



01

개방형 클라우드 플랫폼, 파스-타란 무엇인가



1) 공공 클라우드 기반 플랫폼



세종사이버대학교

개방형클라우드플랫폼 ‘파스-타’ 정부기관 공표

디지털 정부혁신 추진계획

2019. 10. 29.

관계부처 합동

3. 디지털 서비스 전문계약 활성화 (201분기 전문계약 제도 마련)

○ 20.1분기까지 각급기관이 참고할 수 있는 디지털 서비스 전문계약 제도 마련(국가계약법 하위법령 개정으로 명확한 근거 제공)

- △ 중앙재, △ 장기계약, △ 서비스 상세협상, △ 가변요금 방식 등 디지털 서비스 이용계약에 필요한 특성이 잘 반영될 수 있도록 설계
- 전문계약 제도의 효율적 운영을 위해 전문 유통플랫폼 운영(20년~)

- 15 -

<영국 클라우드 전문계약제도 운영 성과>

○ 영국은 클라우드 우선 전략에 따라 클라우드 전문계약 제도(Cloud Framework)를 마련하고, 12년부터 전문 유통플랫폼(Digital Marketplace)을 운영중

- 12 → 18년까지 참여기업수가 258 → 3,505개, 등록서비스수가 1,700 → 24,543개, 계약금액이 104억 → 1조 7,421억원(※ 공공 ICT예산의 약 1/5 수준)으로 증가
- (대기업연중 시장) 중소기업이 계약건수액 71%, 계약금액의 45%를 차지 (지역연중 시장) 시장규모 12개 기업에 집중 → 영국 전체에서 4천여개 기업 참여

중장기 범정부 디지털 전환 로드맵 수립

○ 6대 우선 추진과제의 조속한 추진을 통해 낡은 체제를 극복할 계기를 마련하고, 중장기 관점의 디지털 전환을 위한 로드맵 수립(20년)

로드맵 핵심 내용

① 공공서비스 주주기 혁신	② 데이터-클라우드-인공지능 선순환 전략
③ 데이터중심 정보보호 체제	④ 개방형 디지털정부 생태계 조성
⑤ 디지털 전환 제도·절차 및 추진조직 정비, 핵심인재 확보, 공무원 역량강화	

5 클라우드와 디지털서비스 이용 활성화

○ (민간클라우드 이용확대) 안보·수사, 내부시스템을 제외한 전체 시스템을 이용대상으로 확대하고, 내부시스템도 부처협의를 거쳐 허용

○ (개방형 플랫폼) AI, 클라우드 등을 활용한 서비스를 쉽게 개발·운영할 수 있도록 개방형 전자정부 클라우드 플랫폼 구축

1 민간 클라우드 이용범위 대폭 확대 (19.下 지침 개정)

행안부·국정원·과학기술부 등

○ (현재) △ 안보·수사, △ 개인정보영향평가 대상 정보, △ 중앙부처·지자체의 내부시스템은 민간 클라우드 이용 제외

○ (개선) ① △ 안보·수사, △ 내부시스템을 제외한 전체 시스템을 대상으로 민간 클라우드 이용 허용

- ② 내부 업무시스템 등도 관계부처 협의를 거쳐 민간 클라우드 이용 허용

2 개방형 전자정부 클라우드 플랫폼 구축 (19.下 착수)

행안부·과학기술부

○ 개방형 전자정부 클라우드 플랫폼* 구축(19.下~)에 착수, '22년 국가정보자원관리원 대구센터에 우선 적용, 대전·광주센터로 확산

- AI 등을 활용한 응용서비스를 클라우드 기반으로 손쉽게 개발·실행하고 오픈소스로 공유

- 기능개선과 기술지원을 전담할 개방형 클라우드 플랫폼 센터* 설치(20년)

