

# 제127회 정보관리기술사 해설집

2022.04.16

## 국가기술자격 기술사 시험문제

기술사 제 127 회

제 1 교시(시험시간: 100 분)

|    |      |          |         |          |  |        |  |
|----|------|----------|---------|----------|--|--------|--|
| 분야 | 정보통신 | 자격<br>종목 | 정보관리기술사 | 수검<br>번호 |  | 성<br>명 |  |
|----|------|----------|---------|----------|--|--------|--|

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

1. 대체불가능 토큰(Non-Fungible Token)
2. 모델옵스(ModelOps)
3. 추정통계의 표본과 불편 추정량
4. 균형성과기록표(Balanced Score Card)의 구성요소
5. 메시지 인증 코드(Message Authentication Code)
6. 스프링 부트(Spring Boot)
7. 테스트 커버리지(Test Coverage)와 코드 커버리지(Code Coverage)
8. MongoDB의 개념 및 특징
9. 변조(Modification)와 위조(Fabrication)
10. SVM(Support Vector Machine)
11. 과적합(Overfitting)의 발생이유와 해결방안
12. 디지털 트윈(Digital Twin)과 메타버스(Metaverse)
13. DaaS(Desktop as a Service)

|      |  |     |           |
|------|--|-----|-----------|
| 01   | NFT(Nonfungible Token)   |     |           |
| 문제   | NFT(Nonfungible Token)   |     |           |
| 도메인  | 디지털서비스   | 난이도 | 중 (상/중/하) |
| 키워드  | 민팅(Minting), ERC-721, ERC-1155, 메타버스   |     |           |
| 출제배경 | 최근 디지털 자산에 대한 인증 및 소유권에 대한 관심 증가   |     |           |
| 참고문헌 | NFT(Non Fungible Token)을 둘러싼 최근 이슈와 저작권 쟁점(한국저작권위원회 / 전재림)<br>대체 불가능 토큰(wikipedia.org) |     |           |
| 해설자  | 안응원 기술사(제119회 정보관리기술사 / tino1999@naver.com)  |     |           |

## 1. 대체불가능한 토큰, NFT의 개념

### 가. NFT(Nonfungible Token)의 정의

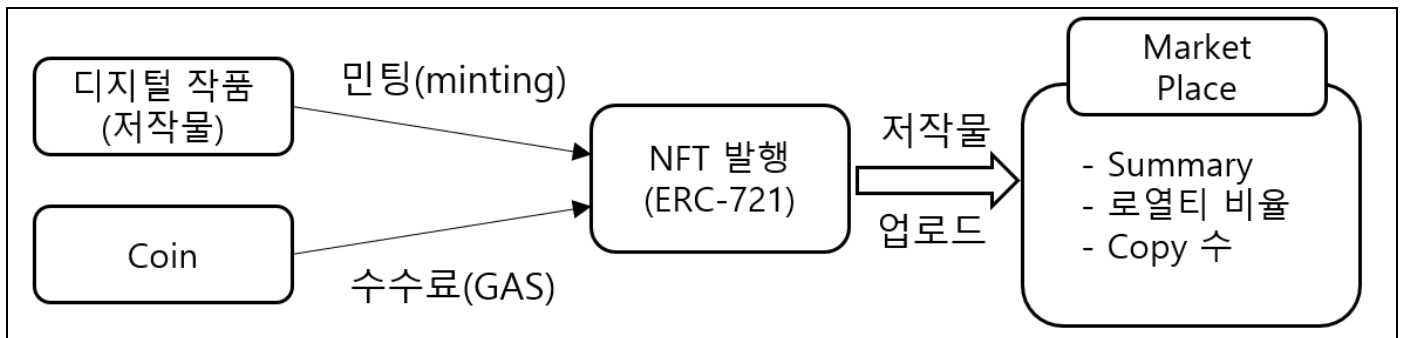
- 이미지나 동영상 등의 디지털 자산에 블록체인 기술을 이용하여, 원본인증 및 소유권 증명 개념을 부여하는 기술

### 나. NFT의 특징

- 위조의 어려움 : 복제가 어려워 희소성 보장 및 위조품으로 인한 가치 보장
- 소유권 인정 : 부분에 대한 소유권 인정해 토큰을 1/N 형태로 나누어 거래가능

## II. 이더리움 기반의 NFT 발행 프로세스 및 상세설명

### 가. 이더리움 기반의 NFT 발행 프로세스



- ERC(Ethereum Request for Comments)-721를 주로 활용하며 최근 ERC-1155표준도 활용

### 나. NFT의 표준동향 및 활용분야

| 구분 | 항목       | 설명   |
|----|----------|--|
| 표준 | ERC-721  | - 예술품, 골동품, 캐릭터에 대해 NFT를 발행하여 토큰을 생성하고, 해당 토큰의 고유한 가격이 결정되면 가상자산으로 거래하는 방식   |
|    | ERC-1155 | - ERC-20과 ERC-721의 장점을 혼합하여, 두 토큰이 연동하여 거래할 수 있도록 설계된 프로토콜<br>- 하나의 트랜잭션을 이용해 한 명 이상의 수신자에게 두 개 이상의 토큰을 보낼 수 있는 멀티 전송(Multi-transfers)이 가능한 프로토콜 |

|        |        |  |
|--------|--------|--|
| 수집     | 디지털 아트 | - 서명과 소유권을 보장하는 블록 체인 기술의 능력 때문에 NFT의 초기부터 많이 사용됨    |
|        | 수집품    | - 카드 컬렉션등의 수집품 형태로 활용                                |
| 엔터테인먼트 | 게임자산   | - 개발자 허용없이 자산을 타사 마켓플레이스에서 거래 가능                     |
|        | 음악     | - 음악가는 자신의 작품을 토큰화 하고 게시하여 수익 창출                     |
|        | 영화산업   | - 대용량의 영화는 작은 크기의 블록체인 기술에 적용이 어려워 NFT 사업에 참여가 낮은 특징 |
|        | 스포츠    | - 계약을 토큰화 하거나, 디지털 하이라이트 영상 등을 활용                    |

- ERC-1155 트랜잭션은 ERC-20의 약 60%의 GAS로 처리 가능하여 네트워크 병목현상 개선 및 효율성이 향상
- NFT는 다양한 플랫폼을 통해 서비스중

### III. NFT Platform의 분류

| 구분            | 설명                              | 사례                                  |
|---------------|---------------------------------|-------------------------------------|
| 오픈 플랫폼        | - 누구나 바로 민팅이 가능한 플랫폼            | OpenSea, Rarible                    |
| 승인형(큐레이션) 플랫폼 | - 플랫폼 승인 이후, 홍보, 관리, 지원 가능한 플랫폼 | Foundation, Niftygateway, SuperRare |

“끝”

|      |   |     |           |
|------|---|-----|-----------|
| 02   | 모델옵스(ModelOps)  |     |           |
| 문제   | 모델옵스(ModelOps)  |     |           |
| 도메인  | 인공지능  | 난이도 | 중 (상/중/하) |
| 키워드  | ML, Operation, DataOps, DevOps  |     |           |
| 출제배경 | 2020년에 가트너에서 정의한, 학습 모델에 대한 자동화되고 체계적인 운용 방법에 대한 이해   |     |           |
| 참고문헌 | MLOps and ModelOps (KDnuggets.com / 2022)<br>인공지능 모델 관리와 AI 개발·배포에 혁신적인 '모델옵스(ModelOps)'는 무엇인가?(인공지능신문)<br>ModelOps (wikipedia.org) |     |           |
| 해설자  | 안응원 기술사(제119회 정보관리기술사 / tino1999@naver.com)   |     |           |

## I. 기계학습과 생명주기의 효율적 관리, ModelOps의 개요

### 가. ModelOps의 정의

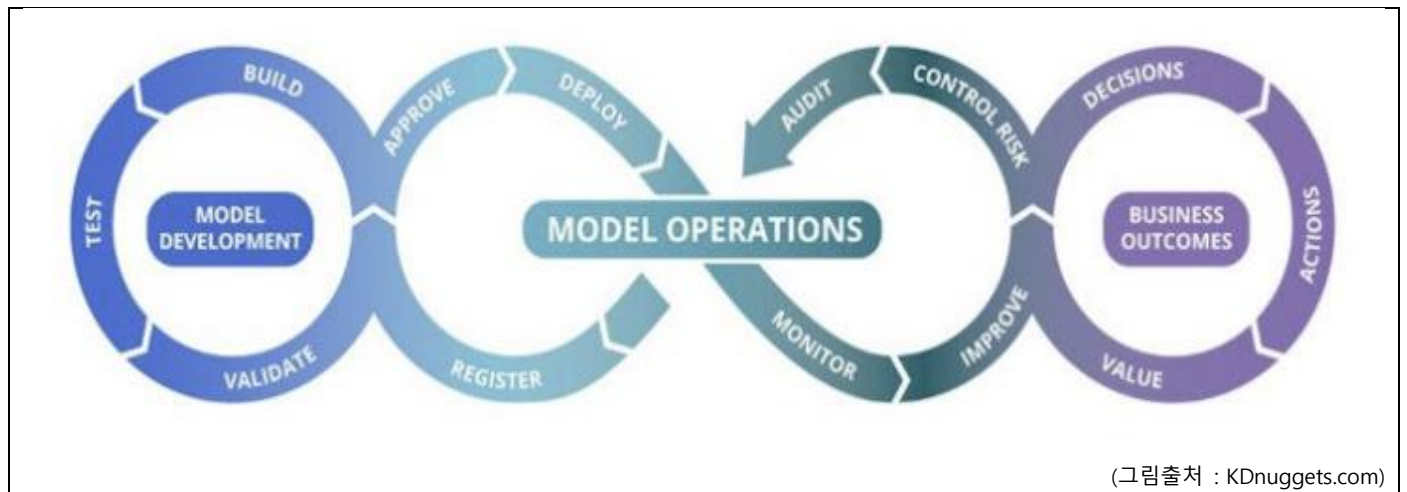
- 기계학습 모델과 운영(operation)을 결합한 용어로 기계학습(ML) 모델의 거버넌스와 생명 주기 관리를 효과적으로 관리하고 운용하는데 초점을 맞춘 방법론

### 나. ModelOps의 등장배경

|          |  |
|----------|--|
| 학습모델 자동화 | 모델을 학습하고 배포하는 과정은 대부분 전문가의 수작업에 의존하여 자동화 필요                      |
| Mash Up  | 데이터옵스(DataOps), 데브옵스(DevOps)와 결합하여 기존 서비스에 기계학습(ML) 기능을 쉽게 연동 가능 |

