## 나. 데이터 스페이스(Data Space)의 구성요소

구분	구성 요소	설명
데이터 스페이스 거버넌스	- 연합형 및 분산형 구조	- 데이터 주권 보장, 합의된 공통 규칙 기반 구성
	- 신뢰성, 보안 및 투명성	- 신뢰 기반 데이터 교환 및 서비스 모델 운영 체계 구성
	- 거버넌스 유연성 및 자율성	- 거버넌스 유연성 및 참여자 자율성 확보 체계 구성
비즈니스 계층	- 도메인별 Use Case 개발	- 가치 창출 위한 실 사용 사례 발굴 및 확산
	- 참여자 등록 및 계약 관리	- 데이터 거래 참여자의 등록 및 역할, 계약 관리
	- 접근 권한 및 정책 설정	- 참여자별 접근 권한 및 이용 정책 설정 관리
	- 데이터 제품·서비스 제공	- 데이터 제품 및 데이터 거래 서비스 제공
기술 계층	- 표준 API, 데이터 커넥터	- 표준 기반 API 및 데이터 교환 도구 제공
	- ID 및 신뢰 관리	- DID, VC, eIDAS 등 참여자 인증 및 자격 검증
	- 정책·계약 관리 모듈	- Policy-as-Code 등 접근·사용 제어
	- 연합 카탈로그	- 표준화된 메타데이터 기반 데이터 상품 및 서비스 등록
	- 메타데이터 브로커	- 거래 대상 데이터 상품 탐색
	- 클리어링하우스	- 데이터 거래 이력 추적 및 정산 모니터링
	- 전자결제 및 디지털 토큰	- NFT, 스테이블 코인 등 기반, 결제 자동화 및 정산

- 데이터 스페이스는 데이터 공급자-수요자 간 협업을 촉진하며 자율적으로 비즈니스 모델을 개발하고 고부가가치 데이터의 안전한 활용을 넘어, 새로운 데이터의 생산·창출·유통까지 이어지는 선순환적 데이터 생태계 조성 지원

## III. 한국형 데이터 스페이스(Data Space) 추진 전략

데이터 스페이스 거버넌스	<ul style="list-style-type: none"> <li>• 참여 구조 정립</li> <li>• 운영 규칙 수립</li> <li>• 데이터 표준 정립</li> </ul>
데이터 스페이스 구축	<ul style="list-style-type: none"> <li>• 데이터 연계 체계 구축</li> <li>• 국가 데이터 인프라 컴플라이언스 연계</li> <li>• 법, 제도 보안 기준 적용</li> </ul>
비즈니스 가치 창출	<ul style="list-style-type: none"> <li>• AI 혁신 서비스 실증, 확산</li> <li>• 고품질 AI 학습용 데이터셋 생산 및 유통</li> </ul>

- 한국형 데이터스페이스 운영모델(제휴형·출자형·조합형 등)을 정립하고 초기에는 정부가 공공·민간 수요를 반영한 데이터 스페이스 시범사업을 직접 주도하고, 이후 민간 중심 자율적 확산이 가능하도록 운영 가이드라인과 기술 지원 체계 마련 필요

“끝”

05	생체인식정보 보호 6 대 원칙		
문제	생체인식정보 보호 6 대 원칙		
도메인	보안	난이도	하 (상/중/하)
키워드	(적비목투통안) 적법성, 비례성, 목적제한, 투명성, 통제권보장, 안정성		
출제배경	생체인식정보의 안전한 활용기반을 조성하기 위한 생체정보 보호 안내서 발간		
참고문헌	생체정보 보호 안내서(개인정보보호위원회, 2024.12)		
출제자	박서현 기술사(제 131 회 정보관리기술사 / mondaysss@naver.com)		

## I. 개인정보의 한 유형, 생체인식정보의 개요

### 가. 생체인식정보의 정의

- 생체정보 중 특정 개인을 인증, 식별할 목적으로 일정한 기술적 수단을 통해 처리되는 정보

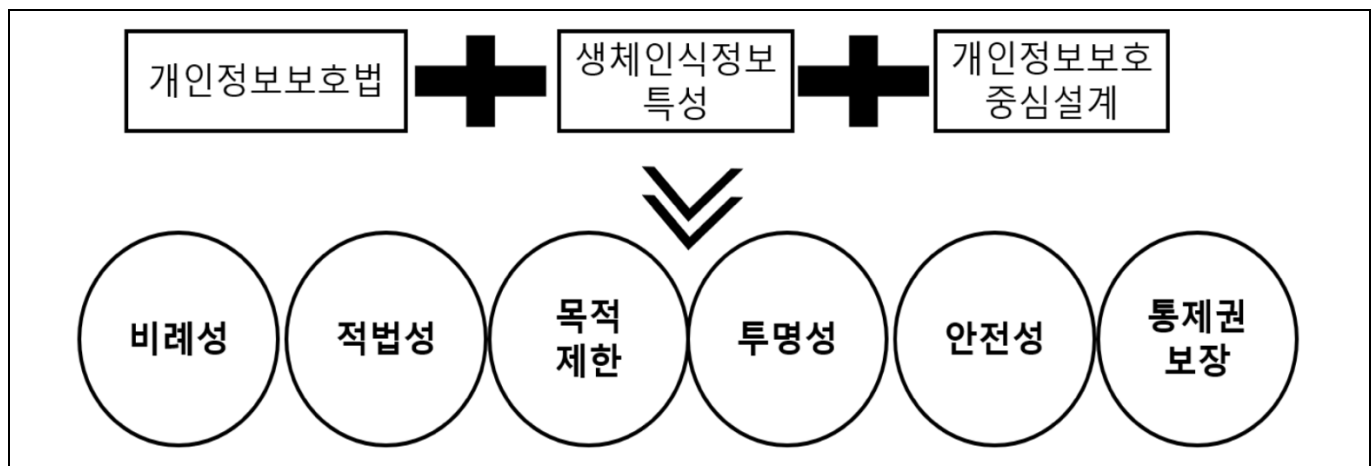
### 나. 생체인식정보의 특성

유일성, 불변성	- 생체인식정보는 그 자체로 개인을 유일하게 식별할 수 있고 변경이 불가능한 특성으로 인해 한번 유출되면 그 피해를 복구하기 어려움
민감정보 추출 가능성	- 원본정보로부터 인증·건강 등 인증·식별 목적과 무관한 민감정보가 추출될 수 있음
위변조 가능성	- 실리콘 인공지문, 캡처된 얼굴·홍채사진 등 위·변조된 생체인식정보를 이용한 해킹 사례가 지속적으로 발생하고 있음

- 「개인정보 보호법」 상의 개인정보보호 원칙(제 3 조)을 기본으로 생체인식정보의 특성, 개인정보보호 중심 설계(PbD) 원칙 등을 반영하여 생체인식정보 보호 6 대 원칙을 도출

## II. 생체인식정보 보호 6 대원칙의 배경 및 설명

### 가. 생체인식정보 보호 6 대 원칙의 배경



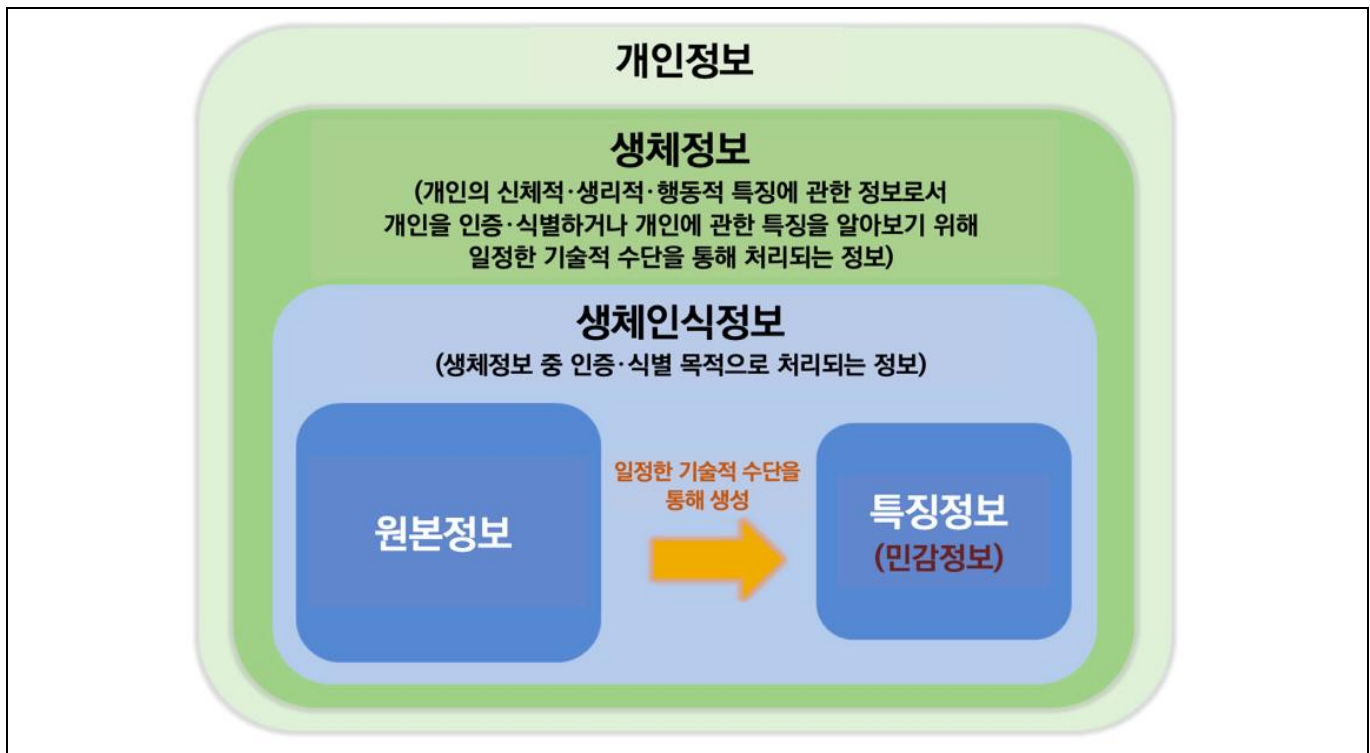
- 생체인식정보 보호의 6 대 원칙은 개인정보보호법을 바탕으로 생체인식정보의 특성과 유럽의 개인정보보호 중심설계를 기초로 만들어짐