- 공공과 민간, 글로벌과 국내 오픈소스 커뮤니티를 연결하는 구심점으로 육성

1) 공공 클라우드 기반 플랫폼



세종사이버대학교



‘공공 클라우드 플랫폼(파스-타)’본격 구동

과학기술정보통신부와 한국정보화진흥원은 8.12일
대한상공회의소에서 개방형 클라우드 플랫폼 센터를 개소



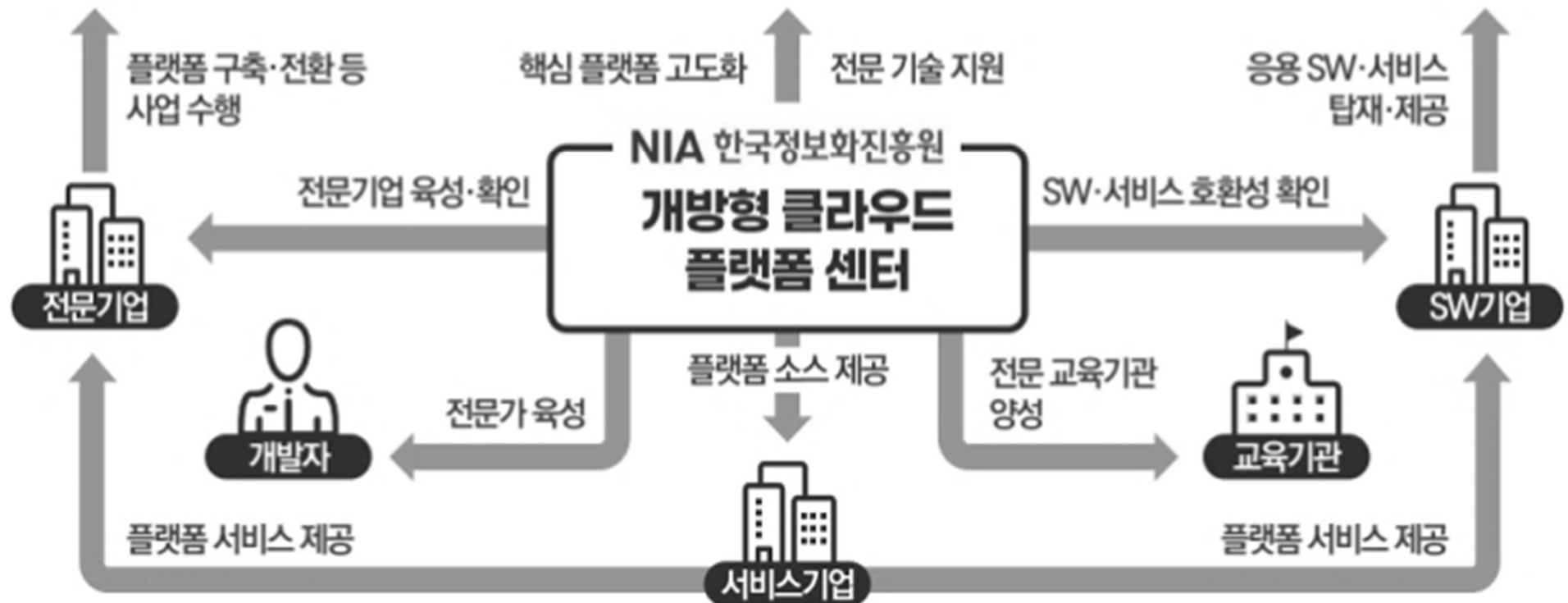
공공 클라우드 플랫폼 관계 기술 표준 개발



공공/민간 전 영역에서 국산 클라우드 플랫폼(파스-타,
PaaS-TA) 확산



수요기관(공공기관)을 위한 고도화 및 기술지원 수행 등





2) 개방형 클라우드 플랫폼, 파스-타는 무엇?



‘공공 클라우드 플랫폼(파스-타)’본격 구동

파스-타는 전자정부 서비스 개발 및 제공을 위한
클라우드 서비스 플랫폼 기반임

다음과 같은 기반과 밀접함

- 전자정부 표준프레임워크 기반
- 정부3.0 실현 지원
- 오픈소스 PaaS기반
- 창조경제 실현 지원

이를 통해서 공공기관의 전자정부 서비스들과
밀접한 기반으로 제공되기 위해 준비함



전자 정부 표준프레임 워크 기반

전자정부 표준프레임워크 확장 개발

표준프레임워크
eGovFrame

오픈소스 PaaS 기반

오픈 소스 활용 개발 ▶ 결과물 오픈소스로 공개



정부 3.0 실현 지원

IT자원의 공동 활용 촉진

행복한
대한민국을 여는
정부 3.0

IT자원의 공동 활용 촉진



창조경제 실현 지원

창조경제를 실현하는 핵심 인프라 제공



2) 개방형 클라우드 플랫폼, 파스-타는 무엇?



세종사이버대학교

PaaS-TA

클라우드 인프라 환경을 제어하면서 애플리케이션을 쉽게 개발하고 안정적으로 운영할 수 있도록 관리 · 지원하는 개방형(오픈소스 기반의) 클라우드 플랫폼



PaaS-TA
OpenPaaS

3) PaaS-TA 단계별 추진 준비(2014년~현재)