## II. ModelOps의 개념도와 기술요소

### 가. ModelOps의 개념도



- ModelOps는 AI모델이 생성되는 방식에 관계없이 비즈니스를 효과적으로 관리

#### 나. ModelOps의 프로세스 및 기술요소

| 구분   | 항목            | 설명  |
|------|---------------|---|
| 프로세스 | 1) Discovery  | 데이터 과학자, 분석가, 비즈니스 사용자가 선택한 소프트웨어를 사용하여 가설을 테스트하고 모델을 실험하고 구축   |
|      | 2) Governance | Discovery Environment에서의 모든 검증된 모델은 비즈니스 규칙 및 결정과 마찬가지로 모델 Repository에 저장(GitLab, GitHub 등)                     |
|      | 3) Deployment | - CI/CD를 통해 자동화 파이프라인을 생성.<br>- Repository에 저장된 모델, 비즈니스 규칙 및 결정은 코드 Repository에 저장된 정보와 결합되어 모델을 프로덕션에 원활하게 배포 |
| 기능   | 운영 자동화        | 모델 유형, 개발 방법, 모델 실행 위치에 관계없이 모델 운영의 모든 측면을 자동화  |
|      | 생명주기 관리       | 규제, 규정 준수, 위험관리 등 생산모델의 모든 측면 관리  |
|      | AI 가속화 및 확장   | 엔터프라이즈 기능을 MLOps 도구의 효율성과 결합  |

- 모델의 성능은 통계 및 비즈니스 KPI를 기반으로 추적되고 거버넌스 환경에 저장

#### III. ModelOps와 MLOps의 비교

| 구분  | ModelOps                     | MLOps              |
|-----|------------------------------|--------------------|
| 대상  | Enterprise                   | 데이터 사이언스 플랫폼       |
| 사용자 | Enterprise operation         | Data scientist     |
| 목적  | 생산모델에 대한<br>엔터프라이즈 거버넌스 및 운영 | ML 모델의 신속한 실험 및 배포 |

- ModelOps 를 효과적으로 적용하기 위해, 프로세스의 표준화 및 자동화 필요

“끝”

|      |   |     |           |
|------|---|-----|-----------|
| 03   | 추정통계의 표본과 불편 추정량                            |     |           |
| 문제   | 추정통계의 표본과 불편 추정량                            |     |           |
| 도메인  | 통계  | 난이도 | 중 (상/중/하) |
| 키워드  | 추정통계, 표본평균, 표본분산, 자유도, 추정량                  |     |           |
| 출제배경 | 추정통계에 대한 지속적인 출제로 인한 개념 확인                  |     |           |
| 참고문헌 | 지능적 분석 알고리즘 (연세대학교 강의자료)<br>자유도 (위키피디아)     |     |           |
| 해설자  | 안응원 기술사(제119회 정보관리기술사 / tino1999@naver.com) |     |           |

### I. 모집단의 통계량 도출, 추정통계의 개념

- 전수조사가 불가능한 모집단의 통계량 구하기 위해, 모집단에서 표본을 추출하여 모집단의 통계량을 추정하는 통계 기법

### II. 추정통계의 표본과 불편 추정량

#### 가. 추정통계의 표본의 설명

|  |  |
|--|--|
|  | <ul style="list-style-type: none"> <li>- 모집단에서 표본은 n개 추출할 수 있으며, 추출한 표본은 각각 평균과 분산을 통해 모집단을 추정 가능</li> <li>- 표본의 평균을 <math>\bar{X}_n</math>, 분산을 <math>S_n</math>이라고 할때, 이것들을 표본통계량(추정량)이라고 함</li> <li>- 모집단의 평균은 표본평균의 평균을 이용하여 추정하고, 모집단의 분산은 표본분산의 평균을 이용하여 추정</li> <li>- 모수를 추정하는 값들을 '추정량'이라 함</li> <li>- 표본의 평균은 모집단의 평균과 같은 특성이 있음</li> </ul> |
|--|--|

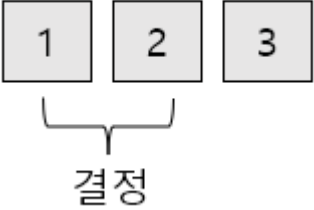
- 추정량의 기대값과 모수의 차이가 적을수록, 모집단의 통계량을 추정하기 용이함

#### 나. 불편추정량의 설명

| 구분        | 설명   |   |
|-----------|--|---|
| 개념        | - 추정량의 기대값과 실제 모수와의 차이인 편의(Bias)가 없는 추정량으로, 기대값과 모수가 동일한 추정량                   |   |
| 표본평균의 추정량 | $E(\bar{X}_1) = \frac{x_1 + x_1 + \dots + x_n}{n} = a$                         | - 표본평균은 모수와 동일한 특성으로 인해 불편추정량임.                                     |
| 표준분산의 추정량 | $V(\bar{X}_1) = \frac{(x_1 - a)^2 + (x_2 - a)^2 + \dots + (x_n - a)^2}{n - 1}$ | - 표준분산은 분모를 n으로 나누게 되면, 모수의 표준분산과 달라지게되어, 불편추정량 생성을 위해 n-1로 나누어야 함. |

- 표본분산의 추정량을 위해 분모를  $n-1$ 로 나누는데, 이것을 자유도라고 함

### III. 불편추정량의 자유도

|   |   |
|---|---|
|  | <ul style="list-style-type: none"> <li>- 3개의 변수에서 평균이 6일 때, 2개의 변수가 결정되면 나머지는 한 개의 변수는 자동으로 결정이 되는 개념</li> <li>- 넓은 의미로, 자유롭게 선택할 수 있는 변수의 수를 의미</li> </ul> |
|---|---|

“끝”



|             |   |            |                  |
|-------------|---|------------|------------------|
| <b>04</b>   | <b>균형성과기록표(Balanced Score Card)</b>         |            |                  |
| <b>문제</b>   | 균형성과기록표(Balanced Score Card)의 구성요소          |            |                  |
| <b>도메인</b>  | 경영전략  | <b>난이도</b> | <b>하</b> (상/중/하) |
| <b>키워드</b>  | 재무, 고객, 내부프로세스, 학습과 성장,                     |            |                  |
| <b>출제배경</b> | 125회 BSC 전략맵 출제에 대한 기본 개념 확인                |            |                  |
| <b>참고문헌</b> | ITPE 서브노트                                   |            |                  |
| <b>해설자</b>  | 안응원 기술사(제119회 정보관리기술사 / tino1999@naver.com) |            |                  |

## I. 균형성과기록표(BSC)의 개요

### 가. 균형성과기록표(BSC)의 정의

- 재무적 관점과 고객, 내부프로세스, 학습과 성장측면의 3가지 비재무적 관점을 측정 가능한 핵심성과지표(KPI)로 전환하여 관리함으로써, 균형적인 경영성과관리를 실행하도록 도와주는 성과관리기법

### 나. 균형성과기록표(BSC)의 특징

|             |  |
|-------------|--|
| 지표간의 균형     | 재무-비재무, 결과-과정, 내부-외부, 과거-미래 지표간의 균형              |
| 전략과 연계      | 조직의 전략으로부터 도출되어, 조직의 비전 및 전략을 이행하기 위한 목표를 기반으로 함 |
| 전략에 대한 의사소통 | 전략에 대한 조직 구성원간의 의사소통이 원활해짐으로써 공통의 목표를 지향하게 함     |

## II. 균형성과기록표(BSC)의 구성도 및 구성요소

### 가. 균형성과기록표(BSC)의 구성도



나. 균형성과기록표(BSC)의 구성요소

| 관점         | 설명  | KPI                                     |
|------------|---|---|
| 재무적관점      | - 전통적인 성과측정 관점으로 매출액, 순이익, 원가절감 등 재무적인 평가               | - 매출액<br>- 현금흐름<br>- 영업이익률              |
| 고객관점       | - 고객의 충성도 제고, 고객의 만족도향상 등 고객관계 관점 평가                    | - 고객 만족도<br>- 적시 공급률                    |
| 내부 프로세스 관점 | - 조직 내부프로세스의 최적화 관점에서의 성과를 평가하고, 프로세스 향상에 대한 노력을 성과로 인정 | - 생산 Cycle Time<br>- 단위원가<br>- 신제품 개발건수 |
| 학습과 성장 관점  | - 조직에서 구성원의 중요성을 강조<br>- 구성원의 학습노력과 역량증진에 대한 성과를 측정     | - 종업원 만족도<br>- 자격증 취득건수                 |

- BSC는 무형자산들을 지표화함으로써, 경영의 성과를 관리하며, IT 분야에서는 IT-BSC도 활용

III. BSC와 IT-BSC 개념 비교

| 구분   | BSC  | IT-BSC   |
|------|--|--|
| 개념   | - 재무/고객/내부프로세스/학습과 성장 관점의 성과지표를 도출하여 균형적인 성과관리방법 | - BSC를 개선, IT 서비스 품질향상 및 개발생산성 향상을 위해 기업공헌도/사용자/운영 프로세스/미래지향적 관점의 성과관리방법 |
| 목적   | - 내부 비즈니스 프로세스와 고객 서비스 및 제품의 개선과 재무적 성장 관계 파악    | - 내부 기술에 대한 고객 서비스 측면의 성과 측정   |
| 목표   | - 기업의 행동과 프로세스 변화                                | - IT Value 전달 및 성과 달성 목적을 제시   |
| 활동   | - 비즈니스 거버넌스                                      | - IT 거버넌스  |
| 중점분야 | - 비즈니스 분야  | - 기술적 분야   |
| 활용사례 | - 경영전략 수립, 주주대상 보고                               | - IT부서평가, IT투자 전개 참고   |
| 고려사항 | - 기업내 전 영역 종합적 평가 필요                             | - IT담당자, 주요관점별 실무지표  |

“끝”

|      |  |     |           |
|------|--|-----|-----------|
| 05   | 메시지 인증 코드(Message Authentication Code)   |     |           |
| 문제   | 메시지 인증 코드(Message Authentication Code)   |     |           |
| 도메인  | 보안   | 난이도 | 중 (상/중/하) |
| 키워드  | 무결성, 인증, Key, Hashing, Nested MAC, HMAC, CMAC  |     |           |
| 출제배경 | 117회 메시지 인증, HMAC 출제 후 메시지 인증에 대한 토픽 이해 확인   |     |           |
| 참고문헌 | ITPE 서브노트<br><a href="https://ko.wikipedia.org/wiki/메시지_인증_코드">https://ko.wikipedia.org/wiki/메시지_인증_코드</a> |     |           |
| 해설자  | 정상반 이상헌 기술사(제 118회 정보관리기술사 / bluesanta97@naver.com)  |     |           |