나. 생체인식정보 보호 6 대 원칙 설명

구분	6 대 원칙	설명
법적, 윤리적 정당성	- 적법성	- 생체인식정보의 수집이용제공 등 생체정보 처리의 근거는 적법 명확 - 개인정보보호법과 관련된 법률적 근거와 정당성 필요 - 관련법 : 개인정보 보호법 제 15 조, 제 23 조
	- 비례성	- 생체인식정보 처리에 따른 편익에 비해 개인정보 침해 위험성 비교 - 위험성을 고려하여 생체인식정보의 활용 여부 판단 - 관련법 : 개인정보 보호법 제 16 조, 제 23 조
	- 목적제한	- 생체인식정보를 정보주체에게 동의 받은 인증식별만으로 사용 - 인증식별 외의 무단 사용시 법적 책임 필요 - 관련법 : 개인정보 보호법 제 18 조
정보주체 권리 중심	- 투명성	- 생체인식정보 보호에 관한 사항을 정보주체에게 알기 쉽게 공개 - 수집 및 처리과정, 보관방식 등을 정보주체에게 알려야 함 - 관련법 : 개인정보 보호법 제 17 조
	- 통제권 보장	- 정보주체가 자신의 생체인식정보를 스스로 통제할 수 있는 수단 제공 - 정보주체가 생체정보의 열람, 정정, 삭제를 요구할 수 있어야 함 - 관련법 : 개인정보 보호법 제 35 조
기술적, 관리적 보호조치	- 안정성	- 생체인식정보가 분실·도난·유출·위조·변조 또는 훼손되지 않도록 안전하게 처리하고 관리 - 관련법 : 개인정보 보호법 제 29 조

- 생체인식정보는 생체정보 및 개인정보와 구분하여 인식할 필요가 있음

III. 개인정보, 생체정보, 생체인식정보의 관계



- 개인의 신체적, 생리적, 행동적 특징에 관한 정보가 인증·식별 목적 또는 개인에 관한 특징을 알아보기 위해 기술적으로 처리되지 않는다면 일반적인 개인정보에 해당,
- 인증·식별 이외의 목적으로 개인에 관한 특징(연령·성별·감정 등)을 알아보기 위해 기술적으로 처리되는 정보는 일반적인 생체정보에 속하지만, 생체인식정보에는 해당하지 않음

“끝”

<b>06</b>	<b>SDDC(Software-Defined Data Center)</b>		
<b>문제</b>	SDDC(Software-Defined Data Center)의 기술요소 설명 및 Data Center 와 비교 설명		
<b>도메인</b>	디지털서비스	<b>난이도</b>	<b>중</b> (상/중/하)
<b>키워드</b>	SDC, SDN, SDS, 가상화, NFV, Automation, Orchestration, Autoscaling		
<b>출제배경</b>	113 회 기출문제로 재출제 가능성 및 최근 SDDC 시장 규모 성장세 가속화		
<b>참고문헌</b>	ITPE 기술사회 서브노트		
<b>출제자</b>	박서현 기술사(제 131 회 정보관리기술사 / mondaysss@naver.com)		

## I. 소프트웨어 정의 데이터센터, SDDC(Software-Defined Data Center)의 개념

### 가. SDDC 개념

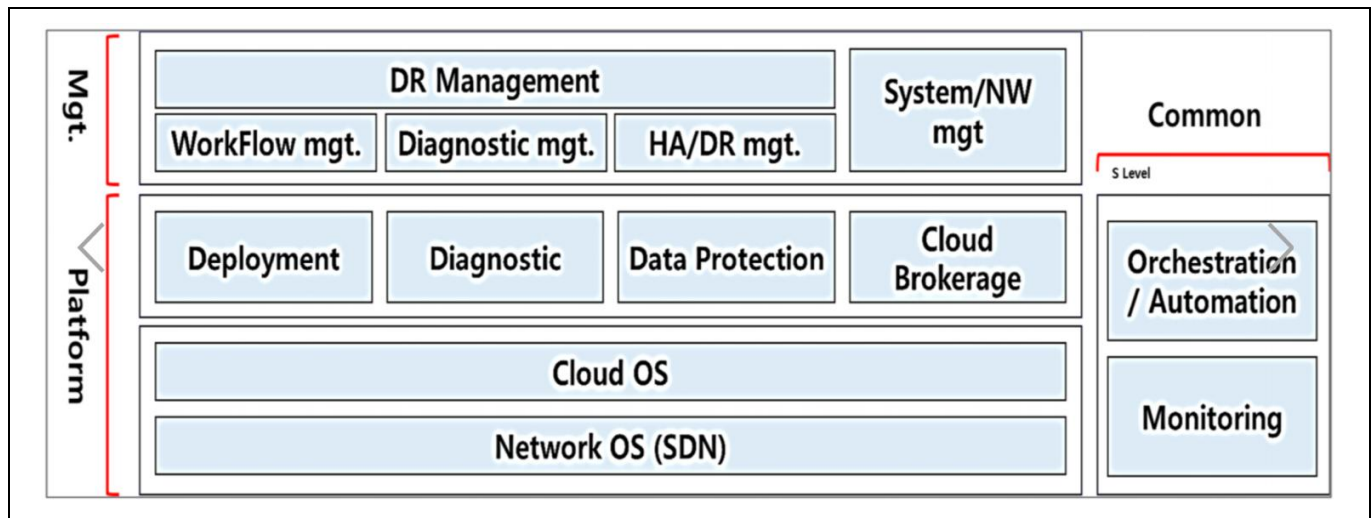
- 모든 하드웨어가 가상화되어 가상 자원의 Pool 을 구성하고, 데이터센터 전체를 운영하는 소프트웨어가 필요한 기능 및 규모에 따라 동적으로 자원을 할당, 관리, 소멸하는 역할을 수행하는 데이터센터

### 나. SDDC 특징

<b>자동화, 정책 기반 관리</b>	- 정의된 정책에 따라 IT 리소스 검색, 프로비저닝, 구성을 자동화
<b>벤더 종속성 제거</b>	- 특정 벤더의 장비에 Lock-in 되지 않고 범용적인 HW 사용 가능

## II. SDDC(Software-Defined Data Center)의 개념도 및 기술요소

### 가. SDDC(Software-Defined Data Center)의 개념도



- 데이터센터를 구성하는 시스템, 네트워크, 운영 관리 등 전 영역을 소프트웨어로 자동 통제 및 관리

### 나. SDDC(Software-Defined Data Center)의 기술요소

구분	기술 요소	설명
시스템	- 서버가상화	- 컴퓨팅 자원과 HW 를 추상화하고 논리적 분할, 통합하여 자원을 효율적으로 사용하는 기술
	- 플랫폼기술	- 컴퓨팅, 스토리지 자원을 제어하는 주요 플랫폼 이용하여 시스템 설계하는 기술

		- 오픈스택, 클라우드 스택
	- SDS(Software-Defined Storage)	- 스토리지 저장공간을 소프트웨어로 가상화하여 이기종 스토리지 간의 통합, 동적 자원할당, 데이터 보호 기술
	- SDC(Software-Defined Compute)	- 하이퍼바이저(Hypervisor)를 사용하여 하나의 물리적 서버에서 여러 개의 가상 머신(VM)이나 컨테이너를 실행 - Container : Docker, LXC/LXD, JAIL, Zones
네트워크	- SDN	- 가상화된 네트워크 자원을 개방형 네트워킹 API(Open Flow) 이용하여 통합 관리 및 제어 - Overlay N/W : Network slice(VxLAN)
	- NFV	- 스위치, 라우터, 방화벽, 로드밸런서 등 네트워크의 주요 기능을 x86 서버 기반으로 가상화하고 SW 로 관리하는 기술
운영관리	- Orchestration	- 자원관리, SW 프로비저닝을 통해 SDDC 구성요소 오케스트레이션
	- Autoscaling	- 자원 사용률에 따라 동적으로 시스템 자원 증설 및 회수하여 효율적인 IT 투자 및 활용을 구현하는 기술 - Monitoring, Metering

- SDDC 는 가상화 기술이 핵심이며, Vendor Lock-in 탈피 및 오픈소스 활용으로 인한 라이선스비 절감 가능

### III. SDDC 와 Data Center 의 비교

비교 항목	SDDC	Data Center
하드웨어	- 범용 x86 장비	- 특정 제조사 장비
API	- 오픈소스 API(ex. 오픈스택 Restful)	- 각 제조사 API
초기투자	- 단계별 투자로 낮음	- 높음
비용	- 비교적 낮음	- 비교적 높음(제조사에 따라 다양한 비용)
효율성	- 높음	- 낮음

- Data Center 에 비해 SDDC 는 데이터센터 공간과 유지보수 비용, 인건비 등 간접비 비용 절감 가능

“끝”



<b>07</b>	<b>임베디드 파이낸스(Embedded Finance)</b>		
<b>문제</b>	임베디드 파이낸스(Embedded Finance)		
<b>도메인</b>	디지털서비스	<b>난이도</b>	<b>중</b> (상/중/하)
<b>키워드</b>	BaaS, OpenAPI, 클라우드 네이티브, MSA, AI/ML		
<b>출제배경</b>	금융 서비스 관련 IT 기술의 최신 트렌드 이해		
<b>참고문헌</b>	<a href="https://b2broker.com/ko/news/what-is-embedded-finance-and-how-does-it-shape-the-modern-financial-industry/">https://b2broker.com/ko/news/what-is-embedded-finance-and-how-does-it-shape-the-modern-financial-industry/</a>		
<b>출제자</b>	이다연 기술사(제 135 회 정보관리기술사 / dlekusz@naver.com)		