세종사이버대학교



IaaS, PaaS, SaaS 통합 모니터링 기능 확장 개발,
Application Gateway 기능 개발



IaaS, PaaS, SaaS 통합 모니터링 기능 개발,
모니터링 기반 오토스케일링 기능 개발



이종 클라우드 기반의 개방형 PaaS 플랫폼 지원 검증



응용의 전 주기 라이프 사이클 관리 기능 개발,
PaaS상에 동작하는 응용을 위한 마켓플레이스 기능 개발

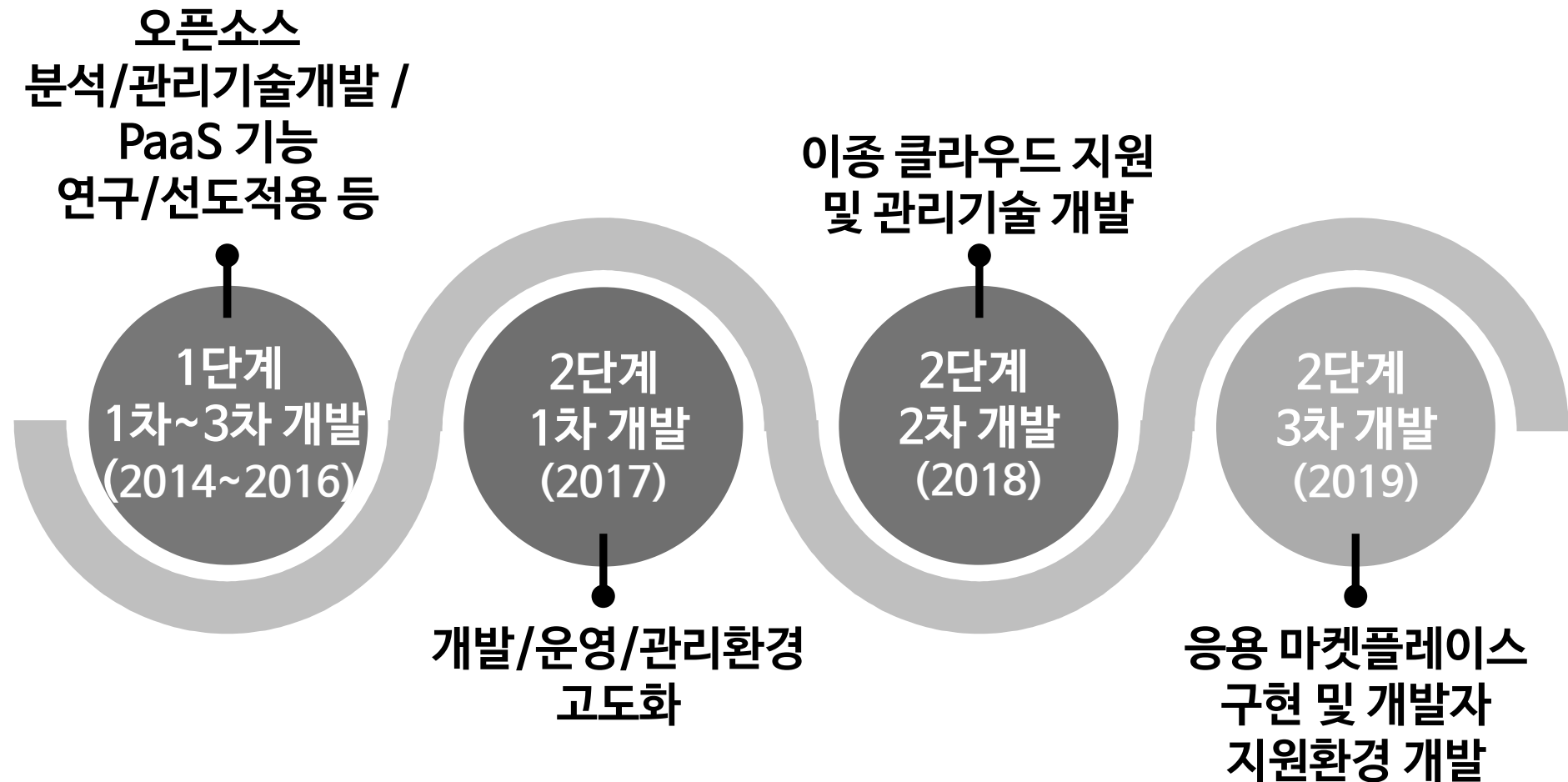


응용의 사용량 측정 기능 고도화, On-Demand 서비스
설치 개발

3) PaaS-TA 단계별 추진 준비(2014년~현재)



세종사이버대학교





02

중요 용어 및 개발 개념 이해

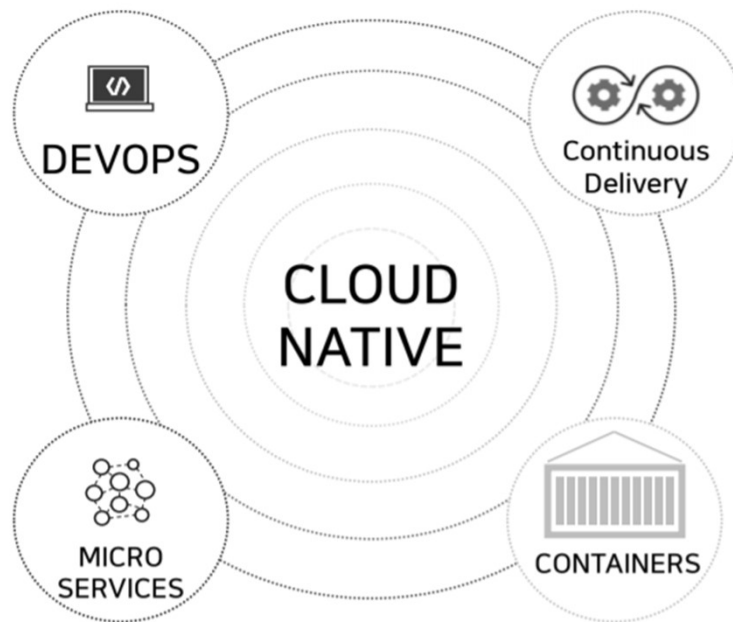


1) 클라우드 네이티브 응용 프로그램



세종사이버대학교

“ Cloud 환경에 최적화 되어
서비스 되도록 개발된 어플리케이션 ”



1) 클라우드 네이티브 응용 프로그램



클라우드 컴퓨팅 모델의 장점을 활용 할 수 있도록
응용 프로그램을 개발하고 실행하기 위한 접근 방식



클라우드 환경에서 효과적으로 작동



클라우드의 특징을 활용

- 탄력성
- 사용량 기반 과금
- 장애인지

1) 클라우드 네이티브 응용 프로그램



세종사이버대학교

Cloud Native Application의 핵심은 ‘서비스’



어플리케이션을 여러 개의 서로 독립적인 기능을 하는 서비스로 구분(마이크로 서비스)



서비스들을 어떻게 구성하고 어떻게 연결하고 어떻게 관리하느냐가 관건



‘서비스’들을 묶어서 하나의 통합된 ‘(비즈니스) 서비스’를 할 수 있도록 하기 위한 다양한 기능과 기술 필요

2) Cloud Native Application의 개발 원칙



세종사이버대학교

중요 원칙



무상태 (Stateless)

독립적/자율적 (Bind/분산DB지향)

느슨한 결합 (http, Rest)

CI/CD (DevOps, 자동화)