## I. 데이터 무결성 보장, 메시지 인증 코드(Message Authentication Code)의 개요

### 가. 메시지 인증 코드(Message Authentication Code)의 정의

- 송·수신자 간 비밀키를 공유하는 것으로 안전한 채널이 확보되지 않아도 무결성과 전송자의 신원 인증을 가능하게 하는 기술

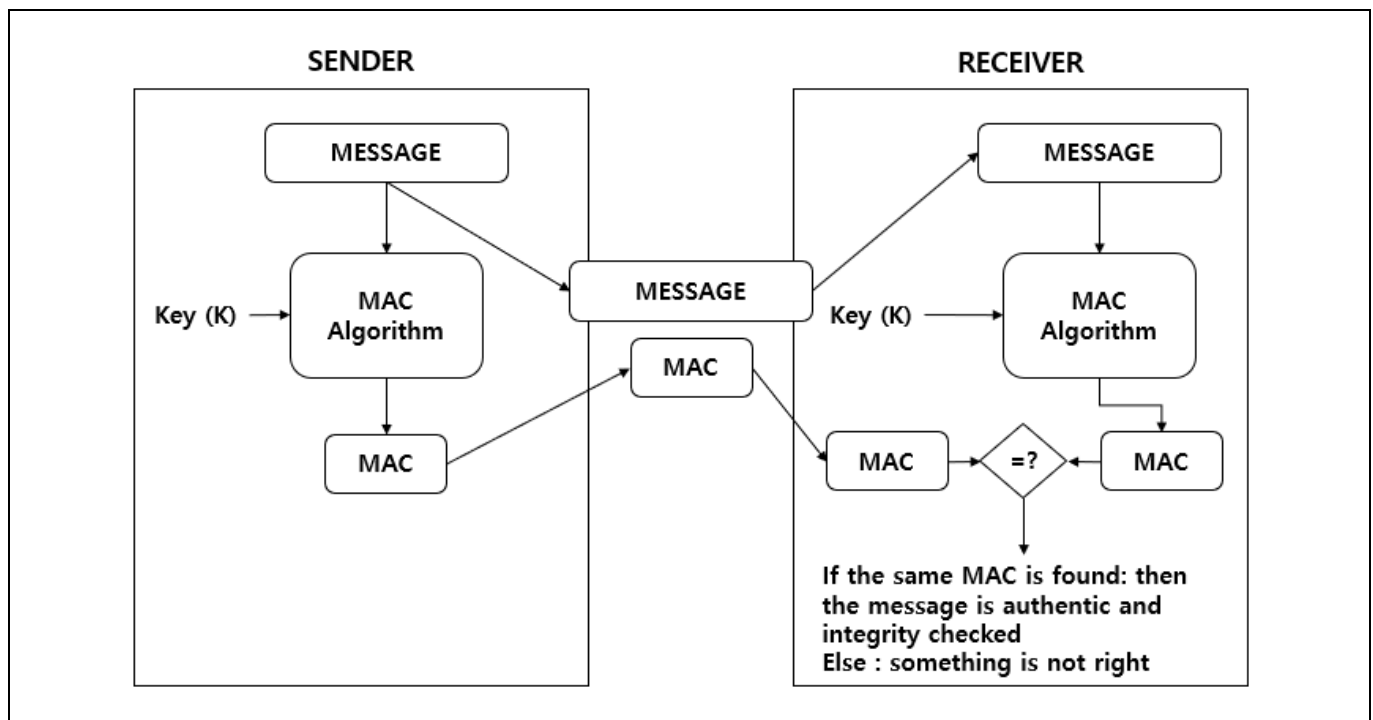
### 나. 메시지 인증 코드의 특징

|         |                                      |
|---------|--------------------------------------|
| 메시지 무결성 | - 인가된 당사자에 의해 인가된 방법으로 문세를 변경 가능한 성질 |
| 메시지 인증  | - 송신자로부터 보낸 메시지 여부를 확인 할 수 있는 성질     |

- 메시지 인증 코드는 전송 도중 위변조 여부확인에 사용되며, 전송 상호간 비밀키를 공유하여야 함

## II. 메시지 인증 코드의 동작 절차 및 종류

### 가. 메시지 인증 코드의 동작 절차



| 대상       | 동작절차      | 설명  |
|----------|-----------|---|
| 공통       | 1. 키교환    | - 송신자와 수신자는 사전에 키를 공유함. (공유키)   |
| SENDER   | 2. MAC 계산 | - 송신자는 사전에 공유한 공유키를 기초로 MAC 값 계산  |
|          | 3. 메시지 전송 | - 송신자는 송신할 메시지와 계산된 MAC 값을 수신자에게 전송                                     |
| RECEIVER | 4. MAC 계산 | - 송신자의 메시지 기반으로 공유키를 이용하여 MAC를 계산                                       |
|          | 5. MAC 비교 | - 송신자의 MAC와 수신자가 계산한 MAC값 비교  |
|          | 6. 결과 검토  | - 두 MAC 값이 동일하면 메시지는 인증되고 무결한 것임<br>- 값이 같지 않다면 송신자에게서 온 메시지가 아닌 것으로 판단 |

나. 메시지 인증 코드의 종류

| 구분            | 개념도   | 설명  |
|---------------|---|---|
| Nested MAC    | <pre> graph TD     subgraph InputBox [ ]         Key1[Key]         MESSAGE[MESSAGE]     end     InputBox --&gt; HASH1[HASH]     HASH1 --&gt; subgraph InputBox2 [ ]         Key2[Key]         MAC1[MAC]     end     InputBox2 --&gt; HASH2[HASH]     HASH2 --&gt; MAC[MAC]         </pre>   | <ul style="list-style-type: none"> <li>- 키와 메시지를 Hashing하고 나온 MAC를 다시 MAC와 키로 Hasing 하여 최종 MAC를 구하는 기법</li> </ul>                             |
| HMAC (Hash)   | <pre> graph TD     MESSAGE[MESSAGE] --&gt; HASH1[HASH Function]     Key[Key] --&gt; IPAD[IPAD]     Key --&gt; OPAD[OPAD]     IPAD --&gt; K1[K1]     OPAD --&gt; K2[K2]     K1 --&gt; HASH1     HASH1 --&gt; H_prime[H' ← HASH(K1  M)]     H_prime --&gt; HASH2[HASH Function]     K2 --&gt; HASH2     HASH2 --&gt; MAC[MAC]         </pre>      | <ul style="list-style-type: none"> <li>- 일방향 Hash 함수를 이용하여 MAC를 구하는 기법</li> <li>- 두단계 Hash 계산</li> <li>- 계산 중 Padding값 포함되어 훨씬 복잡함</li> </ul> |
| CMAC (Cipher) | <pre> graph TD     M1[M1] --&gt; E1[Encryption algorithm]     M2[M2] --&gt; E2[Encryption algorithm]     Mn[Mn] --&gt; En[Encryption algorithm]     Key[k] --&gt; E1     Key[k] --&gt; E2     Key[k] --&gt; En     E1 --&gt; E2     E2 --&gt; En     En --&gt; Select[Select n leftmost bits]     Select --&gt; CMAC[n-bit CMAC]         </pre> | <ul style="list-style-type: none"> <li>- 블록암호 기반 메시지 인증 코드</li> <li>- CBC-MAC의 변형</li> </ul>  |

- 사전 공유키를 송신자와 수신자가 공유해야 하기 때문에 키 공유를 위한 키 배송문제를 해결할 필요가 있음

## III. 메시지 인증 코드의 주요 활용 사례

| 활용 사례   | 설명   |
|---------|--|
| SWIFT   | - 송금결제 메시지를 인증하고 무결성을 보장하기 위해 메시지 인증 코드 사용               |
| IPSec   | - 통신 내용의 인증과 무결성을 확인하기 위해 메시지 인증 코드를 이용                  |
| SSL/TLS | - 통신 내용의 인증과 무결성을 확인하기 위해 메시지 인증 코드를 이용                  |
| VPN     | - 공용 네트워크로 전달되는 트래픽은 암호화 및 메시지인증 코드 등을 사용하여 기밀성과 무결성을 제공 |

“끝”

|      |   |     |           |
|------|---|-----|-----------|
| 06   | 스프링 부트(Spring Boot)   |     |           |
| 문제   | 스프링 부트(Spring Boot)   |     |           |
| 도메인  | 소프트웨어 공학  | 난이도 | 중 (상/중/하) |
| 키워드  | Controller Layer/Service Layer/Repository Layer/Domain Layer, Annotation                                  |     |           |
| 출제배경 | JAVA 개발 시 많이 사용되는 스프링 프레임워크를 좀더 쉽게 사용할 수 있게 하는 스프링 부트에 대한 기본 이해 확인  |     |           |
| 참고문헌 | <a href="https://www.bottlehs.com/springboot/스프링-부트-구조">https://www.bottlehs.com/springboot/스프링-부트-구조</a> |     |           |
| 해설자  | 정상반 이상헌 기술사(제 118회 정보관리기술사 / bluesanta97@naver.com)   |     |           |