## I. 임베디드 파이낸스(Embedded Finance)의 개요

### 가. 임베디드 파이낸스(Embedded Finance)의 정의

- 비 금융 기업의 서비스 또는 제품에 금융 기능(결제, 대출, 보험 등)을 내장(Embedded)시켜 판매/제공하는 기술 또는 비즈니스 모델

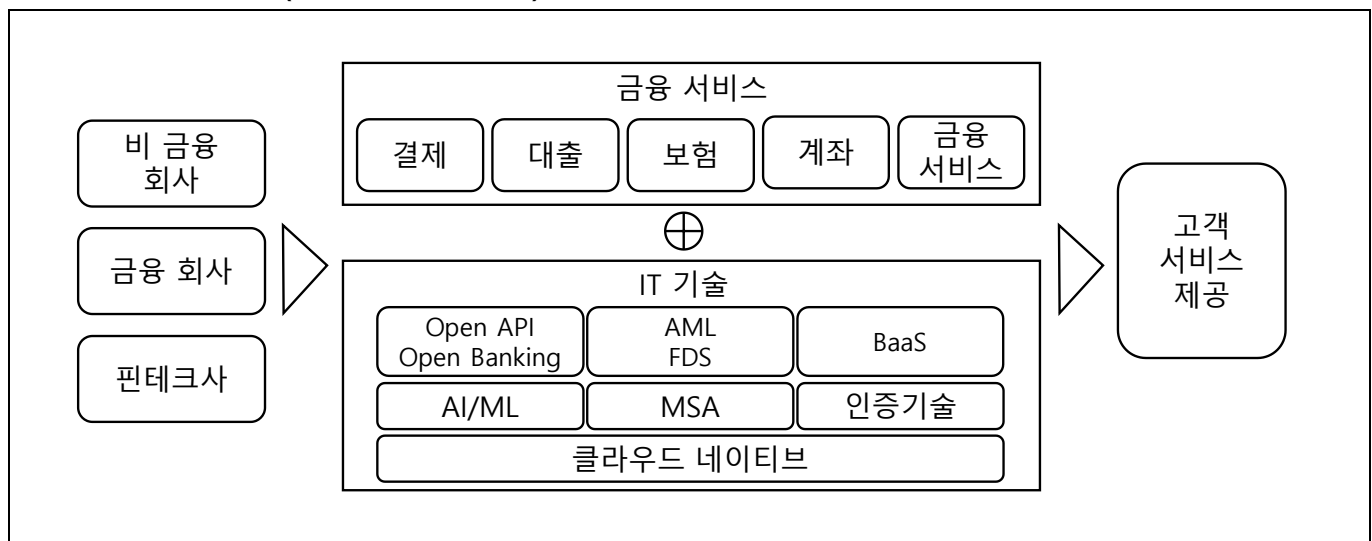
### 나. 임베디드 파이낸스(Embedded Finance)의 등장 배경

<b>플랫폼 경제 확산</b>	- 금융 서비스가 플랫폼 내부에 구축되어 부가가치 창출 추구
<b>API 기술 발전</b>	- 결제, KYC, AML 등의 서비스가 API 로 서비스되어 쉽게 탑재 가능
<b>고객 경험 중심 경쟁</b>	- 신속하고 간편하게 금융 서비스를 사용하고자 하는 사용자 요구 심화
<b>AI/ML 기반 모델 발전</b>	- 거래/이용 데이터 기반으로 고객 맞춤형 서비스 분석 가능

- IT 기술이 발전함에 따라 비 금융 서비스에 금융 서비스를 자연스럽게 결합시켜 사용자가 간편하게 금융서비스를 이용할 수 있도록 하는 기술로 발전

## II. 임베디드 파이낸스(Embedded Finance)의 개념도와 기술요소

### 가. 임베디드 파이낸스(Embedded Finance)의 개념도



- IT 기술과 기존 금융서비스를 결합하여 사용자가 서비스 또는 상품을 구매 할 때 금융 서비스를 포함하여 제공

## 나. 임베디드 파이낸스(Embedded Finance)의 기술요소

구분	기술요소	설명
금융 서비스 연동 기술	- Open API, Open Banking, FAPI	- 금융 기능을 외부 플랫폼에 제공하도록 하는 API
	- BaaS(Bank-as-a-Service)	- 금융 기능(결제, 대출 등)을 클라우드 형식으로 제공하는 서비스
	- OAuth2.0, OpenID	- 외부 플랫폼에서의 사용자 신원 인증
신원/규제	- eKYC, 생체인증	- 신분증, 안면인식, 지문인식 등으로 본인 인증 기능 제공
	- AML/FDS	- 자금세탁 방지, 이상금융탐지 통한 금융 규제 대응
클라우드 및 보안	- AI/ML	- 고객 맞춤 서비스 분석, 리스크 탐지
	- 클라우드 네이티브	- 서버리스 기반 확장성 및 탄력성 제공
	- MSA	- 금융 기능을 서비스 단위로 분리하여 관리
	- 암호화, 비식별화	- 개인정보, 민감정보 보호 위한 기술

- 금융 서비스를 제공하는 API 기반으로 하여 비 금융 서비스와 통합하여 제품/서비스 제공 흐름의 자동화를 구현하여 **고객 여정 중심의 경험을 유도**하는 것이 핵심 요소

## III. 임베디드 파이낸스(Embedded Finance)의 활용사례

구분	기업	사례
국내	- NAVER Financial	- 대안 신용평가 시스템 기반 신용대출
	- kakao	- 결제, 은행, 증권, 디지털 손해보험 서비스 제공
	- 삼성생명	- 플랫폼사와 연계하여 서비스 구매 시 보험 제공
국외	- Tesla	- 실시간 정보 기반 사고위험 및 수리비용 예측 서비스
	- Google Map	- 네비게이션에서 팝업창 통한 결제 서비스 제공
	- Lambda	- 수업료 결제 시 할부, 미래소득 발생 따른 지불계약 체결

- **고객경험(UX) 향상과 함께 플랫폼 Lock-in 효과 통한 충성고객 창출, 비즈니스 효율성/확장성 증가**와 같은 이유로 기업의 임베디드 파이낸스 서비스 제공 증가

“끝”

08	에이전틱 코딩(Agentic Coding)		
문제	에이전틱 코딩(Agentic Coding)		
도메인	인공지능	난이도	중 (상/중/하)
키워드	코딩 에이전트, 멀티 협업 도구, 지능형 의사결정, AI 오케스트레이션		
출제배경	다양한 AI 기반 개발 기술의 등장 따른 IT 기술 최신 트렌드로서 출제 가능		
참고문헌	<a href="https://hunking.tistory.com/1717">https://hunking.tistory.com/1717</a> <a href="https://digitalbourgeois.tistory.com/1445">https://digitalbourgeois.tistory.com/1445</a>		
출제자	이다연 기술사(제 135 회 정보관리기술사 / dlekusz@naver.com)		

## I. AI 를 통한 개발 효율화 실현, 에이전틱 코딩(Agentic Coding)의 개요

### 가. 에이전틱 코딩의 정의

- 소프트웨어 개발 SDLC 의 일부 또는 전체 단계를 AI 에이전트에 위임하여 개발 목표를 세우고 단계별 개발 활동을 수행하는 패러다임

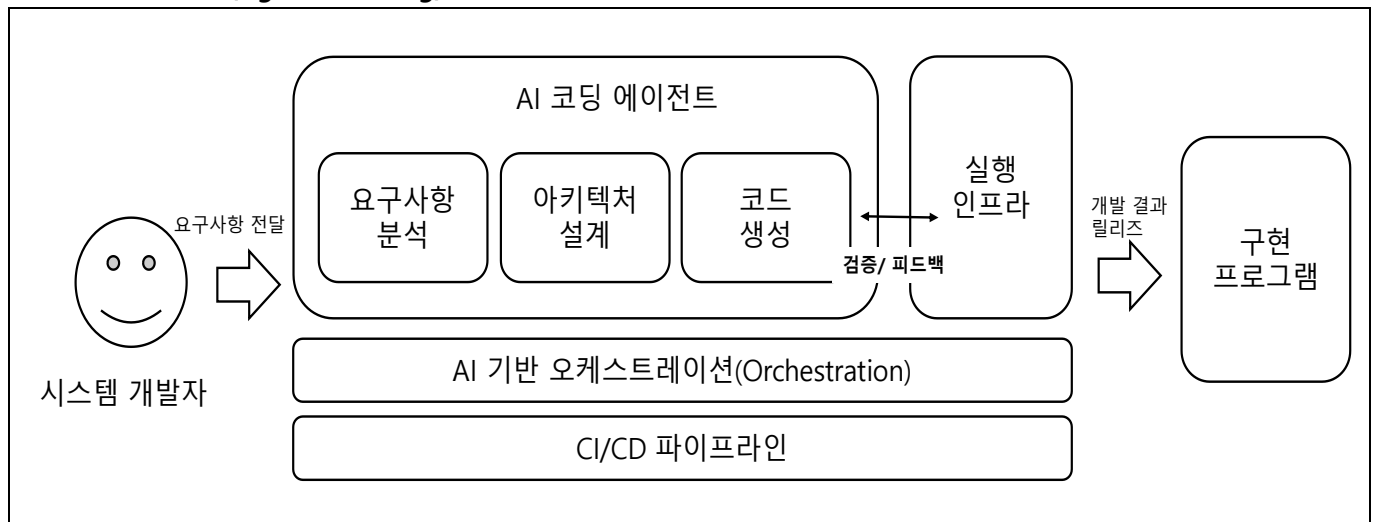
### 나. 에이전틱 코딩의 등장 배경

개발 생산성 향상 요구	- 복잡한 로직 단시간 구현 가능, 잠재적 오류 예측 가능하여 오류율 감소
SW 복잡도 증가	- 클라우드 네이티브, MSA 와 같은 아키텍처 증가로 구현 난이도 상승
DevOps, MLOps 고도화	- CI/CD 파이프라인과 연동 가능하여 개발 범위 확대