2) Cloud Native Application의 개발 원칙



세종사이버대학교

개발 방법론 : 12 Factors

01 Codebase

02 Dependencies

03 Config

04 Backing Services

05 Build, Release, Run

06 Stateless Processes

07 Port Binding

08 Concurrency

09 Disposability

10 Dev/Prod Parity

11 Logs

12 Admin Processes

3) Cloud Native Application의 주요 특징

1/2

구분	전통적인 어플리케이션 아키텍처의 특징	클라우드 네이티브 어플리케이션 아키텍처의 특징
확장성	Scale Up (확장성을 위해 서버 등 자원을 대체하여 성능을 향상)	Scale Out (확장성을 위해 업무를 여러 서버로 분산하여 처리)
구조	단일구조 & 계층형 (Monolithic & Layered)	분산 & Microservices
상태관리	Stateful(고정적인)	Stateless(유동적이고 일시적인)
확장성	인프라 종속적 & 고정 인프라/고정된 용량	인프라와 무관 & 유연한 인프라/유연한 용량
종속성	대기시간 불허용, 단단한 결합	대기시간 허용, 느슨한 결합

3) Cloud Native Application의 주요 특징

2/2

구분	전통적인 어플리케이션 아키텍처의 특징	클라우드 네이티브 어플리케이션 아키텍처의 특징
관계형	Consolidated/Clustered DB/Rich/Chatty Client	Sharded/Replicated/Distribut ed DB/Mobile/Thin Client
라이선스	상업 라이선스	오픈소스/지원 라이선스
가용성	인프라 지원 가용성 - 인프라 중복	앱 지원 가용성 - 복원력
복구/관리	수동 빌드/배치, 수동 오류 복구	자동 복구
백업/이중화	Active/Passive/DR	Active/Active

4) 파스-타 전환 시 기존 서비스가 통자구조인 경우 문제점



세종사이버대학교

마이크로 서비스로 되어 있지 않은
규모가 큰 기존 애플리케이션은 불리함

빌드, 배포, 서버 기동 시 시간이 오래 걸림

컴포넌트들이 서로 로컬 콜(Call-by-reference) 기반으로
강하게 결합(Tightly Coupled)함

└→ 특정 컴포넌트나 모듈에서의 성능 문제나 장애가 다른
컴포넌트에까지 영향을 미침

코드가 너무 커져서 유지보수 어려움

4) 파스-타 전환 시 기존 서비스가 통자구조인 경우 문제점



세종사이버대학교

배포가 잦은 시스템에 불리함

사소한 컴포넌트의 수정인데도 전체 어플리케이션을
재컴파일하여 배포를 하고, QA를 거쳐야 함

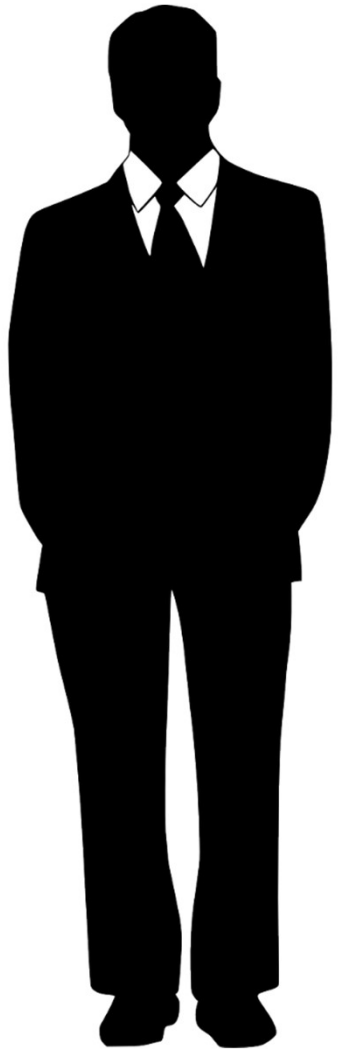
모듈화 되어 있는 것들은
어느 정도 파스타 컨테이너로 분리하는게 덜 무리가 됨

BUT → 통자로 되어 있는 것은 그대로 올리려고 할 때 파스타로
올리기는 어려움

5) 마이크로 서비스(Microservices)



세종사이버대학교



마틴 파울러(Martin Fowler)

“마이크로 서비스 아키텍처 스타일이란
단일 어플리케이션을 자체 프로세스로 실행되고
경량 매커니즘(주로 HTTP 리소스 API)으로 통신하는
작은 서비스들의 모음으로 개발하는 방식이다”

5) 마이크로 서비스(Microservices)



세종사이버대학교

마이크로 서비스로
개발 시

개선/기능 추가가 용이함

마이크로 서비스로
배포 시

전체를 배포하지 않고
변경 부분만 배포하여
유연하게 변경관리 가능

예

넷플릭스의 경우 마이크로 서비스로 개발/배포/운영 진행 중

5) 마이크로 서비스(Microservices)



세종사이버대학교

아키텍처 구성하기 위한 핵심 요소

서비스

DevOps

데이터 분리

API
Gateway

- 각 컴포넌트는 서비스라는 형태로 구현되고 API를 이용하여 타 서비스와 통신
- 서비스 경계는 구문 또는 도메인(업무)의 경계를 따름
 - 예 사용자 관리, 상품 관리, 주문 관리와 같은 각 업무 별로 서비스를 나눠서 정의
- REST API에서 /users, /products와 같이 주요 URI도 하나의 서비스

5) 마이크로 서비스(Microservices)