## I. 스프링을 더 쉽게 사용할 수 있게 하는 스프링 부트(Spring Boot)의 개요

### 가. 스프링 부트(Spring Boot)의 정의

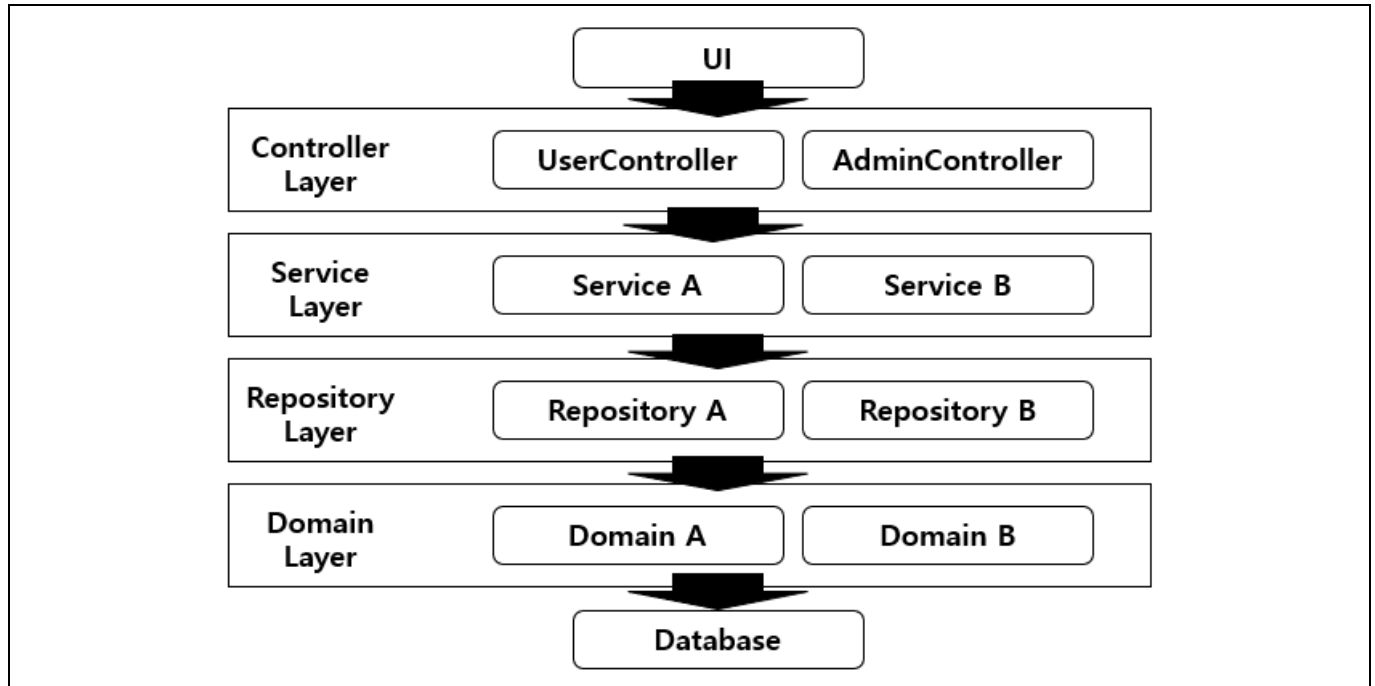
- 스프링 프레임워크를 사용하는 프로젝트를 복잡한 설정 없이 쉽고 빠르게 만들어주는 라이브러리/프레임워크

### 나. 스프링 부트의 장점

|                  |   |
|------------------|---|
| 라이브러리관리 자동화      | - 스프링 부트의 Starter 라이브러리를 등록하여 라이브러리 의존성 관리  |
| 라이브러리 버전 자동관리    | - pom.xml에 스프링 부트 버전을 입력하면 스프링 라이브러리뿐 아니라 서드 파티 라이브러리 버전도 관리  |
| 설정의 자동화          | - @EnableAutoConfiguration 어노테이션을 선언해서 스프링에서 자주 사용했던 설정들을 알아서 등록  |
| 내장 Tomcat        | - 스프링 부트는 Tomcat을 내장하고 있기 때문에 @SpringBootApplication 어노테이션이 선언되어 있는 클래스의 main() 메소드를 실행하는 것만으로 서버를 구동 시킬 수 있음 |
| 독립적으로 실행 가능한 JAR | - 스프링 부트는 내장 톰캣을 지원하기 때문에 JAR 파일로 패키징해서 웹 애플리케이션을 실행시킬 수 있음   |

## II. 스프링 부트의 Architecture 및 구성요소

### 가. 스프링 부트의 Architecture



### 나. 스프링 부트의 구성요소

| 구분         | 항목                       | 설명   |
|------------|--------------------------|--|
| Layer      | Controller Layer         | - UI에서 요청을 받고 응답을 전달   |
|            | Service Layer            | - 주요 비즈니스 로직을 구현   |
|            | Repository Layer         | - 데이터베이스에서 가져올 쿼리를 구현<br>- JPA를 이용하는 경우 정해진 규칙에 따라서 메소드를 사용하거나, 만들어 놓으면 적절한 쿼리를 수행할 수 있음 |
|            | Domain Layer             | - 실제로 데이터베이스 물리 테이블과 1:1 매핑이 되어 바인딩 됨.   |
| Annotation | @RestController          | - @Controller 와 @ResponseBody를 합친 어노테이션으로 메소드의 반환 결과를 JSON 형태로 반환                        |
|            | @RequestMapping          | - 요청 URL을 어떤 메소드가 처리할 것인지 매핑   |
|            | @Bean                    | - 스프링 컨테이너에 Bean을 등록   |
|            | @Autowired               | - 생성자나 setter 메소드 없이 의존성을 주입해서 자동으로 객체를 생성   |
|            | @Configuration           | - 해당 클래스가 환경 설정과 관련된 파일이라는 것을 알려줌  |
|            | @ConfigurationProperties | - 프로퍼티 파일을 읽어와서 해당 값들을 사용할 수 있게 만들어 줌  |
|            | @Transactional           | - 해당 클래스 또는 메소드에 트랜잭션을 적용  |
|            | @Controller              | - 요청과 응답을 처리해주는 클래스에 사용  |
|            | @Service                 | - 비즈니스 로직을 담당하는 클래스에 사용  |

|  |                   |                                       |
|--|-------------------|---------------------------------------|
|  | @Repository       | - DB에 액세스하는 클래스에 사용                   |
|  | @ExceptionHandler | - 컨트롤러 단에서 발생하는 예외를 잡아서 처리해주는 메소드에 사용 |
|  | @ControllerAdvice | - 전역에서 발생하는 예외들을 잡아서 처리해주는 클래스에 사용    |

- 스프링 부트 사용시 스프링 프레임워크에서 관리하기 어려웠던 사항에 대한 개발자 편의성을 높임.

### III. 스프링 부트의 일반적인 소스 코드 구조

```

com
+- example
  +- myapplication
    +- Application.java
    |
    +- customer
      +- Customer.java
      +- CustomerController.java
      +- CustomerService.java
      +- CustomerRepository.java
  
```

“끝”



|      |  |     |           |
|------|--|-----|-----------|
| 07   | 테스트 커버리지(Test Coverage), 코드 커버리지(Code Coverage)  |     |           |
| 문제   | 테스트 커버리지(Test Coverage)와 코드 커버리지(Code Coverage)  |     |           |
| 도메인  | 소프트웨어 공학   | 난이도 | 중 (상/중/하) |
| 키워드  | 테스트 커버리지: Statement Coverage, Branch Coverage, Path Coverage<br>코드 커버리지: 구문 커버리지, 결정 커버리지, 조건 커버리지, 조건/결정 커버리지, 변경조건/결정 커버리지, 다중조건 커버리지, 경로 커버리지 |     |           |
| 출제배경 | 소프트웨어 테스트의 기본인 커버리지에 대한 명확한 이해 확인  |     |           |
| 참고문헌 | ITPE 서브노트<br><a href="https://www.simform.com/blog/test-coverage/">https://www.simform.com/blog/test-coverage/</a>                               |     |           |
| 해설자  | 정상반 이상헌 기술사(제 118회 정보관리기술사 / bluesanta97@naver.com)  |     |           |

## I. 테스트 케이스 실제 수행 여부 확인, 테스트 커버리지(Test Coverage) 설명

### 가. 테스트 커버리지(Test Coverage)의 정의 및 특징

| 구분 | 설명   |   |
|----|--|---|
| 정의 | - 테스트 케이스가 실제로 애플리케이션 코드를 커버하는지 여부와 테스트 케이스를 실행할 때 얼마나 많은 코드가 실행되는지를 결정하는 기술 |   |
| 특징 | 높은 커버리지  | - 우수한 커버리지 보장을 위해 더 많은 테스트케이스 생성              |
|    | 중복 제거  | - 프로젝트에서 별로 의미가 없는 테스트 사례를 식별하고 제거하는 데 특히 유용함 |
|    | 높은 ROI   | - 테스트 적용 범위는 ROI에 상당한 영향을 미칠 수 있음             |
|    | 테스트 미수행 지역 발견  | - 테스트 케이스는 테스트 케이스에서 다루지 않는 영역 찾음             |

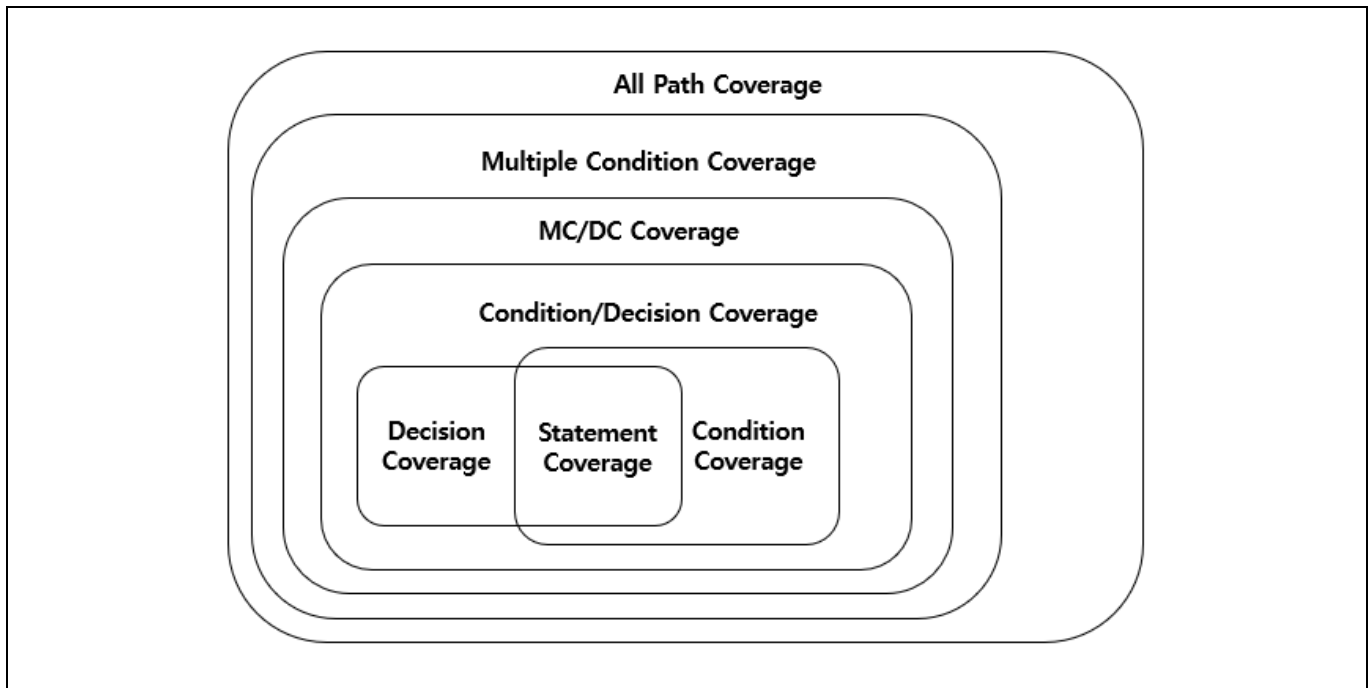
### 나. 테스트 커버리지(Test Coverage)의 유형

| 구분 | 설명  |
|----|---|
| 예시 | <pre> graph TD     1[Read X, Read Y] --&gt; 2{X+Y &gt; 20}     2 -- TRUE --&gt; 3[Print 'A']     2 -- False --&gt; D(( ))     D --&gt; 4{X &gt; 50}     4 -- TRUE --&gt; 5[Print 'A']     4 -- False --&gt; H(( ))     style D fill:none,stroke:none     style H fill:none,stroke:none         </pre> |