- 소프트웨어 개발의 복잡도 상승과 생산성 및 품질 관련 요구사항 증대로 인해 AI 를 활용한 개발 패러다임 등장

## II. 에이전틱 코딩(Agentic Coding)의 개념도와 기술요소

### 가. 에이전틱 코딩(Agentic Coding)의 개념도



- 사용자의 요구사항을 분석하여 적합한 기술 및 아키텍처 선정 후 실제 코드 작성과 테스트까지 수행하는 파이프라인으로 구성

## 나. 에이전틱 코딩(Agentic Coding)의 기술요소

구분	핵심 기술	설명
에이전트 도구	- AI 코딩 에이전트	- 지능형 의사결정 가능 - 요구사항 분석, 문서작성과 같은 개발 전반 활동 수행
	- 멀티 에이전트 협업 도구	- 프론트엔드/백엔드/테스트/성능/보안과 같은 각 개발 단계의 역할을 수행하는 에이전트
개발도구	- Repository	- 개발 단계별 산출물(작성코드, 문서, 테스트 결과) 저장
	- 실행 인프라	- 실제 개발환경에 명령 실행 - 컨테이너 기반 멀티 프로세스 작업 수행
	- CI/CD 파이프라인	- 개발 사이클 자동화 수행 - 변경사항, 테스트 결과, 운영 자동 배포
	- 품질/보안 관리 체계	- 취약점 분석, 안전한 코드 생성 및 관리 - 오류 시 자동 롤백 수행
	- AI 기반 관리 오케스트레이션	- 개발 진행 검증 및 통제 - 단계별 산출물 검토

- AI 코딩 에이전트 및 다양한 기능을 수행하는 협업 에이전트 도구 통한 코딩의 전반적인 프로세스 자동화 수행 가능하도록 구축하는 것이 중요

## III. 에이전틱 코딩(Agentic Coding) 도입 시 고려사항

고려사항	설명
고가의 비용	- 고가의 AI 코딩 에이전트 도입 인한 개발 비용 상승 초래
병렬 관리 어려움	- DB, API 등의 실행 도구에 병렬 접근 시 시스템 부하 발생 가능성 및 동시 접근 인한 데이터 정합성 문제 발생 가능성 존재
에코시스템 의존성 증가	- AI 시스템의 빠른 발전 인한 협업 도구의 잦은 교체 발생 가능성

- 복잡한 개발 아키텍처를 효율적으로 구현 가능하나 AI 이용 시 발생하는 비용 및 의존성 문제 해결 필요

“끝”

09	데이터베이스 리플리케이션(Replication)과 쿼리 오프로딩(Query-off Loading)		
문제	데이터베이스 리플리케이션(Replication)과 쿼리 오프로딩(Query-off Loading)		
도메인	데이터베이스	난이도	하 (상/중/하)
키워드	Master DB, Slave DB, Update/Read 트랜잭션 분리, CDC		
출제배경	데이터베이스 성능 관리 위한 기본적인 지식 확인		
참고문헌	ITPE 기술사회 자료		
출제자	이다연 기술사(제 135 회 정보관리기술사 / dlekduz@naver.com)		

### I. 데이터 복제 통한 일관성과 가용성 확보, 데이터베이스 리플리케이션 설명

구분	설명	
개념	- 데이터베이스의 일관성 및 가용성을 유지하기 위해 Master/Slave 로 나누어 데이터를 저장하는 방식	
개념도	<pre> graph TD     App[Application] -- "Insert Update Delete" --&gt; Master[Master]     App -- "Select" --&gt; Slave[Slave]     Master -- "Replication" --&gt; Slave </pre>	
목적	- 데이터 백업 통한 일관성, 가용성 향상	
구조	- Master-Slave	<ul style="list-style-type: none"> <li>- Master : 변경 관리, Slave 노드 전달</li> <li>- Slave : 읽기전용</li> <li>- 장점 : 읽기 요청 부하 분산</li> <li>- 단점 : 쓰기 성능 한계</li> </ul>
	- Peer-to-Peer	<ul style="list-style-type: none"> <li>- Peer : 변경 사항 서로 전달, 동등 관계</li> <li>- 장점 : 단일 장애 발생점(SPOF) 없음</li> <li>- 단점 : 구조 복잡, 충돌 가능성</li> </ul>
구현기법	- 로그 기반(Log-based) 복제	<ul style="list-style-type: none"> <li>- 트랜잭션 로그 이용 변경 사항 전달</li> <li>- 데이터 일관성 향상, 실시간 복제 가능</li> </ul>
	- 상태 기반(State-based) 복제	<ul style="list-style-type: none"> <li>- 스냅샷, Master-Slave, 다중 마스터 방식</li> <li>- 데이터 상태 기반 데이터 복제</li> </ul>

- 리플리케이션 기술 적용한 트랜잭션 처리 성능 향상 위한 쿼리 오프로딩 구현 가능

## II. 데이터베이스 처리 성능 향상, 쿼리 오프로딩 설명

구분	설명	
개념	- DB 트랜잭션을 <b>Update/Read</b> 로 분리하여 DB 처리량을 향상시키는 기법	
개념도	<pre> graph TD     App[Application] --&gt; CP_Update[Connection Pool Update]     App --&gt; CP_Read[Connection Pool Read]     CP_Update --&gt; Master[Master DB]     Master -- "Replication CDC" --&gt; Staging[Staging DB]     Staging -- "Replication CDC" --&gt; Slave[Slave DB]     Slave --&gt; Slave     </pre>	
목적	- DB 처리량을 증가시켜 전체적인 성능 향상	
구성요소	- Master DB	- Create/Update/Delete 트랜잭션 수행
	- Slave DB	- Read 트랜잭션만 수행 - N 개 DB 로 구성 - 장애 시 HA 제공
	- Staging DB	- Master->Slave 복제 위한 경유지 역할
주요기술	- Replication(CDC)	- Master 에서 Slave 로 데이터 복제 - Back Log 이용
	- Load Balancing	- Slave DB 부하 분산

- 리플리케이션과 쿼리 오프로딩은 DB 성능 향상을 위한 기법이나 구현방식 및 처리 방식에서의 차이 존재

## III. 리플리케이션과 쿼리 오프로딩 비교

비교 항목	리플리케이션	쿼리 오프로딩
목적	- 일관성 및 가용성 향상	- 데이터베이스 처리 성능 향상
처리방식	- 다수 DB 에 동일 데이터 복제	- Update/Read 트랜잭션 분리
구현주체	- DBMS 자체 기술	- Oracle OGG, Share Flex 등 SW
복제방식	- 비동기/동기식 복제 - 로그기반/상태기반 복제	- CDC(Change Data Capture) - Staging DB 이용 연계
공통점	- 데이터베이스 성능 향상	

- 기타 데이터베이스 성능 향상을 위해 샤딩, 파티셔닝, 클러스터링 적용 가능

“끝”

10	프롬프트 인젝션(Prompt Injection)		
문제	프롬프트 인젝션(Prompt Injection)		
도메인	정보보안	난이도	중 (상/중/하)
키워드	직접 프롬프트 인젝션: 이중 문자, 가상화, 난독화, 페이로드 분할, 적대적 접미사, 지시 조작 간접 프롬프트 인젝션: 능동적 주입, 수동적 주입, 사용자 주도 주입, 가상 프롬프트 주입, 다중체인 프롬프트 주입		
출제배경	가트너 2026 10대 전략 기술 'AI 보안 플랫폼' 연관 토픽		
참고문헌	생성형 AI 공격기법 및 대응기술 동향(주기동, 2025.09) 2024년 상반기 EQST 리포트로 알아보는 보안 트렌드(SK윌더스, 2024.07.05) ITPE 기술사회 자료		
출제자	소민호 기술사(제 119회 정보관리기술사 / mhsope@naver.com)		

## I. OWASP top 10 for LLM Application 취약점, 프롬프트 인젝션(Prompt Injection)의 개요

### 가. 프롬프트 인젝션(Prompt Injection)의 정의

- LLM 모델의 응답을 조작하기 위해, 공격자가 프롬프트에 정교하게 조작된 입력 값을 주입하여 민감 데이터를 유출하는 공격기법

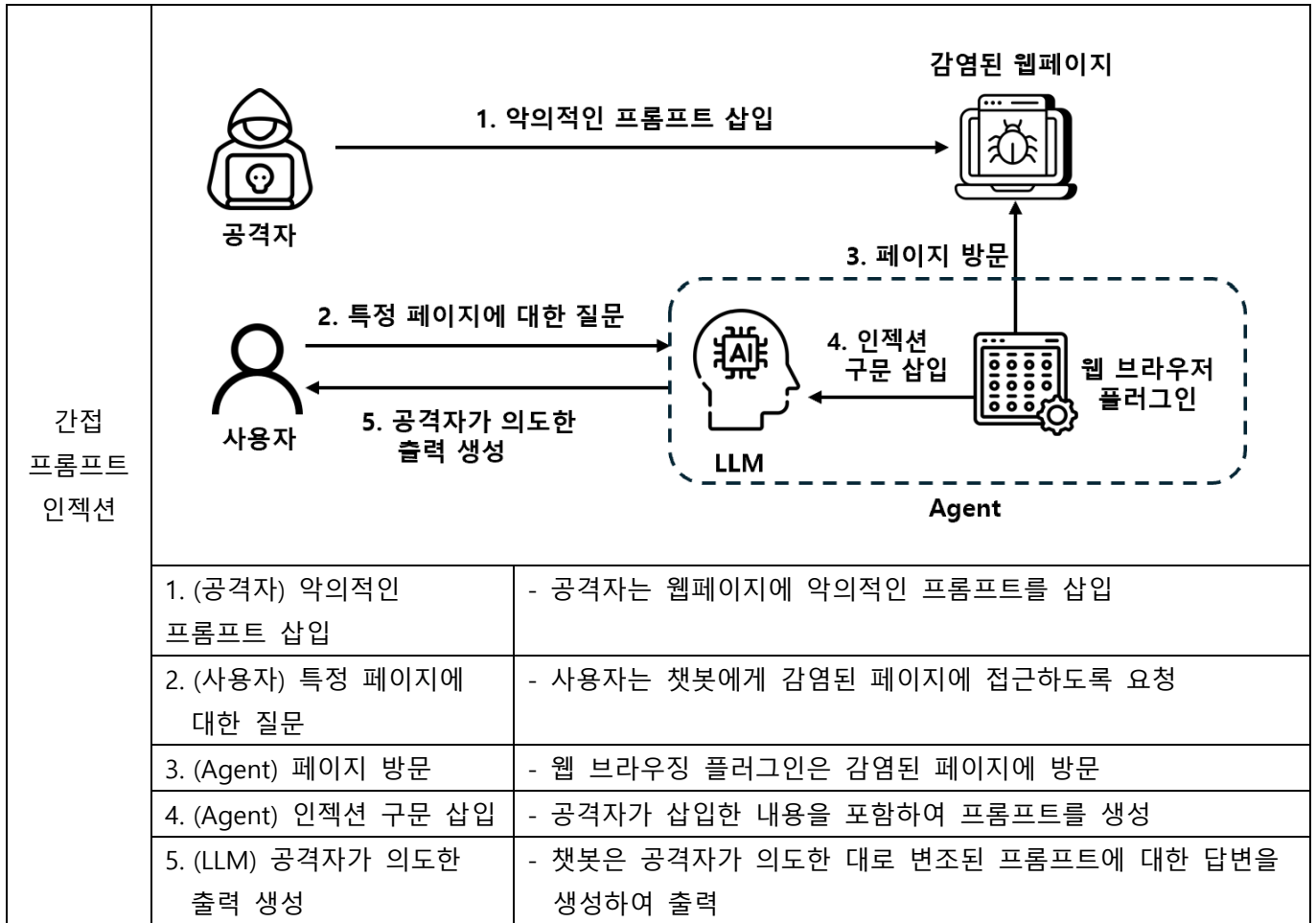
### 나. 프롬프트 인젝션의 특징

악의적인 입력 조작	- 공격자가 AI 모델이 잘못된 명령을 수행하도록 입력을 교묘하게 조작
모호성 활용	- AI 모델이 의도하지 않은 방식으로 해석하도록 입력을 설계하여 혼란을 초래
자동화 가능성	- 반복적으로 여러 AI 시스템을 대상으로 자동화된 방식으로 수행될 수 있음