아키텍처 구성하기 위한 핵심 요소

서비스

DevOps

데이터 분리

API
Gateway

- DevOps는 CI에서 좀더 진화된 형태
- 개발, 테스트, 배포를 모두 자동화 시켜 개발 사이클이 끊임없이 순환되도록 함으로서 개발의 속도를 최대화 시키는 개발스타
- 배포가 서비스의 수 만큼 이루어지게 될 뿐만 아니라 테스트 또한 각각의 서비스가 연동되어 발생하는 집합체 Aggregate의 수 만큼 필요하게 되므로 필연적으로 DevOps 필요

5) 마이크로 서비스(Microservices)



세종사이버대학교

아키텍처 구성하기 위한 핵심 요소

서비스

DevOps

데이터 분리

API
Gateway

- 서비스 별로 필요에 따라 별도의 데이터베이스를 사용
- 서비스가 API에서부터 데이터베이스까지 분리되는 수직 분할 원칙 (Vertical Slicing)에 따름
- 데이터베이스의 종류 자체를 다르게 하거나, 같은 데이터베이스를 사용하더라도 스키마를 나누는 방법 사용

5) 마이크로 서비스(Microservices)



아키텍처 구성하기 위한 핵심 요소

서비스

DevOps

데이터 분리

API
Gateway

- 모든 api에 대한 End Point를 통합하고, 몇가지 추가적인 기능을 제공하는 미들웨어
- EndPoint 통합과 토폴로지 정리
- Orchestration
 - 여러 서비스를 묶어서 하나의 서비스 생성
- 공통 기능 처리(Cross Cutting Function Handling)
 - API 인증(Authentication), Logging 등

5) 마이크로 서비스(Microservices)



아키텍처 구성하기 위한 핵심 요소

서비스

DevOps

데이터 분리

API
Gateway

- Mediation
 - 메시지 포맷 변환, 프로토콜 변환, 메시지 라우팅 등

5) 마이크로 서비스(Microservices)



세종사이버대학교

적용 시 고려사항

▶ 서비스 범위 설정 문제

어디서부터 어디까지를 묶어야
독립적으로 운영 가능한 서비스가 되는가?

AMI(Amazon Machine Image)

- 마이크로 서비스를 도입하더라도 (일정 기간은) 기존의 시스템들과의 공존이 필연적으로 존재할 수밖에 없는 상황에서 기존 시스템들과 어떻게 연계할 것인가?

5) 마이크로 서비스(Microservices)



적용 시 고려사항

▶ 운영 오버헤드

마이크로 서비스는 엄청나게 많은 양의 배포작업 작업들을
자동화할 수 있도록 DevOps 도입

인터페이스 불일치

- 각각의 서비스의 인터페이스를 변경하는 것에 대한 영향범위를 파악?
- 서비스 외부로 제공하는 인터페이스가 의도하지 않은 형태로 사용되지 않도록 할 것인가?
- 전체 시스템의 인터페이스 맵을 만들고 팀 간 커뮤니케이션 비용을 효과적으로 제어할 수 있는가?