|                    |   |
|--------------------|---|
| Statement Coverage | <ul style="list-style-type: none"> <li>- 소스 코드의 모든 명령문이 최소 한 번 테스트되었음을 보장</li> <li>- 커버리지: 1A-2C-3D-E-4G-5H</li> </ul>  |
| Branch Coverage    | <ul style="list-style-type: none"> <li>- 가능한 모든 경로 또는 분기가 적용되는지 확인</li> <li>- 커버리지#1(True 조건 점검): 1A-2C-3D-E-4G-5H</li> <li>- 커버리지#2(False 조건 점검): 1A-2B-E-4F</li> </ul>  |
| Path Coverage      | <ul style="list-style-type: none"> <li>- 가능한 모든 실행 경로를 찾기 위해 프로그램의 소스 코드를 사용하는 것</li> <li>- 처음부터 끝까지 모든 경로의 적용을 보장함</li> <li>- 커버리지#1: 1A-2B-E-4F</li> <li>- 커버리지#2: 1A-2B-E-4G-5H</li> <li>- 커버리지#3: 1A-2C-3D-E-4G-5H</li> <li>- 커버리지#4: 1A-2C-3D- E-4F</li> </ul> |

- 이외 각 테스트 케이스의 경계 값을 검증하는 Boundary Value Coverage도 있음

## II. 테스트 수행범위 검증, 코드 커버리지(Code Coverage) 설명



| 구분 | 설명  |  |
|----|---|--|
| 정의 | - 테스트 대상의 전체범위에서 테스트를 수행한 범위 정도로 테스트 대상을 얼마나 테스트했는지 테스트의 정확성을 판단하는 척도 |  |
| 유형 | 구문 커버리지(SC)   | - 모든 명령문을 적어도 한번 수행하는 테스트 케이스                            |
|    | 결정 커버리지(DC)   | - 전체 결정문이 적어도 한번은 참과 거짓의 결과를 수행하는 테스트 케이스                |
|    | 조건 커버리지(CC)   | - 명령문 내의 각 개별 조건식이 적어도 한번은 참과 거짓의 결과가 되도록 수행하는 테스트 케이스   |
|    | 조건/결정 커버리지(C/DC)  | - 전체 결정문뿐만 아니라 개별 조건식도 참 한번, 거짓 한번의 결과가 되도록 수행하는 테스트 케이스 |

|  |                      |   |
|--|----------------------|---|
|  | 변경조건/결정 커버리지 (MC/DC) | - 각 개별 조건식이 다른 개별 조건식에 영향을 받지 않고 전체 결정문에 독립적으로 영향을 주도록 하는 테스트 케이스 |
|  | 다중조건 커버리지(MCC)       | - 모든 개별식 조건이 모든 조합을 고려한 커버리지                                      |
|  | 경로 커버리지(PC)          | - 모든 경로를 적어도 한번 수행하는 테스트 케이스                                      |

- 다중조건 커버리지가 가장 넓은 커버리지 이나, 상황에 적합한 테스트 케이스 선택이 바람직함

### III. 테스트 커버리지와 코드 커버리지 비교

| 구분   | 테스트 커버리지                          | 코드 커버리지          |
|------|-----------------------------------|------------------|
| 유형   | 블랙박스 테스트                          | 화이트박스 테스트        |
| 사용이유 | 소프트웨어 개선에 필요한 새로운 테스트 케이스 발굴 및 측정 | 효율적 테스트 수행 능력 측정 |
| 효과   | 기능 확인                             | 프로그램 흐름에서 버그 발견  |

“끝”

|      |   |     |           |
|------|---|-----|-----------|
| 08   | MongoDB   |     |           |
| 문제   | MongoDB의 개념 및 특징  |     |           |
| 도메인  | 데이터베이스  | 난이도 | 중 (상/중/하) |
| 키워드  | Document DB, 비관계형, 비 트랜잭션, 확장성, HA, Collection, Document, Field                     |     |           |
| 출제배경 | 다양한 환경에서 사용되는 NoSQL의 대표적 데이터베이스인 MongoDB에 대한 이해 확인                                  |     |           |
| 참고문헌 | <a href="https://meetup.toast.com/posts/274">https://meetup.toast.com/posts/274</a> |     |           |
| 해설자  | 정상반 이상헌 기술사(제 118회 정보관리기술사 / bluesanta97@naver.com)                                 |     |           |

## I. Document 기반 NoSQL, MongoDB의 개념

|   |  | 정의   |
|---|--|--|
| <div style="display: flex; justify-content: space-around; align-items: center;"> <div style="text-align: center;"> <b>RDBMS</b><br/> <div style="border: 1px solid black; padding: 5px; margin: 5px;">Database</div> <div style="border: 1px solid black; padding: 5px; margin: 5px;"> Tables <div style="border: 1px solid black; padding: 5px; margin: 5px;"> Rows <div style="border: 1px solid black; padding: 5px; margin: 5px;">Columns</div> </div> </div> </div> <div style="text-align: center;"> <b>MongoDB</b><br/> <div style="border: 1px solid black; padding: 5px; margin: 5px;">Database</div> <div style="border: 1px solid black; padding: 5px; margin: 5px;"> Collection <div style="border: 1px solid black; padding: 5px; margin: 5px;"> Documents <div style="border: 1px solid black; padding: 5px; margin: 5px;">Fields</div> </div> </div> </div> </div> |  | <ul style="list-style-type: none"> <li>- 프로덕션 환경에서 데이터베이스를 확장할 수 있는 기능을 제공하는, C++로 작성된 Document 지향의 NoSQL 데이터베이스 관리 시스템</li> </ul> |

## II. MongoDB의 특징 및 구성 방식

### 가. MongoDB의 특징

| 특징                | 설명   |
|-------------------|--|
| Document-oriented | <ul style="list-style-type: none"> <li>- MongoDB는 JSON형태로 데이터를 관리</li> <li>- 도큐먼트는 MongoDB가 데이터를 저장하는 최소 단위</li> </ul>   |
| 유연한 스키마           | <ul style="list-style-type: none"> <li>- 스키마의 선언 없이 필드의 추가와 삭제가 자유로운 Schema-less 구조</li> </ul>   |
| 비 관계형 데이터베이스      | <ul style="list-style-type: none"> <li>- 관계(Relationship) 개념이 없는 비 관계형 데이터베이스</li> <li>- 조인(Join)을 지원하지 않으며, 대신 임베디드 방식의 도큐먼트 구조를 사용하거나 레퍼런스 방식의 도큐먼트 구조를 사용한 후 애플리케이션에서 조인</li> </ul> |
| 비 트랜잭션            | <ul style="list-style-type: none"> <li>- 트랜잭션을 지원하지 않고 각각의 도큐먼트 단위로 처리됩니다.</li> <li>- 트랜잭션을 지원하지 않으므로 Commit 또는 Rollback 개념이 없으며 모두 Auto Commit으로 처리</li> </ul>                        |
| 높은 확장성            | <ul style="list-style-type: none"> <li>- Auto-Sharding방식 제공</li> </ul>   |
| 높은 안전성(HA)        | <ul style="list-style-type: none"> <li>- Master/slave방식 제공, replica-set 방식 제공, auto-sharding 방식과 혼합하여 구성할 경우 band-width, 트래픽 분산의 이점이 있음</li> </ul>                                     |

|            |                               |
|------------|-------------------------------|
| Map-Reduce | - 맵리듀스 기능 제공                  |
| GridFS     | - 분산 파일 저장을 MongoDB가 자동으로 수행함 |

#### 나. MongoDB의 구성방식

| 구성방식        | 개념도 | 설명   |
|-------------|-----|--|
| Replica Set |     | <ul style="list-style-type: none"> <li>- Master/Slave 구조보다 안정적</li> <li>- Primary 서버에 장애 감지시 새로운 Primary 서버 선정 (Automated Failover)</li> <li>- Data Redundancy</li> <li>- 일관성/가용성보장</li> </ul> |
| Sharding    |     | <ul style="list-style-type: none"> <li>- Range 파티션 기법으로 Shard Key 중심으로, Shard 서버에 분산 저장하는 방식. 데이터 자동 Balancing</li> </ul>  |

- MongoDB는 다양한 환경에서 사용가능 하나 여러 단점들을 확인하고 개선 및 용도 확인 후 사용해야 함.

#### III. MongoDB의 주요 단점

| 단점       | 설명   |
|----------|--|
| Join 미지원 | - Join 기능을 제공하지 않아 Join이 필요 없도록 데이터 구조화가 선행되어야 함                   |
| 메모리 의존적  | - Memory Mapped File으로 파일 엔진 DB이며 메모리 관리를 OS에 위임하고 메모리 크기가 성능을 좌우함 |
| 성능 저하    | - 인덱스는 B-Tree 인덱스 사용하여 크기가 커질수록 새로운 데이터 입력하거나 삭제시 성능 저하됨           |

“끝”

|      |   |     |           |
|------|---|-----|-----------|
| 09   | 변조와 위조  |     |           |
| 문제   | 변조(Modification)와 위조(Fabrication)   |     |           |
| 도메인  | 보안  | 난이도 | 중 (상/중/하) |
| 키워드  | 가로채기, 변경, 잘못된 정보, 데이터 생성, 전송  |     |           |
| 출제배경 | 보안 위협 요소에 대한 정확한 개념 확인을 위해 출제   |     |           |
| 참고문헌 | 보안공격( <a href="http://www.ktword.co.kr/test/view/view.php?m_temp1=2288">http://www.ktword.co.kr/test/view/view.php?m_temp1=2288</a> ) |     |           |
| 해설자  | 서경석 기술사(제119회 정보관리기술사 / akslemlf@naver.com)   |     |           |

### I. 원래의 정보가 전달 되지 않는 보안 위협, 변조와 위조의 개념

| 변조   | 위조  |
|--|---|
| - 발신자가 전송할 정보를 중간에 가로채서 정보의 일부 또는 전부를 변경하여 수신자에게 잘못된 정보를 전송하여 통신망을 위협하기 위한 공격 기법 | - 발신자가 전송하지 않은 정보를 생성하여 수신자에게 전송하여 발신자가 잘못된 전송한 것처럼 정보를 전송하여 통신망을 위협하기 위한 공격 기법 |