## II. 프롬프트 인젝션(Prompt Injection)의 공격 시나리오 및 유형

### 가. 프롬프트 인젝션 공격 시나리오

구분	공격 시나리오	
직접 프롬프트 인젝션	<pre> graph LR     A[공격자] -- "1. 악의적인 질문 입력" --&gt; B[암호화 변환]     B -- "2. 변환된 질문 입력" --&gt; C[챗봇]     C -- "3. LLM 결과 출력" --&gt; A </pre>	
	1. 악의적인 질문 입력	- 공격자는 악의적인 질문을 입력 ROT13 암호화를 통해 변환
	2. 변환된 질문 입력	- 공격자는 암호화된 질문을 LLM(챗봇)에게 질문
	3. LLM 결과 출력	- LLM은 암호화된 질문을 이해하여 악성 답변을 생성하여 답변



- 공격자가 사용자 입력·웹 콘텐츠·문서 등에 숨겨둔 지시문을 LLM이 그대로 실행하도록 유도해 원래 의도와 다른 명령 수행이나 정보 유출을 발생시키는 공격 방식

#### 나. 프롬프트 인젝션 공격 유형

구분	유형	설명
직접 프롬프트 인젝션	- 악의적인 사용자가 프롬프트를 조작해 의도하지 않은 모델 동작을 유도	
	이중 문자	- 정상 응답 + 규칙 무시 응답을 동시에 요구해 제약을 우회하는 이중 응답 기반 Jailbreak 기법
	가상화	- 개발자 모드, 가상 머신 모드 등 무제한 환경처럼 가장해 모델 제약을 우회
	난독화	- 악의적 지시를 Base64·특수문자 등으로 인코딩해 필터를 회피
	페이로드 분할	- 악성 지시를 여러 프롬프트로 나눠 보내고 결합될 때 악의적으로 작동하도록 하는 기법
	적대적 접미사	- 무작위 텍스트를 접미사로 붙여 LLM의 안전 규칙을 흔들고 악성 응답을 유도.
	지시 조작	- 내부 시스템 프롬프트 노출 또는 규칙 무시를 요구하여 안전 장치 자체를 무력화.
간접 프롬프트	- 입력 콘텐츠가 외부 소스와 연동되어 연쇄적인 모델 동작의 변경을 유도	



인젝션	능동적 주입	- 이메일·문서·업무툴 등 <b>LLM 확장 기능이 자동 처리하는 입력 경로</b> 에 악성 프롬프트를 삽입해 LLM이 능동적으로 이를 실행하도록 유도
	수동적 주입	- 웹페이지·공개 문서 등에 <b>악성 텍스트를 미리 배치</b> 해, LLM이 이를 읽을 때 조작되도록 하는 방식
	사용자 주도 주입	- 사회공학을 이용해 <b>사용자가 직접 악성 프롬프트를 복사·붙여넣도록 유도</b> 하는 기법
	가상 프롬프트 주입	- 지시사항 튜닝 데이터나 시나리오를 <b>은밀히 조작</b> 해, 모델이 특정 상황에서 공격자의 지시를 수행하도록 만드는 방식
	다중체인 프롬프트 주입	- 여러 LLM 호출이 연결된 <b>체인 구조의 중간 단계</b> 를 악용해, 하나의 악성 프롬프트가 전체 체인에 전파되도록 유도

- 프롬프트 인젝션 공격은 직접 입력 조작부터 외부 콘텐츠 주입까지 다양한 방식으로 LLM의 의사결정 흐름을 왜곡해 원래 의도와 다른 동작을 수행하게 만드는 공격

### III. 프롬프트 인젝션 공격 대응방안

구분	대응방안	설명
입력 단계 보안	<b>입력 검증</b>	- 사용자 입력을 사전 검사하여 악성 프롬프트·명령 삽입을 필터링
	<b>필터링 적용</b>	- 금치어·민감 키워드·NLP 기반 의미 분석으로 위험 입력 차단
	<b>콘텐츠 분리</b>	- 외부 입력을 명확히 태그 또는 메타데이터로 분리하여 시스템 지침과 혼동 방지
모델 동작 제어	<b>동작 제한</b>	- 모델의 역할, 권한 범위, 무시 금지 지침(Role Prompting)을 명확히 설정
	<b>최소 권한</b>	- 모델이 수행할 수 있는 기능을 최소 권한으로 제한(외부 자원 접근 차단 등)
	<b>컨텍스트 격리</b>	- 시스템 프롬프트, 개발자 지침, 사용자 입력을 논리적으로 분리하여 영향 최소화
출력 보안	<b>출력 제한</b>	- 생성된 응답이 민감 정보, 내부 지침, 시스템 명령을 포함하지 않도록 제한
	<b>출력 검증</b>	- 응답 형식 지정, 인용 요구, 코드 정적 검사 등으로 결과물 구조적 검증
운영·관리 보안	<b>사람 승인 프로세스</b>	- 고위험 작업(삭제, 업데이트, 외부 호출 등)은 반드시 사람 검증 후 실행
	<b>적대적 테스트</b>	- 공격 시나리오 기반 프롬프트 침투 테스트·방어 시뮬레이션 정기 운영
데이터·모델 보안	<b>데이터 신뢰성 확보</b>	- 신뢰할 수 있는 데이터 사용 및 데이터 무결성 검증
	<b>모델 보안 강화</b>	- 모델 업데이트, 취약점 패치, 안전성 관련 튜닝·가드레일 적용
사용자 측 보안	<b>사용자 교육</b>	- 사용자에게 위험 프롬프트, 보안 정책, 안전 가이드라인을 교육
환경 격리	<b>샌드박스</b>	- 모델이 생성한 코드·명령을 실제 시스템과 분리된 안전 환경에

		서만 실행
요청 처리 보안	프록시 기반 실행	- 모델이 외부 API·파일·DB에 직접 접근하지 못하도록 중간 프록시 적용
정책 기반 보호	안전 정책 엔진	- 모델의 요청/응답을 사전 정의된 정책 기반으로 승인·차단·치환
로깅·모니터링	감사 로그 기록	- 비정상 프롬프트, 우회 시도, 위험 응답에 대한 로그 기록 및 분석
자동 탐지	이상행위 탐지 (LLM IDS)	- 공격 패턴·Role 공격·탈출 프롬프트 등을 자동 탐지하는 ML 기반 시스템

- 이외에도 LLM 모델의 허용 가능한 행동을 정의하고 준수를 강제하는 엄격한 정책 구현 등을 할 수 있음

“끝”

11	멀티 에이전트 시스템(Multi Agent System)		
문제	멀티 에이전트 시스템(Multi Agent System)		
도메인	인공지능	난이도	중 (상/중/하)
키워드	A2A, MCP, CoT, Multi-Agent Task Scheduler, Supervisor Agent, Worker/Expert Agent, Expert Agent, MSA, 분산 합의		
출제배경	가트너 2026 10대 전략 기술 '다중 에이전트 시스템' 발표에 따른 출제예상		
참고문헌	ITPE 기술사회 서브노트, <a href="#">Agentic AI: Single vs Multi-Agent Systems   Towards Data Science</a>		
출제자	소민호 기술사(제 119회 정보관리기술사 / mhsopo@naver.com)		