03

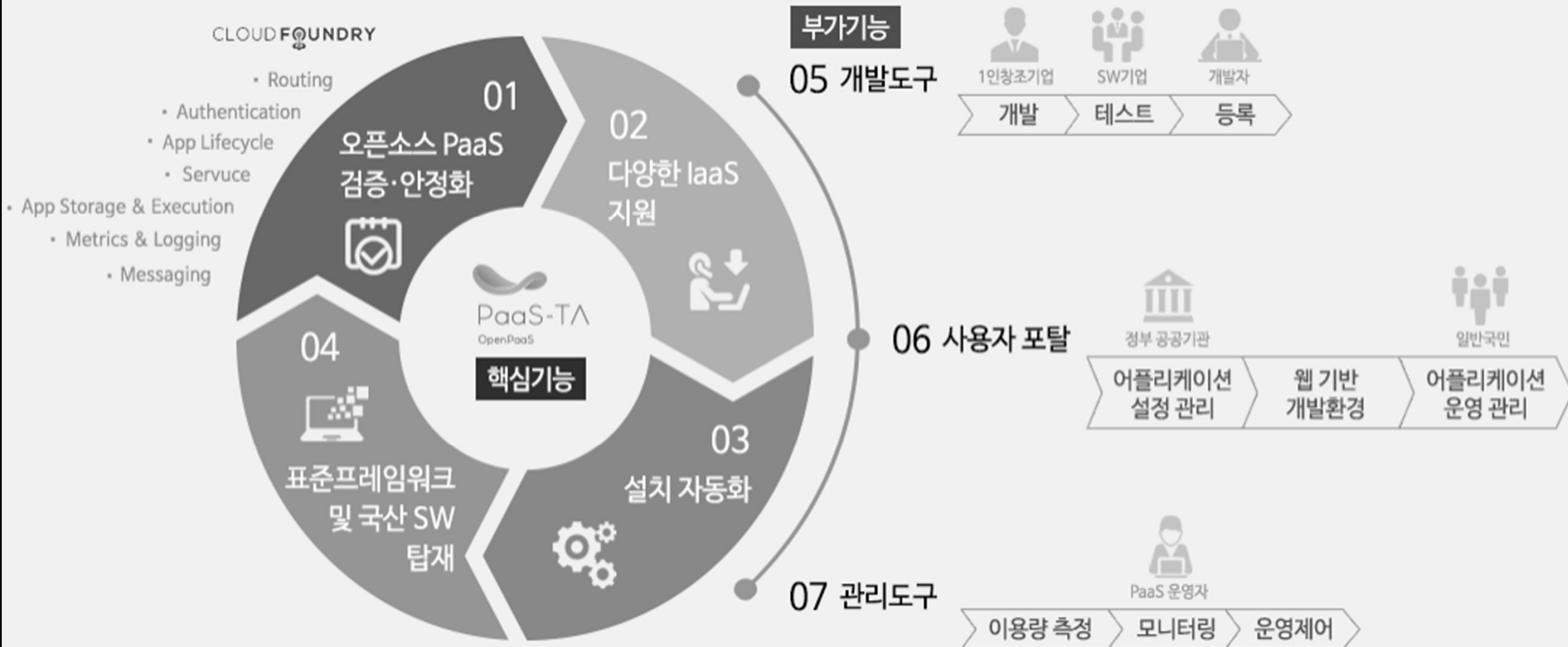
파스-타 클라우드 플랫폼의 주요 특징



1) 파스-타 플랫폼 핵심기능



세종사이버대학교



IaaS > PaaS > SaaS : 개발/관리자가 작업해야 하는 분량이 더 많음



2) 파스-타 플랫폼 특징

인프라 종속 탈피

국산 SW서비스 지원

풍부한 개발 운영 환경

CF · 8s 듀얼 통합 환경

자동확장 및 장애 복구

통합 모니터링

마켓플레이스

11종의 다양한
인프라(IaaS) 지원



2) 파스-타 플랫폼 특징



세종사이버대학교

인프라 종속 탈피

국산 SW서비스 지원

풍부한 개발 운영 환경

CF·8s 듀얼 통합 환경

자동확장 및 장애 복구

통합 모니터링

마켓플레이스

우수한 국산 솔루션 탑재와
학산 촉진



2) 파스-타 플랫폼 특징



세종사이버대학교

인프라 종속 탈피

국산 SW서비스 지원

풍부한 개발 운영 환경

CF · 8s 듀얼 통합 환경

자동확장 및 장애 복구

통합 모니터링

마켓플레이스

풍부한 개발 환경 ·
데브옵스 도구 지원

개발언어
9종

미들웨어
17종

프레임워크
5종

TOOL
5종

2) 파스-타 플랫폼 특징



세종사이버대학교

인프라 종속 탈피

국산 SW서비스 지원

풍부한 개발 운영 환경

CF · 8s 듀얼 통합 환경

자동확장 및 장애 복구

통합 모니터링

마켓플레이스

Cloud Foundry와 Kubernetes 선택형 통합 개발 지원