### II. 변조와 위조의 개념도, 동작 절차 및 대응 방안

#### 가. 변조와 위조의 개념도 및 동작 절차

| 변조   | 위조   |
|--|--|
| <pre> graph LR     A[사용자 A] -- ① --&gt; B[사용자 B]     A -- ② --&gt; C[공격자 2]     C -- ③ --&gt; B           </pre> | <pre> graph LR     A[사용자 A]     B[사용자 B]     C[공격자 1] -- ② --&gt; B           </pre> |
| 1. 사용자 A가 사용자 B에게 데이터 전송   | 1. 공격자가 실제로 전송되지 않은 데이터 생성   |
| 2. 공격자가 A로부터 데이터를 탈취 후 변조  | 2. 위조 생성된 데이터를 사용자 B에게 전송  |
| 3. 변조된 데이터를 사용자 B에게 전송   |  |

- 변조와 위조는 공격 시 실제 전송 데이터 기반 여부로 구분

#### 나. 변조와 위조의 공격 기법 및 대응 방안

| 구분    | 구분 | 설명                         |
|-------|----|----------------------------|
| 공격 기법 | 변조 | - 파일 내용 수정 : 파일 내용 수정 후 전송 |
|       |    | - 프로그램 변경 : 소스 코드 변경 후 반영  |
|       |    | - 전송 메시지 변경 : 오입력 메시지 전송   |

|       |            |                               |
|-------|------------|-------------------------------|
|       | 위조         | - 허위 메시지 삽입 : 메시지 생성, 전송      |
|       |            | - 레코드 추가 : 송수신 데이터 무결성 위협     |
|       |            | - 순서 변경 : 수신 데이터 오류 발생        |
| 대응 방안 | - 전자 서명    | - 원본 데이터 변경 여부 확인             |
|       | - 해쉬 함수    | - 원본 데이터 진위 여부 검증             |
|       | - 접근 통제 모델 | - BLP, BIBA, Clark and Wilson |

- 시스템 안정성 확보를 위해 정보보안의 3대 요소인 CIA 확보 필요

### III. 정보보안 핵심 3요소, CIA

| 구분              | 설명                                   |
|-----------------|--------------------------------------|
| Confidentiality | - 인가된 사용자만 정보 자산에 접근할 수 있도록 하는 성질    |
| Integrity       | - 적절한 권한을 가진 사용자가 인가한 방법으로만 정보 변경 가능 |
| Availability    | - 필요한 시점에 정보 자산에 대한 접근이 가능하도록 유지     |

- 그 외에 추가적으로 인증, 부인 방지 등의 보안 요소 존재

“끝”

|      |   |     |           |
|------|---|-----|-----------|
| 10   | SVM   |     |           |
| 문제   | SVM(Support Vector Machine)                 |     |           |
| 도메인  | 인공지능  | 난이도 | 하 (상/중/하) |
| 키워드  | 초평면, 마진, 서포트벡터, 슬랩, 커널트릭                    |     |           |
| 출제배경 | 데이터 마이닝 기법 중 SVM의 용도, 개념 확인을 위한 출제          |     |           |
| 참고문헌 | ITPE 기술사회 서브노트                              |     |           |
| 해설자  | 서경석 기술사(제119회 정보관리기술사 / akslemlf@naver.com) |     |           |

## I. 여백을 최대화하는 통계적 분석 기법, SVM(Support Vector Machine)의 개요.

### 가. SVM(Support Vector Machine)의 정의

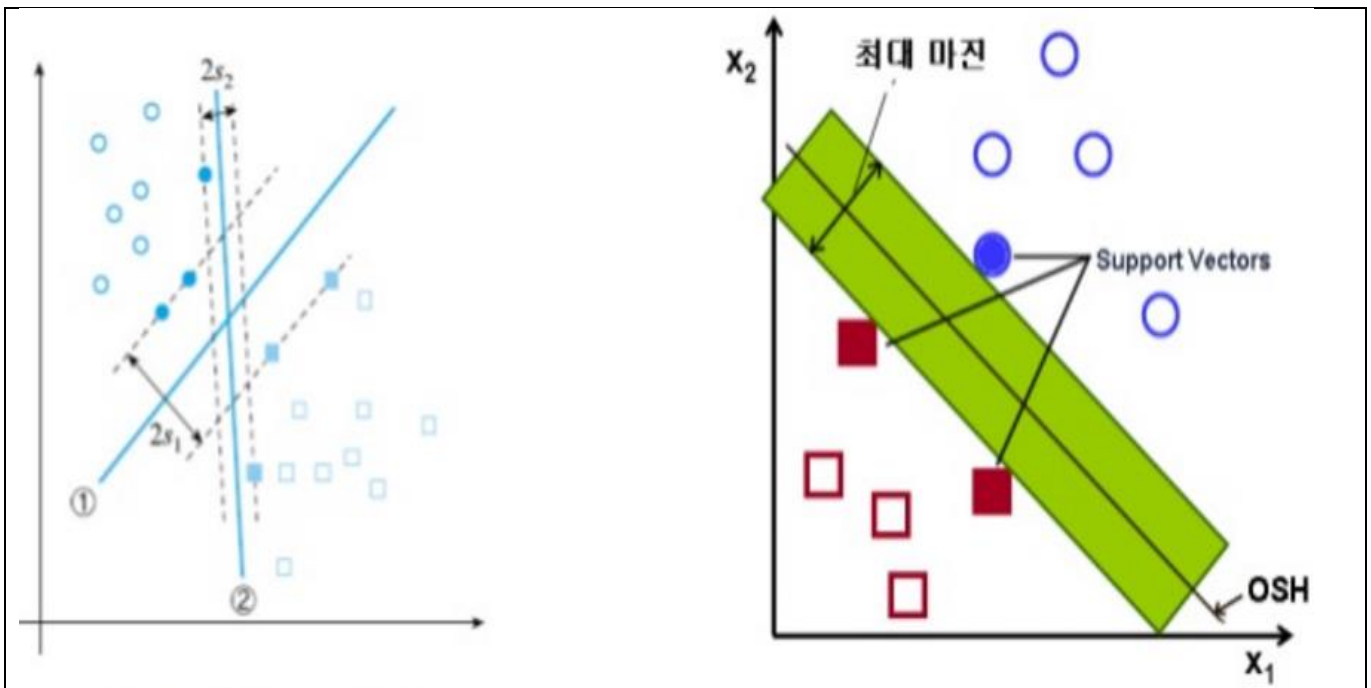
- 데이터가 사상 된 공간에서 경계선과 가장 근접한 데이터(Support Vector)간의 거리가 가장 큰 경계를 식별하는 알고리즘(MMH: Maximum Margin Hyper plane)
- 여백(Margin)을 최대화하여 일반화 능력의 극대화하여 결정직선을 찾는 두 범주를 갖는 객체들을 분류하는 방법

### 나. SVM(Support Vector Machine)의 목적

- 주어진 많은 데이터들을 가능한 멀리 두 개의 집단으로 분리시키는 최적의 초평면(hyperplane)을 Training data에서 두 클래스의 어떤 Point로부터도 최대한 멀리 떨어져 있는 Decision boundary 확보

## II. SVM의 개념도와 구성 요소

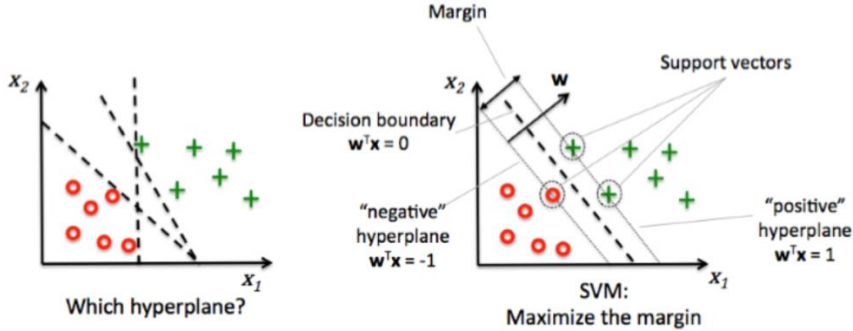
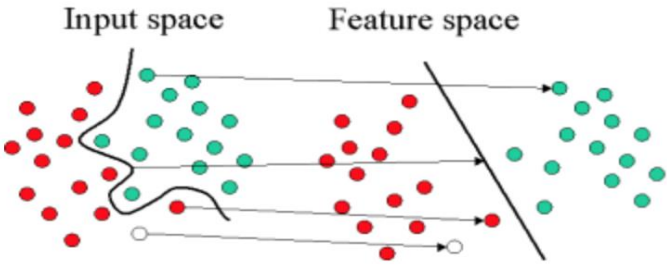
### 가. SVM의 개념도



- 마진을 최대로 하는 분리 초평면을 구하여 데이터를 분류



나. SVM의 구성 요소 및 수행 절차

| 구분    | 핵심 기술               | 설명  |
|-------|---------------------|---|
| 구성 요소 | Support Vector      | - 학습 데이터 중 분류 경계에 가장 가까운 곳에 위치한 데이터   |
|       | Margin              | - 학습 데이터 중 분류 경계에 가장 가까운 데이터와 분류 경계 간 거리  |
|       | 초평면 (hyperplane)    |  <p>- 다차원의 공간의 구분을 위해 결정 되는 n-1평면</p>   |
| 수행 절차 | 커널기법 (Kernel trick) |  <p>- 비선형 패턴 분리 위해 비 선형 패턴의 input space를 선형 패턴의 feature space 로 변환(고차 공간) 하고 해당 비선형 경계면을 찾는 방법</p> |
|       | 데이터 전처리             | - 데이터를 원하는 형태로 가공하는 과정  |
|       | 변수 선택               | <p>- 중요한 변수(독립변수)를 찾는 과정</p> <p>- 너무 많은 독립변수는 정확도에 악영향</p> <p>- 적절한 변수 활용 통한 정확도, 계산속도, 모형 해석 능력 향상</p>   |
| 수행 절차 | 통계적 분류              | - 종속변수 예측 및 종속변수에 영향을 미치는 독립변수를 찾는 과정   |