## I. 자율적 분산지능을 통한 문제 해결 프레임워크, 멀티 에이전트 시스템(Multi Agent System)의 개요

### 가. 멀티 에이전트 시스템(Multi Agent System)의 개념

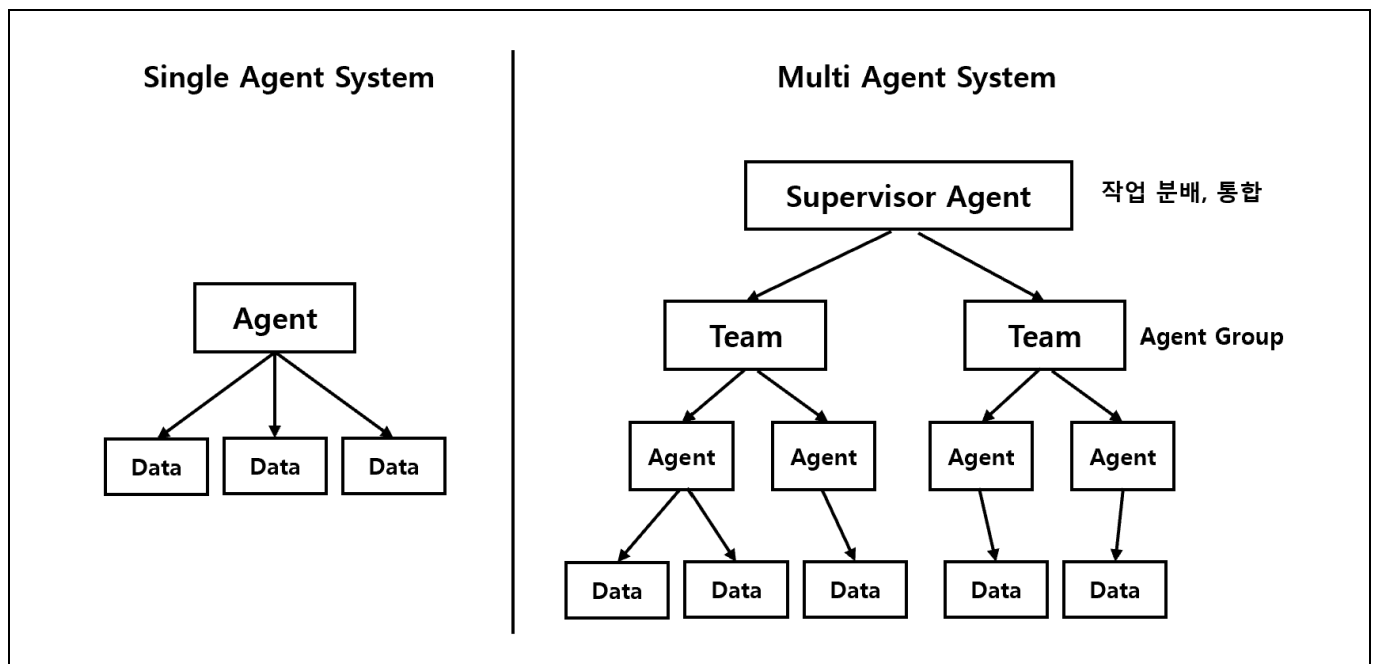
- 여러 개의 자율적·지능적 에이전트가 서로 협력·조정·경쟁하며 공동의 목표를 달성하는 분산 인공지능 시스템

### 나. 멀티 에이전트 시스템(Multi Agent System)의 특징

자율성	- 각 에이전트가 독립적으로 추론·의사결정·행동을 수행하여 외부 제어 없이 작동 가능
협력성	- 에이전트 간 역할 분담, 정보 공유, 공동 문제 해결을 통해 전체 효율을 극대화
분산성	- 중앙 집중이 아닌 분산된 구조로 작동하여 확장성·내결함성·지연 감소를 확보
적응성	- 환경 변화·입력 변화에 따라 에이전트가 스스로 학습하고 전략을 조정

## II. MAS (Multi Agent System)의 개념도와 기술요소

### 가. MAS (Multi Agent System)의 개념도



- MAS는 여러 개의 독립적 에이전트가 상호 협력·경쟁하며 복잡한 문제를 분산적으로 해결하는 지능형 시스템

#### 나. MAS (Multi Agent System)의 기술요소

구분	기술요소	설명
통신 (Communication)	A2A(Agent-to-Agent)	- 에이전트 간 통신 프로토콜
	MCP(Model Context Protocol)	- 외부 시스템·DB·API·툴을 안전하게 호출하고 컨텍스트를 공유하는 표준 프로토콜
	메시지/이벤트 기반 통신 (FIPA ACL, MQTT/Kafka)	- 상태·결과·명령을 비동기 전달하여 대규모 MAS 확장성과 안정성 확보
추론·계획 (Reasoning & Planning)	체인-오브-소트(CoT), 트리-오브-소트(ToT)	- LLM 기반 단계적 사고, 분기 reasoning을 통해 문제 해결 정확도 향상
	작업 계획(Task Planning)	- 복잡한 목표를 하위 작업으로 분해하고 실행 순서를 조정하는 계획 생성
	의사결정·전략 생성	- 목표 달성을 위한 최적의 행동 정책을 선택(LLM 또는 규칙·RL 기반)
	Multi-Agent Task Scheduler	- 역할 분담과 협업·경쟁 전략을 통해 복잡한 작업을 효율적으로 수행
	Multi-Agent Planning/ Negotiation	- 여러 에이전트가 협상·계획을 공유하며 최적의 공동 목표를 달성하도록 지원
역할 구조(Role Composition)	리더 에이전트 (Supervisor Agent)	- 전체 목표 설정, 서브 에이전트 조정, 결과 평가·취합을 담당하는 상위 역할
	서브 에이전트 (Worker/Expert Agent)	- 각 전문 분야(코딩·보안·검색·요약 등)를 담당하여 세부 작업을 수행
	역할 기반 오케스트레이션 (Role Assignment)	- 문제 유형에 따라 적절한 에이전트를 동적으로 배정하거나 해제
전문가 모델·결과 통합	전문가 에이전트 (Expert Agents)	- 코딩, 검색, 수학, 보안, 번역 등 분야별 전문 기능 제공
	결과 통합(Ensemble/Result Aggregation)	- 여러 전문가의 출력 결과를 비교·평가·통합하여 최종 해답 생성
	메타 에이전트(Meta- Reasoner)	- 전문가들의 reasoning 품질을 점검하고 의사결정 오류를 수정
지능·학습	BDI 모델	- 신념·욕구·의도를 기반으로 행동을 선택하는 전통 MAS 의사결정 모델
	멀티 에이전트 강화학습 (MARL)	- 협력/경쟁 상황에서 정책을 학습하여 최적 전략 도출
	LLM + 외부 Tool 결합	- LLM 추론력 + 검색·계산·API 호출 등 툴 기반 문제해결 능력 결합
구조·아키텍처	에이전트 오케스트레이션	- 다중 에이전트의 역할·흐름·의사결정·상태를 통합 관리

	분산 MAS 플랫폼 (JADE, SPADE)	- 전통적 MAS의 통신·상태 관리·작업 분배 기능 제공
	서비스 분해(MSA/SOA 기반)	- 에이전트 기능을 독립 서비스화하여 확장성·장애 격리 구현
보안·신뢰	인증·인가(AuthN/AuthZ)	- 에이전트·도구 호출·메시지 위변조 방지 및 권한 기반 접근 제어
	분산 합의(PBFT, RAFT)	- 여러 에이전트가 일관된 상태·결정에 도달하도록 보장
	신뢰도 기반 상호작용 (Reputation System)	- 에이전트의 신뢰 점수 기반 협력/작업 배정 여부 결정

- MAS의 기술요소는 에이전트 간 통신, 추론·계획, 역할 구조, 전문가 모델 통합, 학습·보안 등 다수 에이전트가 협력해 복잡한 문제를 해결

### III. MAS (Multi Agent System)의 유형

유형	세부유형	설명
Independent (독립형)	이산형(Discrete)	- 에이전트 간 상호작용 없이 독립적으로 개별 작업을 수행하는 방식
	자발적 협력형 (Emergent Cooperation)	- 명시적 규칙 없이 독립적으로 행동하나, 환경 상호작용 속에서 자연스럽게 협력 행동이 발생
Cooperative (협력형)	통신형(Communicating)	- 에이전트가 메시지·정보를 주고받으며 협력하는 형태
	숙고형(Deliberative)	- 계획·추론 기반 협업을 수행
	협상형(Negotiating)	- 협상·합의를 통해 최적 행동을 결정
	비통신형 (Non-communicating)	- 직접 통신은 없지만, 행동 관찰·환경 변화를 통해 간접적으로 협력

- MAS의 유형은 독립적으로 행동하는 방식부터 통신·협상 기반으로 협력하는 방식까지 에이전트 간 상호작용 정도에 따라 구분

“끝”

12	하이퍼스케일 데이터센터(Hyperscale Data Center)		
문제	하이퍼스케일 데이터센터(Hyperscale Data Center)		
도메인	디지털서비스	난이도	중 (상/중/하)
키워드	전력 인프라(UPS·PDU), 냉각 시스템, 랙·케이지 인프라, 서버(범용·GPU), 스토리지(NVMe·분산스토리지), 네트워크 스위치(ToR/Spine), 가상화 플랫폼, 컨테이너 플랫폼(Kubernetes), SDN/NFV 인프라, 접속제어·IAM, 모니터링/관제 시스템, 재해복구(DR) 체계		
출제배경	최근 정부에서 하이퍼스케일 데이터센터 구축 전략 발표에 따른 출제 예상		
참고문헌	ITPE 소원반 서브노트		
출제자	소민호 기술사(제 119회 정보관리기술사 / mhsope@naver.com)		

### I. 하이퍼스케일 데이터센터의 개념 및 기술적 특징

구분	설명	
개념	<ul style="list-style-type: none"> <li>- 수십만 대 이상의 서버, 스토리지, 네트워크 장비를 표준화·자동화 구조로 통합 운영하는 초대형 데이터센터</li> <li>- 클라우드 서비스, AI 모델 학습·추론, 빅데이터 분석 등 대규모 워크로드를 효율적으로 처리하기 위해 설계된 인프라</li> </ul>	
특징	대규모 수평 확장성	- 서버·랙을 필요 시 즉시 확장/축소하여 대규모 트래픽 처리 가능
	표준화·모듈화 인프라	- 장비·전력·냉각을 모듈 형태로 구성해 빠른 구축과 교체 지원
	소프트웨어 정의 운영	- 인프라를 소프트웨어로 제어해 자동화·최적화와 정책 기반 운영 가능
	고효율·고가용성 설계	- 낮은 PUE, 이중화, 장애 도메인 분리로 무중단·안정적 서비스 제공

- 하이퍼스케일 데이터센터는 표준화·자동화·확장성 기반으로 초대형 워크로드를 안정적으로 처리

### II. 하이퍼스케일 데이터센터 구성요소 및 주요 기술요소

#### 가. 하이퍼스케일 데이터센터 구성요소

구분	구성요소	설명
인프라 시설	전력 인프라 (UPS·PDU)	- 대규모 서버 전력 공급 안정성을 위해 UPS·PDU 구성, 이중화 전력 설계 적용
	냉각 시스템	- 고밀도 랙의 발열 해소, 수랭식·액침냉각 등 고효율 냉각 방식 도입
	랙·케이지 인프라	- 고집적 서버 배치를 위한 표준랙·고밀도랙, 물리적 격리 제공
컴퓨팅 자원	서버(범용·GPU)	- 대규모 처리 용량 확보를 위한 x86, ARM, GPU, AI 서버 구성
	스토리지 (NVMe·분산스토리지)	- NVMe 기반 초고속 I/O, Ceph/HDFS 등 분산 스토리지 구조 활용
	네트워크 스위치 (ToR/Spine)	- Spine-Leaf 구조 기반 대규모 동시성 트래픽 처리
가상화·오케스트레이션	가상화 플랫폼 (KVM/VMware)	- 컴퓨팅·스토리지·네트워크 자원을 가상화하여 유연한 운영 제공