2) 파스-타 플랫폼 특징



세종사이버대학교

인프라 종속 탈피

국산 SW서비스 지원

풍부한 개발 운영 환경

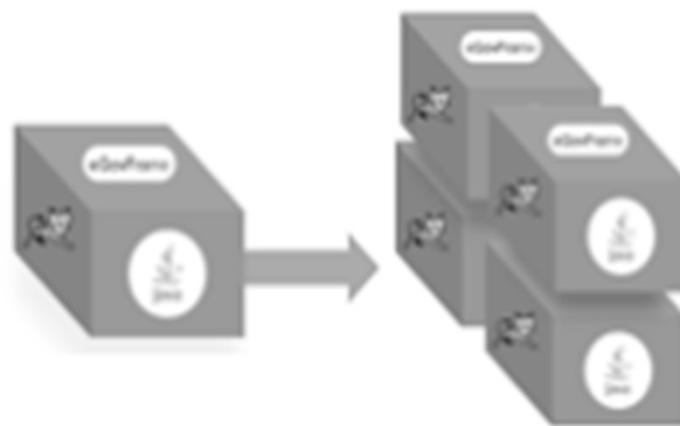
CF·8s 듀얼 통합 환경

자동확장 및 장애 복구

통합 모니터링

마켓플레이스

VM, 컨테이너를 사용량에 따라
자동확장과 장애 복구



2) 파스-타 플랫폼 특징



세종사이버대학교

인프라 종속 탈피

국산 SW서비스 지원

풍부한 개발 운영 환경

CF·8s 듀얼 통합 환경

자동확장 및 장애 복구

통합 모니터링

마켓플레이스

인프라부터 응용서비스까지
통합모니터링



2) 파스-타 플랫폼 특징



세종사이버대학교

인프라 종속 탈피

국산 SW서비스 지원

풍부한 개발 운영 환경

CF·8s 듀얼 통합 환경

자동확장 및 장애 복구

통합 모니터링

마켓플레이스

사용량 기반 미터링, 구매관리,
상품 카탈로그 등 지원

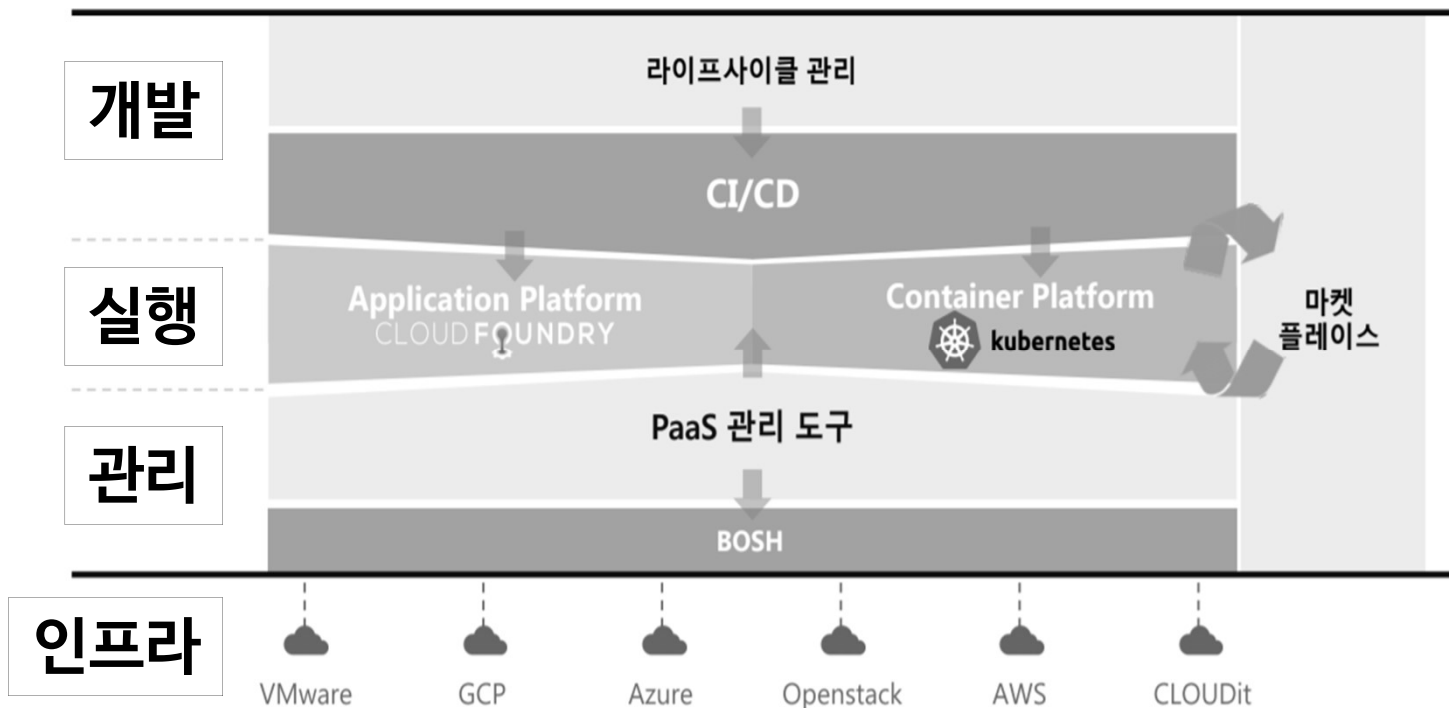


2) 파스-타 플랫폼 특징



세종사이버대학교

파스-타 플랫폼에서는 관리도구를 제공하여
관리/실행/개발을 지원하는 PaaS를 제공



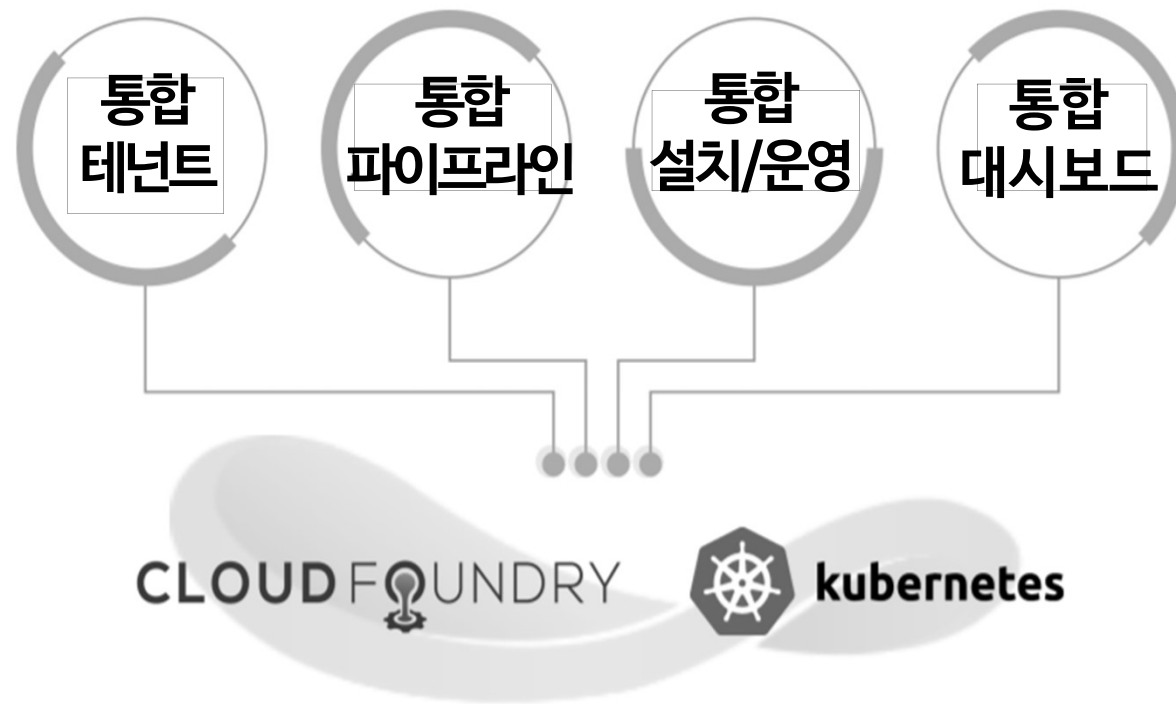
11종의 다양한 IaaS 지원 ➡ 계속 추가 중

2) 파스-타 플랫폼 특징



세종사이버대학교

자동화된 서비스 확장(Auto Scaling)