- 기존 분류기는 오류율을 최소화하는 방법인 반면, SVM은 여백 Margin을 최대화하여 일반화 능력을 극대화

III. SVM의 활용 사례

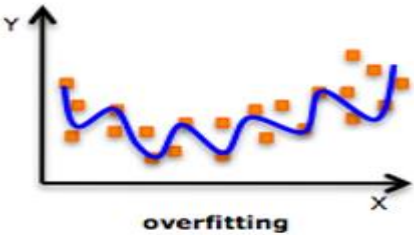
| 사례          | 설명                                      |
|-------------|---|
| 텍스트, 이미지 분류 | - 텍스트와 하이퍼텍스트, 이미지의 특성을 통한 분류, 학습데이터 축소 |
| 의료 정보 분류    | - 분류된 화합물, 의약품에 대한 성분 확인                |
| 문자 인식       | - 손글씨의 특징을 인지(필기체)                      |

- R에서는 LibSVM, SVM-Light 라이브러리 등을 이용하여 SVM의 활용이 가능

“끝”

|      |   |     |           |
|------|---|-----|-----------|
| 11   | 과적합   |     |           |
| 문제   | 과적합(Overfitting)의 발생 이유와 해결방안   |     |           |
| 도메인  | 인공지능  | 난이도 | 하 (상/중/하) |
| 키워드  | 과소적합, 정적적합, 과(이)적합, 차원의저주, 모델의 복잡, Bias-Variance Tradeoff, Dropout, Regularization, Feature Extraction, Feature Selection |     |           |
| 출제배경 |   |     |           |
| 참고문헌 | ITPE 기술사회 서브노트  |     |           |
| 해설자  | 서경석 기술사(제119회 정보관리기술사 / akslemlf@naver.com)   |     |           |

### I. 과잉 학습으로 인한 폐해, 과적합(Overfitting)의 개요

| 개념도  | 정의  |
|--|---|
|  | <ul style="list-style-type: none"> <li>- 기계학습 시 관심집단을 대표할 수 있는 패턴이나 관계를 찾아 훈련 데이터 집합 생성하는 과정에서 너무 학습을 많이 해서 불필요한 내용까지 학습된 상태</li> </ul> |

### II. 과적합의 발생 이유와 해결방안

#### 가. 과적합의 발생 이유

| 구분     | 세부                     | 설명  |
|--------|------------------------|---|
| 모델 측면  | 차원의 저주                 | - 데이터 샘플링을 높이기 위해 고차원 사용시 모델의 복잡도를 증가시키며, 필요한 데이터의 양도 Exponentially하게 증가하며, 기존에 사용하던 Metric이 비정상 동작 |
|        | 모델 복잡도                 | - 정교한 모델이라도 지나치게 복잡하다면, 단순 1차 방정식 형태의 모델보다 성능이 낮은 경우 발생   |
|        | Bias-Variance Tradeoff | - 복잡한 모델은 정교하게 설명할 수 있지만 노이즈에 민감하게 반응. (편차 : Bias, 변동 : Variance)                                   |
| 데이터 측면 | 데이터 불충분                | - 샘플 데이터가 충분하지 못하여 학습 데이터에만 적합  |
|        | Data Skewness          | - 범주 별 데이터셋 분류가 잘못 되어 데이터의 쓸림 현상 발생   |

- 학습 데이터에 대해 과하게 학습하여 실제 데이터에 대한 오차가 증가하는 현상

#### 나. 과적합의 해결 방안

| 구분    | 핵심 기술          | 설명   |
|-------|----------------|--|
| 모델 관점 | Dropout        | - 인공신경망 학습 시, 매번 학습과정에서 입력 또는 히든 뉴런을 랜덤하게 일정 퍼센트를 선택함으로써 모형학습을 단순화       |
|       | Regularization | - 가중치 산출 과정에서 가중치에 패널티 부여하여 상대적으로 덜 중요한 변수에 가중치를 0 또는 근사 조정하여 모형 복잡도 단순화 |

|        |                    |  |
|--------|--------------------|--|
| 특성 관점  | Feature Extraction | - 입력변수가 많은 경우 유사한 변수의 속성을 묶어 제2의 파생변수를 만들어 기존변수를 대체함으로써(차원축소) 모형 단순화 |
|        | Feature Selection  | - 모형학습시 산출되는 변수중요도를 기준으로 상대적으로 중요한 변수를 선택하거나 덜 중요한 변수를 제거해서 모형을 단순화. |
|        | PCA                | - 잡음 제거, 차원 축소   |
| 데이터 관점 | Cross Validation   | - Training/Validation/Test Set                                       |
|        | 추가 데이터 확보          | - 불충분 데이터에 의한 과적합 현상 해결 위한 추가 데이터 확보                                 |

- 데이터 범위 축소, 교차검증, 여러 모델 비교 등을 통한 다양한 관점에서의 과적합 해결 접근

### III. 과적합 방지 이론, 오컴의 면도날

| 구분    | 설명  |
|-------|---|
| 개념    | 어떤 사실 또는 현상에 대한 설명들 가운데 논리적으로 가장 단순한 것이 진실일 가능성이 높다는 건전한 추론을 위한 방법론   |
| 적용 방안 | <ul style="list-style-type: none"> <li>- 복잡한 모델보다는 단순한 모델을 선택(선호)</li> <li>- Variable, Feature 선택 시 중요도가 높은 항목을 선별 사용</li> <li>- 복잡한 비선형 모델보다는 선형적 모델 사용</li> </ul> |
| 사례    | <ul style="list-style-type: none"> <li>- 타이어에 펑크가 났다</li> <li>1. 타이어에 못이 박혔기 때문이다.</li> <li>2. 누군가가 주차장에 들어와 타이어에 구멍을 내고 달아났다.</li> </ul>                           |

- 불필요한 가정을 줄여야 판단 오류 감소 가능성 확보 가능

“끝”

|      |  |     |           |
|------|--|-----|-----------|
| 12   | 디지털트윈, 메타버스  |     |           |
| 문제   | 디지털 트윈(Digital Twin)과 메타버스(Metaverse)  |     |           |
| 도메인  | 디지털서비스   | 난이도 | 하 (상/중/하) |
| 키워드  | 가상세계, 증강현실, 라이프로그, 미래월드, 쌍둥이 모델, 가상모델링, 시뮬레이션, 현실 반영   |     |           |
| 출제배경 | 메타버스 2.0 관련 지식 확인을 위한 출제   |     |           |
| 참고문헌 | 리부트 메타버스(Re-Boot MVS), 2.0시대로의 진화<br>( <a href="https://www.nia.or.kr/site/nia_kor/ex/bbs/View.do?cbIdx=39485&amp;bcIdx=23852&amp;parentSeq=23852">https://www.nia.or.kr/site/nia_kor/ex/bbs/View.do?cbIdx=39485&amp;bcIdx=23852&amp;parentSeq=23852</a> ) |     |           |
| 해설자  | 서경석 기술사(제119회 정보관리기술사 / akslemf@naver.com)   |     |           |

### I. 현실을 반영한 가상 세계 구축 기술, 디지털트윈과 메타버스의 개념

| 디지털트윈(Digital Twin)   | 메타버스(Metaverse)   |
|---|---|
| - 현실 세계에서 사람이 쉽게 접근할 수 없는 공간이나 물건 등 여러 부분들에 대해 가상모델링화 하여 지속적인 관찰과 보수가 가능하게끔 하는 기술 모델. | - '초월, 그 이상' 메타(Meta)와 '세상 또는 우주' 유니버스(Universe)의 합성어로 가상과 현실이 상호작용하며 공진화하고 그 속에서 사회·경제·문화 활동이 이루어지면서 가치를 창출하는 세상 |

### II. 디지털트윈, 메타버스의 기술 요소

#### 가. 디지털트윈의 기술 요소

| 구분            | 세부 기술        | 설명                                      |
|---------------|--------------|---|
| 현실세계 정보 수집 측면 | - Sensor     | - 센서 데이터 및 메타 데이터 수집                    |
|               | - Actuator   | - 절삭/가공에 의한 마모도 및 공정 데이터 수집             |
|               | - Controller | - WPAN, PLC, Modbus, RS-232, 485 등 사용가능 |
| 가상세계 구성 측면    | - 디지털 스레드    | - 물리 자산의 디지털화                           |
|               | - 가상공간 자산    | - 디지털 자산에 대한 설계, 테스트, 피드백 가능            |
|               | - 3D 모델링     | - 전체 공정의 디지털화 가능                        |
| 현실/가상 통합 측면   | - VR/AR 연계   | - 현실세계와 가상세계의 연결 역할                     |
|               | - 양방향 전달     | - 제품에 대한 가상세계에서 다양한 테스트 가능              |
|               | - 공정제어       | - 공정의 시뮬레이션 가능                          |

- 물리적 환경의 디지털화를 통해 현실세계의 특징을 최대한 반영한 가상 시뮬레이션이 가능하며, 이를 통해 다양한 분야에서 활용 가능

#### 나. 메타버스의 기술요소

| 구분   | 핵심 기술                 | 설명   |
|------|-----------------------|--|
| XR   | - 전체 시야 통해 메타버스 정보 표시 | - VR과 AR기술 뿐만 아니라 이들을 혼합한 MR까지도 자유롭게 선택해 활용하여 초실감형 기술 및 서비스 제공 |
| Data | - 데이터기반의 온/오프 라인 연계   | - 현실 환경/자산을 디지털 환경에 접목해 사람과 장소, 사물 간의 복잡한 상호작용을 추적하고 데이터 기반 연계 |

|         |                                 |   |
|---------|---------------------------------|---|
| Network | - 5G의 데이터 전송 속도와 용량 등 적합        | - 클라우드 서버와 장치 간에 엄청난 양의 데이터를 압축할 수 있게 지연 시간이 짧고 빠른 네트워크를 이용한 데이터 전송 |
| AI      | - 다양한 형태의 인공지능 캐릭터 존재 가능        | - 이용자들이 인공지능 캐릭터들과 자연스럽게 소통과 생활이 가능(아바타 현실 존재여부 중요성 감소)             |
| 사운드     | - 소리로 감정을 전달하여 가상세계의 설득력 강화     | - 메타버스 서비스 내에서 현실처럼 옆에 있는 것처럼 전체 감정을 전할 수 있는 3D 오디오 기술로 전달 가능       |
| 블록체인    | - 신뢰성, 안정성이 보장<br>- 플랫폼 안팎에서 사용 | - 디지털화폐 인프라가 갖춰지면 사용자가 창의성을 발휘하고 플랫폼 안팎에서 진정한가치를 얻을 수 있는 기회 확보 가능   |