보안·관리 체계	컨테이너 플랫폼 (Kubernetes)	- 대규모 컨테이너 자동 배포·확장·회복 기능 제공
	SDN/NFV 인프라	- 네트워크 기능을 소프트웨어로 제어/자동화
	접속제어·IAM	- 데이터센터 자원 접근 제어 및 계정·권한 통합 관리
	모니터링/관제 시스템	- 서버·네트워크·전력·냉각 상태를 실시간 수집·분석
	재해복구(DR) 체계	- 데이터센터 장애 시 서비스 지속을 위한 DR 및 백업 구성

- 대규모 컴퓨팅·스토리지·네트워크·전력·냉각 인프라가 모듈형으로 확장되는 데이터센터 구조

#### 나. 하이퍼스케일 데이터센터 기술요소

구분	기술	설명
컴퓨팅	GPU/TPU 기반 가속	- AI·HPC를 위한 대규모 연산 가속 기술 도입
	ARM·x86 혼합 아키텍처	- 저전력 ARM 서버와 고성능 x86 서버 혼용으로 비용·효율 최적화
	서버 디스어그리게이션	- 컴퓨팅·스토리지·메모리를 모듈화·분리해 필요한 만큼만 확장
스토리지	NVMe-oF	- 네트워크를 통해 NVMe 프로토콜로 원격 장치에 접근하는 방식
	객체 스토리지	- 대용량 데이터 저장에 최적화된 확장형 스토리지 기술
	Erasure Coding	- RAID 대비 효율적인 데이터 보호 및 내구성 확보
네트워크	Spine-Leaf 구조	- 수평적 확장이 가능한 데이터센터 네트워크 표준 구조
	100G/400G/800G 이더넷	- 대규모 트래픽 처리를 위한 고대역 네트워크
	SDN	- 중앙 제어 기반 자동화된 네트워크 구성·운영
운영 자동화	IaC	- 인프라를 코드로 정의해 자동 배포·변경 관리
	오토스케일링	- 부하에 따라 자동으로 서버·컨테이너 증감
	AI Ops	- 이상 탐지·장애 예측 등 AI 기반 운영 자동화
에너지 효율	액침냉각(Immersion Cooling)	- 서버를 냉각액에 담가 냉각 효율 대폭 향상
	Free Cooling	- 외기 활용으로 냉각 비용 절감
	PUE 최적화 기술	- 전력 효율(PUE)을 낮추기 위한 설비/운영 최적화 기술

- 고성능 네트워크, 분산 컴퓨팅·스토리지, 에너지 효율·냉각 기술을 통해 초대규모 서비스를 안정적으로 운영

## III. 기존 데이터센터와 하이퍼스케일 데이터센터의 비교

구분	기존 데이터센터	하이퍼스케일 데이터센터
확장 방식	- 서버 단위 점증적 확장 (Vertical/Small-Scale)	- 수천~수만 노드 단위 수평 확장(Horizontal Scaling)
인프라 구조	- 이기종 장비 기반, 커스텀 구성 많음	- 표준화·모듈화된 랙·서버·전력·냉각 구조
운영 방식	- 수동 관리 중심, 부분 자동화	- SDN/NFV 기반 완전 자동화, AIOps 운영
네트워크·전력 효율	- 전통 공냉 방식, PUE 상대적으로 높음	- 액침냉각·AI 기반 전력최적화로 초고효율(PUE 1.1 이하)
가용성·복구 전략	- 단일 센터 기반 DR 의존	- 리전·존 기반 지리적 분산 및 무중단 운영(핫스왑·라이브마이그레이션)
보안·정책	- 경계기반 보안·기본 접근통제	- 제로트러스트·암호화·AI 보안·정책 기반 접근 제어
운영 규모·대상	- 기업 내부용(온프레미스 중심)	- 글로벌 클라우드 사업자(AWS·Google·MS) 중심 초대형 워크로드 처리

- 하이퍼스케일 데이터센터는 대규모 확장성과 자동화 중심 운영으로 기존 데이터센터보다 월등한 효율성과 안정성을 제공

“끝”



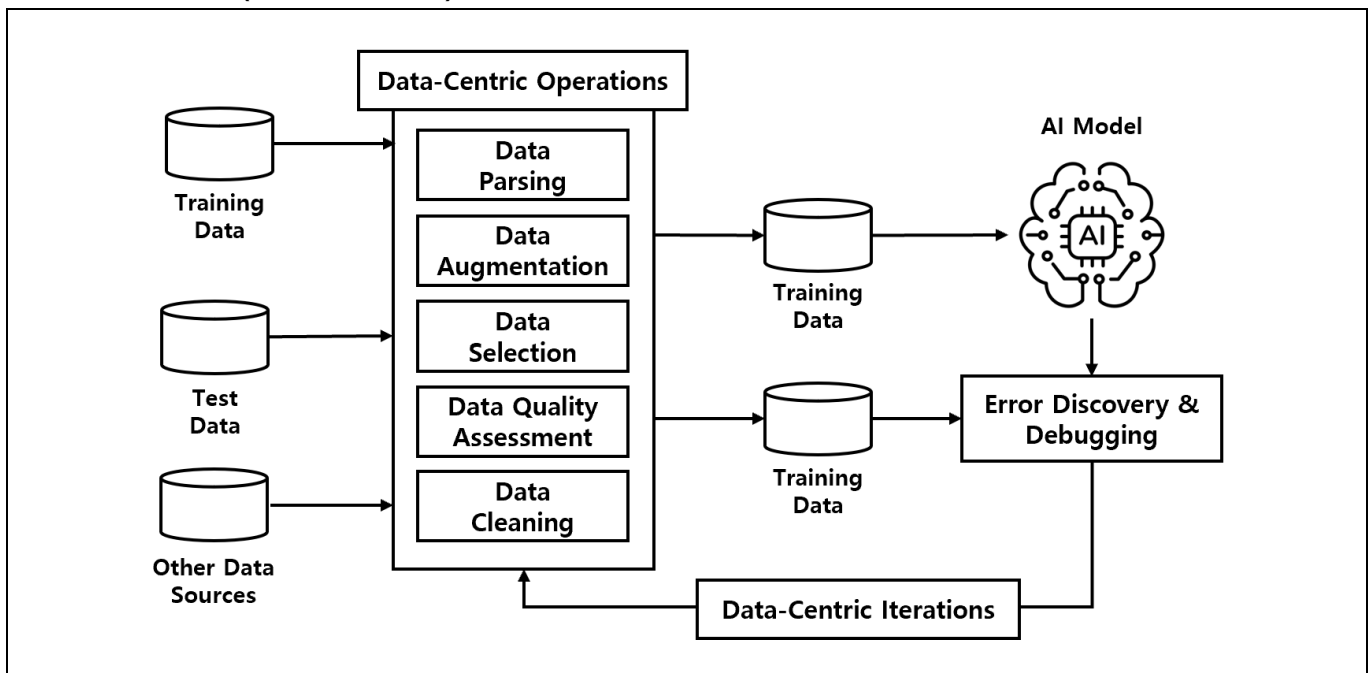
13	데이터 중심 AI(Data-Centric AI)		
문제	데이터 중심 AI(Data-Centric AI)		
도메인	인공지능	난이도	중 (상/중/하)
키워드	자동 어노테이션 및 라벨링 도구, 합성데이터(Synthetic Data), 센서 퓨전(Sensor Fusion), 데이터 품질 측정 지표, 학습용 데이터 품질 고도화, 데이터셋 정제 및 통합, 데이터 품질 진단, 커리큘럼 학습, 능동 학습, RAG, MLOps		
출제배경	데이터 중심의 인공지능 개발 방식 변화에 따른 출제 예상		
참고문헌	데이터 중심 AI, 글로벌 현황(한국데이터산업진흥원, 2025.07)		
출제자	소민호 기술사(제 119회 정보관리기술사 / mhsope@naver.com)		

### I. 모델 중심에서 데이터 중심으로, 데이터 중심 AI(Data-Centric AI)의 개념

구분	설명	
개념	- AI 성능 향상의 초점을 모델 개선이 아닌 데이터 품질에 두는 접근 방식으로, 학습 데이터를 체계적으로 관리·정제·최적화하여 인공지능의 성능과 신뢰성을 높이는 방법론	
특징	데이터 품질 우선 접근	- 모델 개선보다 데이터 정확성·일관성·라벨 품질 향상을 핵심으로 삼아 성능을 극대화하는 접근 방식
	지속적 데이터 개선	- 데이터 정제·라벨 검증·오류 수정·재수집을 반복해 학습 품질을 주기적으로 강화하는 구조
	효율적 학습 및 일반화 성능 향상	- 고품질 데이터 기반 학습으로 소량 데이터·적은 파라미터 모델도 높은 정확도·일반화 성능을 확보