- 메타버스의 발전을 위해 디지털트윈 기술을 활용하여 메타버스 리부트 전략 수립

### III. 메타버스 활성화를 위한 메타버스 2.0의 개념도 및 주요 이슈

| 구분    | 과제   | 설명   |
|-------|--|--|
| 개념도   | <p style="text-align: center;"><b>&lt; 'Metaverse 2.0' 단계적 발전 방향 &gt;</b></p> <div style="display: flex; justify-content: space-around;"> <div style="border: 1px solid black; padding: 5px; background-color: #0056b3; color: white;">(1단계) 영역별 핵심속성을 중심으로 발전</div> <div style="border: 1px solid black; padding: 5px; background-color: #0056b3; color: white;">(2단계) 다양한 속성이 융합되어 "초세계"로 융합</div> </div> <p>The diagram illustrates the evolution of Metaverse 2.0. It shows a progression from individual domains (XR, SNS, Digital Twin, Simulation) to a unified 'Metaverse 2.0' or '초세계' (Chososegye). The diagram includes labels for '증강(Augmentation)', '가상(Virtual)', and '가상세계 (Virtual Worlds)'. It also mentions '외적인(External)' and '내적인(Internal)' aspects. The final stage shows a circular flow of XR, SNS, Digital Twin, and Simulation, all contributing to the '초세계'.</p> |  |
| 주요 이슈 | - 규제 제도 개선   | - 창작물 저작권 메타버스 기능 제약 개인정보보호 아바타 간 법 윤리적 문제 등 발생 예상       |
|       | - 균형을 이루는 생태계  | - 메타버스 균형적 발전을 위해 XR 뿐만 아니라 가상세계 중심 메타버스 생태계에 집중 노력      |
|       | - 상호 운용성 확보  | - 협의, 표준마련을 통한 상호운용성 확보 필요                               |
|       | - 다양한 연령층 이용   | - 디지털 디바이드 없이(정보·접근 격차 없이) 모든 계층이 이용할 수 있는 메타버스 환경 조성 필요 |
|       | - 지속 가능한 환경  | - 메타버스 시장이 지속 성장할 수 있도록 공공과 민간의 노력 및 혁신 서비스 발굴·활용이 중요    |

- 메타버스를 현 시대에 맞게 정의하고 대응하여 지속 발전 가능하도록 선제적 대응 필요

“끝”

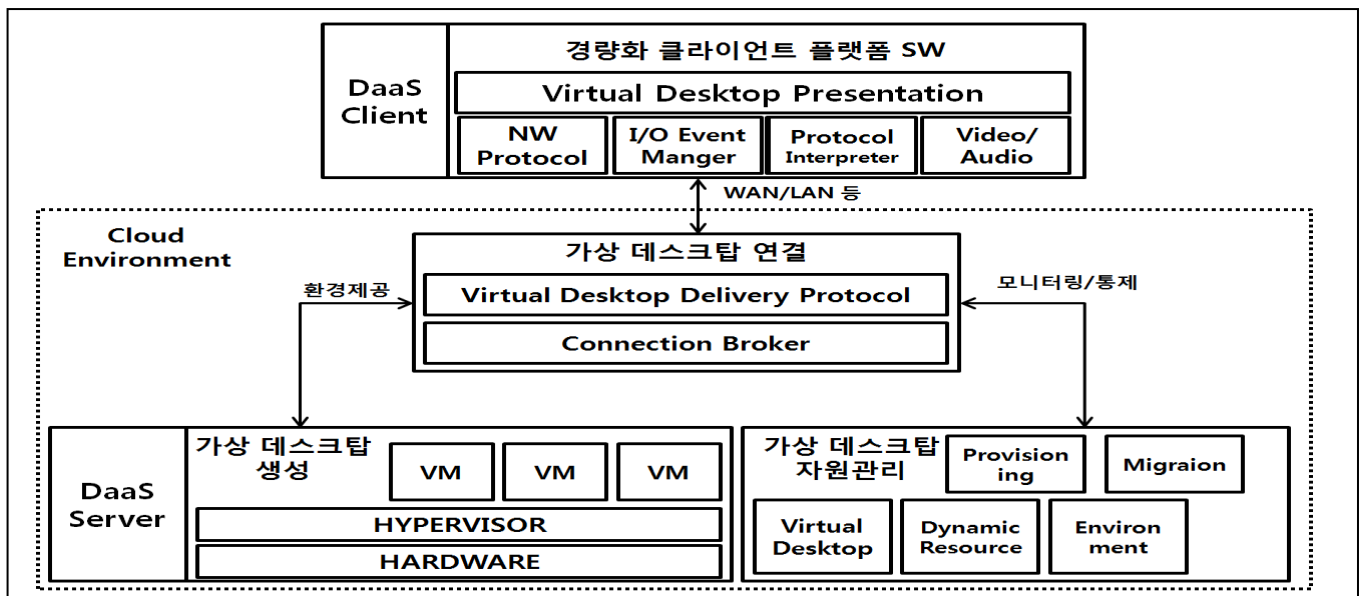
|      |   |     |           |
|------|---|-----|-----------|
| 13   | DaaS  |     |           |
| 문제   | DaaS(Desktop as a Service)  |     |           |
| 도메인  | 디지털서비스  | 난이도 | 하 (상/중/하) |
| 키워드  | 데스크탑 가상화, VDI, SaaS, PaaS,<br>데스크톱 클라이언트, 가상 데스크톱 에이전트, 가상 머신 풀, 데스크톱 연결 브로커<br>접근 게이트웨이, 인증 서버 |     |           |
| 출제배경 | 원격지 개발, 원격 교육 등 활성화에 따른 연관 기술 지식 확인   |     |           |
| 참고문헌 | ITPE 기술사회 서브노트  |     |           |
| 해설자  | 서경석 기술사(제119회 정보관리기술사 / akslemlf@naver.com)   |     |           |

## I. 클라우드 기반 데스크톱 가상화 기술, DaaS의 개요

|  |        |      |     |  |
|--|--------|------|-----|--|
| <div> <div>Desktop 가상화</div> <div>사용자PC 환경을 가상공간에 구축하여 시간/단말 환경 극복</div> </div> <div>클라우드</div> <div>집적·공유된 IT자원을 사용자의 요구에 따라 신축적으로 이용</div> | 유기적 통합 | DaaS | 정의  | - 데스크탑 가상화(VDI, Virtual Desktop Infrastructure)를 클라우드 기술로 구현, 서비스 공급자가 아웃소싱 형태로 제공하는 서비스 |
|  |        |      | 필요성 | 1. 필요 자원의 공급 유연화, 확장성 증대<br>2. 긴급 상황에 대비한 접근성 확보<br>3. VDI 편의측면 문제점 해결, ROI효율 확대         |

## II. DaaS의 구성도 및 구성요소

### 가. DaaS의 구성도



- 가상 데스크탑을 제공하는 서버 및 가상 데스크탑의 상태를 실시간 모니터링 하고 관리자에게 GUI를 제공하며 구동 제어 기능 제공

## 나. DaaS의 구성 요소

| 구분                           | 핵심 기술                  | 설명                                   | 기술            |
|------------------------------|------------------------|--------------------------------------|---------------|
| Virtual Desktop Presentation | - NW Protocol          | - 네트워크 객체간 <u>규격/순서 등을 정의한</u> 통신 규약 | - OSI 7 Layer |
|                              | - I/O Event Manager    | - IRP생성 및 관련 드라이버 제어 정보 관리           | - Device 연계   |
|                              | - Protocol Interpreter | - <u>커넥션 생성</u> 및 커맨드/응답 생성 및 송수신    | - FTP         |
|                              | - Video/Audio          | - 영상/음성 출력                           | - Display     |
| Desktop 연결                   | - Delivery Protocol    | - 가상데스크톱 환경 연동, <u>보안/성능/가용성</u> 확보  | - 리다이렉션       |
|                              | - Connection Broker    | - 데스크톱 환경 연결, 기능 관리 / 모니터링           | - 신규/추가       |
| Desktop 생성                   | - VM                   | - 사용환경 가상화 (수행환경, 메모리, 인터페이스 등)      | - 유저환경        |
|                              | - Hypervisor           | - OS를 물리적 하드웨어와 분리, <u>동시실행환경</u> 구성 | - IF제공        |
|                              | - Hardware             | - 가상화 구현 기반 물리적 장비                   | - CPU/메모리     |
| 자원 관리                        | - Provisioning         | - 요청에 따른 데스크탑환경 제공 및 관리              | - 권한설정        |
|                              | - Migration            | - 데스크탑 상세 환경 구성/이관                   | - Version     |
|                              | - Virtual Desktop      | - 제공 가상화 데스크탑 환경                     | - 개별VM연동      |
|                              | - Dynamic Resource     | - 제공 환경의 <u>자원 다변화</u> 구성            | -제공프로그램       |
|                              | - Environment          | - 자원통합 관리 및 모니터링/통제                  | - 가용메모리       |

- 가상데스크탑 엔진과 전송프로토콜을 독립적으로 적용 가능하며, 기존 환경에 비해 TCO절감 부분 다수 존재

## III. DaaS의 활용 분야

| 활용 분야    | 적용         | 설명                               |
|----------|------------|----------------------------------|
| 스마트워크/교육 | - 다수 환경 제공 | - 초기구성상태유지, 자료 중앙저장, 목적에 따른 환경설정 |
| 망분리/보안   | - 업무망 분리   | - 중앙화된 패치 및 업그레이드, 통합 정책 표준화     |
| 생산성강화    | - 근무지 외 접근 | - 장소/시간 제약 해결, 원격근무, 재택근무, 비상대응  |

- 공공, 기업 뿐만 아니라 교육, 헬스케어, 제조, 개인고객 시장으로 DaaS서비스 확대

“끝”



## ITPE 기술사회

### 제127회 정보관리기술사 기출문제 해설집

|     |   |
|-----|---|
| 대 상 | 정보관리기술사, 컴퓨터시스템응용기술사, 정보통신기술사, 정보시스템감리사 시험  |
| 발행일 | 2022년 04월 16일   |
| 집 필 | 강정배PE, 안응원PE, 서경석PE, 이상헌PE  |
| 출 판 | <b>ITPE(Information Technology Professional Engineer)</b>   |
| 주 소 | ITPE 대치점 서울시 강남구 선릉로 86길 17 선릉애틀빌딩 7층<br>ITPE 선릉점 서울시 강남구 선릉로 86길 15 3층 IT교육센터 아이티피이<br>ITPE 강남점 서울시 강남구 테헤란로 52길 21 파라다이스벤처타워 3층 303호<br>ITPE 영등포점 서울시 영등포구 당산동2가 하나비즈타워 7층 ITPE |
| 연락처 | 070-4077-1267 / <a href="mailto:itpe@itpe.co.kr">itpe@itpe.co.kr</a>  |

본 저작물은 [ITPE\(아이티피이\)](http://itpe.co.kr)에 저작권이 있습니다.

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