### II. 데이터 중심 AI(Data-Centric AI)의 구성도 및 주요 기술

#### 가. 데이터 중심 AI(Data-Centric AI)의 구성도



- 데이터 중심 AI(Data-Centric AI)는 학습에 사용되었던 데이터를 수정하고 보완하여 AI 학습에 활용

#### 나. 데이터 중심 AI(Data-Centric AI)의 주요기술

구분	기술	설명
데이터 수집 및 가공 기술	실시간 데이터 수집 기술	- IoT, 센서, 웹, 로그 등 다양한 경로에서 실시간 데이터를 수집하며, 스트리밍 기술로 대규모 데이터를 신속 처리
	자동 어노테이션 및 라벨링 도구	- 데이터 라벨링을 자동화하거나 반자동화하여 효율성을 높이는 기술
	합성데이터	- 실제 데이터 부족 시 시뮬레이션이나 알고리즘을 통해 유사 데이터를 생성 (GAN, VAE, PBR 등 활용)
	센서 퓨전	- LiDAR·카메라·레이더 등 다양한 센서 데이터를 통합하여 인식 정확도와 신뢰도를 향상
데이터 품질 관리 기술	데이터 품질 측정 지표	- 정확성, 완전성, 일관성, 최신성 등 4대 품질 지표를 통해 데이터 품질을 정량화
	학습용 데이터 품질 고도화 기준	- 클래스 비율, 라벨 신뢰도, 이상치, 중복률 등 품질 기준 적용 및 오버샘플링·교차검증 등으로 품질 향상
	데이터셋 정제 및 통합	- 이상치 탐지, 중복 제거, 누락값 보완, 스키마 정렬 등을 통해 데이터 일관성과 정합성 확보
	데이터 품질 진단 및 자동화 도구	- 데이터 품질을 실시간 진단하고 리포트 제공
	산업별 품질 표준 준수 기술	- 금융·의료·제조 등 산업별 품질 기준을 자동 점검하여 실사용 요구조건 충족을 보장
데이터 중심 학습 기술	커리큘럼 학습 (Curriculum Learning)	- 난이도나 품질에 따라 데이터를 점진적으로 투입해 학습 효율 향상 (기초 → 복잡 순 학습)
	능동 학습 (Active Learning)	- 모델이 불확실한 데이터를 스스로 선택·요청하여 효율적으로 학습하는 방식
	RAG(Retrieval-Augmented Generation)	- 생성형 AI가 외부 DB나 문서를 검색해 관련 정보를 실시간 반영해 응답 생성 (정답형 AI 구현에 활용)
	데이터 중심 MLOps	- 모델 성능 개선을 데이터 품질·흐름 중심으로 자동화 관리하며, 데이터 버전·변경 이력·메타데이터를 통합 관리

- 데이터 중심 AI는 데이터의 품질·흐름·활용을 핵심 자산으로 삼아 모델 성능을 지속적으로 향상시키는 가장 실효적인 AI 개발 전략

## III. Data-Centric AI와 Model-Centric AI 비교

구분	Data-Centric AI	Model-Centric AI
개념	- 모델 구조는 비교적 고정하고 데이터 품질·다양성·정확성·레이블링 개선을 통해 성능을 향상하는 접근 방식	- 주어진 데이터셋을 고정하고 모델 구조, 하이퍼파라미터, 학습 기법 등을 개선하여 성능을 높이는 접근 방식
목표	- 데이터 품질·정확성 향상으로 성능 개선	- 모델 구조·알고리즘 개선으로 성능 향상
필요역량	- 데이터 분석·도메인 지식, 라벨링 품질 관리	- 모델링, 수학·통계, 프로그래밍 능력
접근방식	- 동일 모델 유지, 데이터셋 개선 반복	- 동일 데이터셋 유지, 모델 변경 반복
개선요소	- 데이터 정제, 라벨 품질, 샘플 균형, 데이터 증강	- 아키텍처 설계, 하이퍼파라미터 튜닝, 학습 기법
적용조건	- 데이터 품질·다양성이 부족 - 모델 변경 제약 존재 - 현장 데이터 위주	- 데이터가 이미 충분·표준화되어 있음 - 모델 성능 경쟁이 핵심 - 최신 알고리즘 적용 가능
주요활동	- 데이터 수집·가공, 라벨 오류 수정, 불균형 개선, 잡음 제거	- 새로운 모델 설계, 최적화 알고리즘 적용, GPU/TPU 활용
평가지표	- 데이터 품질 지표, 라벨 정확도, 커버리지, 클래스 균형도	- 정확도(Accuracy), F1-score, 추론 속도, 모델 크기
장점	- 실제 환경 적합성 ↑, 데이터 불량으로 인한 성능 저하 방지	- 최신 기술 적용 가능, 복잡 패턴 학습
단점	- 대규모 라벨링 비용, 품질 관리 자동화 한계	- 데이터 한계 시 성능 정체, 과적합 위험
적합 환경	- 데이터 부족, 현장 데이터 활용, 특화 도메인	- 풍부한 데이터, 알고리즘 성능 경쟁
사례	- 의료 영상 진단, 제조 결함 검사, OCR 데이터 정제	- 이미지넷 챌린지, SOTA 연구 개발
기대효과	- 현실 적합성 향상 - 적은 모델 변경으로 성능 개선	- 최신 SOTA 성능 확보 - 복잡 패턴 학습 가능

- 데이터 중심 AI는 모델보다 데이터를 우선 개선함으로써 정확성과 신뢰성을 지속적으로 향상시키는 차세대 AI 개발 패러다임

“끝”

13	인터리빙(Interleaving)		
문제	상위 인터리빙(High-Order Interleaving)과 하위 인터리빙(Low-Order Interleaving) 설명		
도메인	컴퓨터아키텍처	난이도	중 (상/중/하)
키워드	상위 인터리빙, 하위 인터리빙, 혼합 인터리빙, C-Access, S-Access, MSB, LSB, 디코더		
출제배경	CA 기본 개념으로 상위, 하위 인터리빙에 대한 이해 확인		
참고문헌	ITPE 기술사회 자료		
출제자	소민호 기술사(제 119회 정보관리기술사 / mhsope@naver.com)		

### I. 메모리 액세스의 효율 증대, 메모리 인터리빙 개념

- 버스의 경합이나 기억장치의 충돌 회피를 위해 기억장치를 여러 개의 독립적인 모듈들로 나누고 그 모듈들에서 동시에 액세스 동작이 일어날 수 있도록 하는 기법

### II. 상위 인터리빙(High-Order Interleaving)과 하위 인터리빙(Low-Order Interleaving) 설명

#### 가. 상위 인터리빙(High-Order Interleaving) 설명

구분	설명
개념	- 메모리 주소의 <b>상위 비트(MSB)</b> 를 이용하여 모듈을 선택하고, <b>하위 비트(LSB)</b> 로 큰 주소 블록 단위로 특정 뱅크에 배정하는 방식
개념도	
장점	<ul style="list-style-type: none"> <li>- 설계 단순성과 구현 용이성 제공</li> <li>- 큰 주소 영역 단위의 메모리 관리 효율성 확보</li> <li>- 한모듈 에러 시 해당 모듈만 영향</li> </ul>
단점	<ul style="list-style-type: none"> <li>- 연속된 주소가 동일 뱅크에 집중되어 병렬 접근성이 저하되는 문제 발생</li> <li>- 특정 뱅크로의 트래픽 쏠림 현상으로 인한 성능 저하 가능성 존재</li> </ul>

- 상위 인터리빙은 주소의 상위 비트로 메모리 뱅크를 선택해 병렬 접근 효율을 높이는 방식

나. 하위 인터리빙(Low-Order Interleaving) 설명

구분	설명
개념	- 메모리 주소의 <b>하위 비트(LSB)</b> 를 이용하여 모듈을 선택하고, <b>상위 비트(MSB)</b> 로 연속된 메모리 주소를 서로 다른 뱅크에 순차적으로 분산 배치하는 방식
개념도	
장점	<ul style="list-style-type: none"> <li>- 연속 주소 접근 시 여러 뱅크가 병렬로 동작하여 높은 메모리 대역폭 확보</li> <li>- 부하 분산 효과가 우수하여 대부분의 고성능 시스템에서 활용 가능</li> </ul>
단점	<ul style="list-style-type: none"> <li>- 메모리 주소 매핑 회로가 복잡해지고 설계 난이도 증가</li> <li>- 특정 워크로드에서는 기대만큼 균등 분산되지 않을 가능성 존재</li> <li>- 1개 새로운 모듈 추가 시 하드웨어 구조변경 수반, 한 모듈의 오류가 Memory 전체에 영향 전파</li> </ul>

- 하위 인터리빙은 주소의 하위 비트로 메모리 모듈을 선택해 연속 접근 성능을 높이는 방식

III. 메모리 인터리빙의 접근(access)방식

C-엑세스(Concurrent-access) 방식	S-엑세스(Simultaneous-access) 방식
- 주소들이 프로세스-기억장치간 버스를 통해 순차적으로 기억장치 모듈에 도착하는 방식	- 모든 기억장치 모듈들에서 읽기동작이 동시에 시작하고 읽혀진 데이터를 순차적으로 전송하는 방식

- S-Access 경우 연속적인 입력 데이터 흐름이 필요한 파이프라인 컴퓨터에서 많이 활용

“끝”

[참고] 혼합 인터리빙(High-Low-Order Interleaving) 설명

구분	설명
개념	- 상위 비트로 메모리 뱅크를 선택하고, 하위 비트로 각 뱅크 내 모듈을 순차적으로 배정하는 계층적 인터리빙 구조
개념도	<p>데이터 버스</p> <p>기억장치 M1, 기억장치 M2, ..., 기억장치 Mn-1, 기억장치 Mn</p> <p>뱅크 선택 신호 (MSB), 모듈 선택 신호 (LSB)</p> <p>뱅크선택, 모듈 내의 주소, 모듈선택</p>
장점	- 상·하위 인터리빙의 장점 결합 → 충돌 감소 + 병렬성 확대 - NUMA·멀티프로세서 환경 등에서 고성능 메모리 접근 패턴 지원
단점	- 주소 매핑 구조가 복잡하여 구현 비용 및 설계 난이도 증가 - 문제 발생 시 분석·디버깅 난이도가 높아지는 단점 존재

- 혼합 인터리빙은 상위 비트로 뱅크를, 하위 비트로 모듈을 선택해 병렬성과 연속 접근 효율을 동시에 높이는 방식

“끝”



## 제 39 회 ITPE 실전 명품 모의고사 해설집

대 상	정보관리기술사, 컴퓨터시스템응용기술사, 정보통신기술사, 정보시스템감리사 시험
발행일	2025년 12월 21일
집 필	강정배 PE, 전일 PE, 이상현 PE, 소민호 PE, 현수 PE, 박서현 PE, 배미경 PE
출 판	<b>ITPE(Information Technology Professional Engineer)</b>
주 소	ITPE 대치점 서울시 강남구 선릉로 86길 17 선릉애틀빌딩 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 ITPE 을지로점 서울시 중구 삼일대로 363, 2615호(장교동 장교빌딩) ITPE 강북점 서울 종로구 수표로 96, 7층 (관수동,국일관드림팰리스)
연락처	070-4077-1267 / <a href="mailto:itpe@itpe.co.kr">itpe@itpe.co.kr</a>

본 저작물은 [ITPE\(아이티피이\)](#)에 저작권이 있습니다.

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