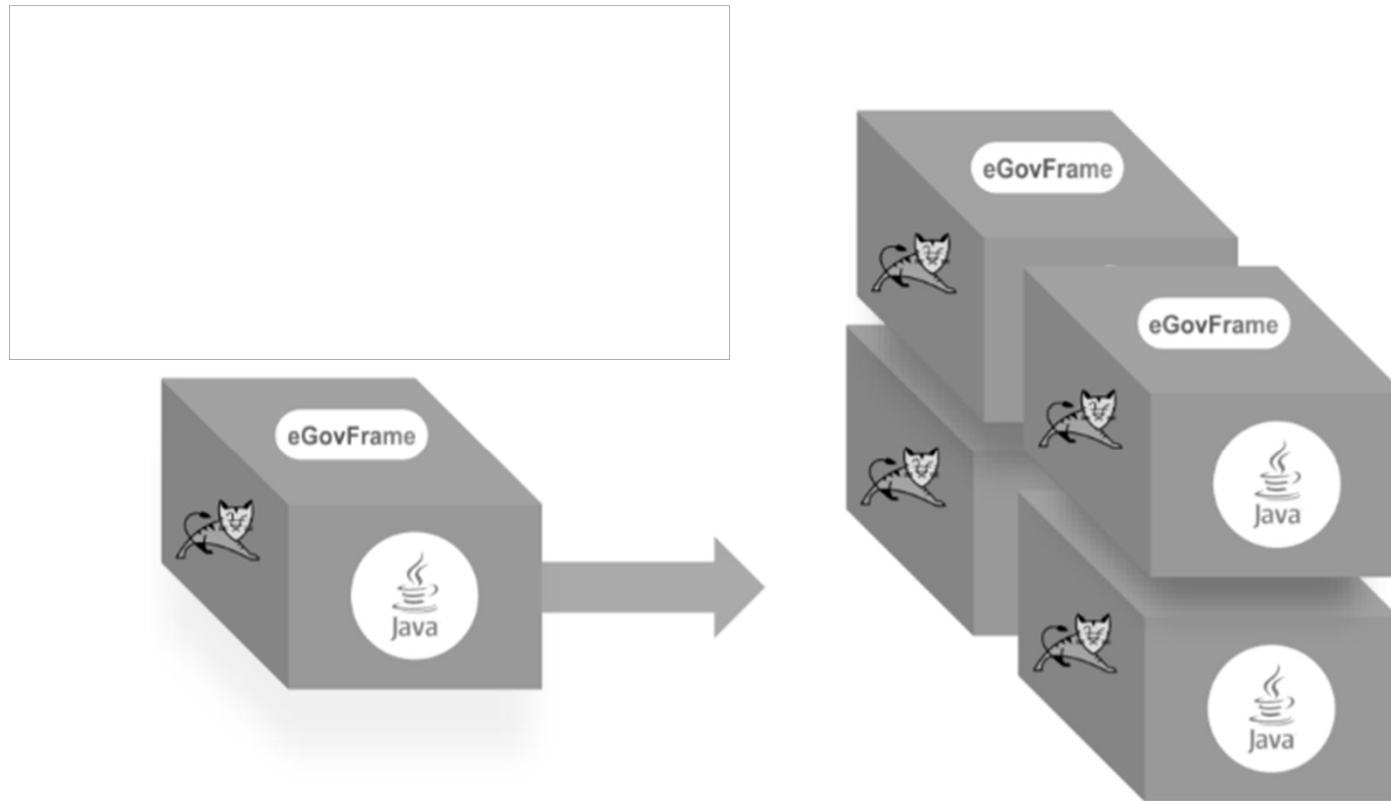
Cloud Foundry와 Kubernetes를 모두 제공하는
일관된 개발 프로세스 구현

2) 파스-타 플랫폼 특징



세종사이버대학교

자동화된 서비스 확장(Auto Scaling)





3) 파스-타 플랫폼 도입사례

수요
기간

다양한 파스-타 도입 기관과 사례

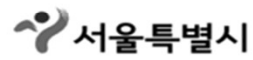
NIA 한국정보화진흥원

자체

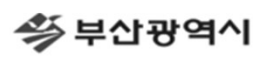
전자정부 클라우드 플랫폼



차세대 업무관리시스템



부산시 자체 IDC 플랫폼



한국전력 HUB-Pop



기후정보서비스 플랫폼



조선·해양부문 R&D
클라우드 플랫폼



대민정보시스템



미래교통운영
기술개발 시스템



퍼블릭

ICT 클라우드 플랫폼



국가 학술정보 시스템



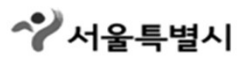
공유대학 플랫폼



문화재 안전관리시스템



교량안전관리 서비스



축제 통합관리 플랫폼



AI 의료영상 분석
보조서비스



공적마스크 앱·웹



3) 파스-타 플랫폼 도입사례



세종사이버대학교



교량안전관리
서비스



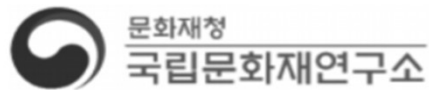
<https://seoul-web.ktpaasta.kr>



국가학술정보
융합데이터
지식 서비스



<http://cloud.nanet.go.kr/main.do>



문화재 자동계측
안전관리 시스템



<https://heritage-web.ktpaasta.kr>



공유대학플랫폼



<https://home.sfup.or.kr>



04

개방형 클라우드 플랫폼 사례



1) 대표사례



세종사이버대학교

마스크 웹/앱

▶ 파스-타 기반으로 구축 : 한국정보화진흥원 주관



파스-타 얼라이언스가 파스-타 플랫폼 제공



시민개발자/스타트업이 마스크 웹/앱 개발

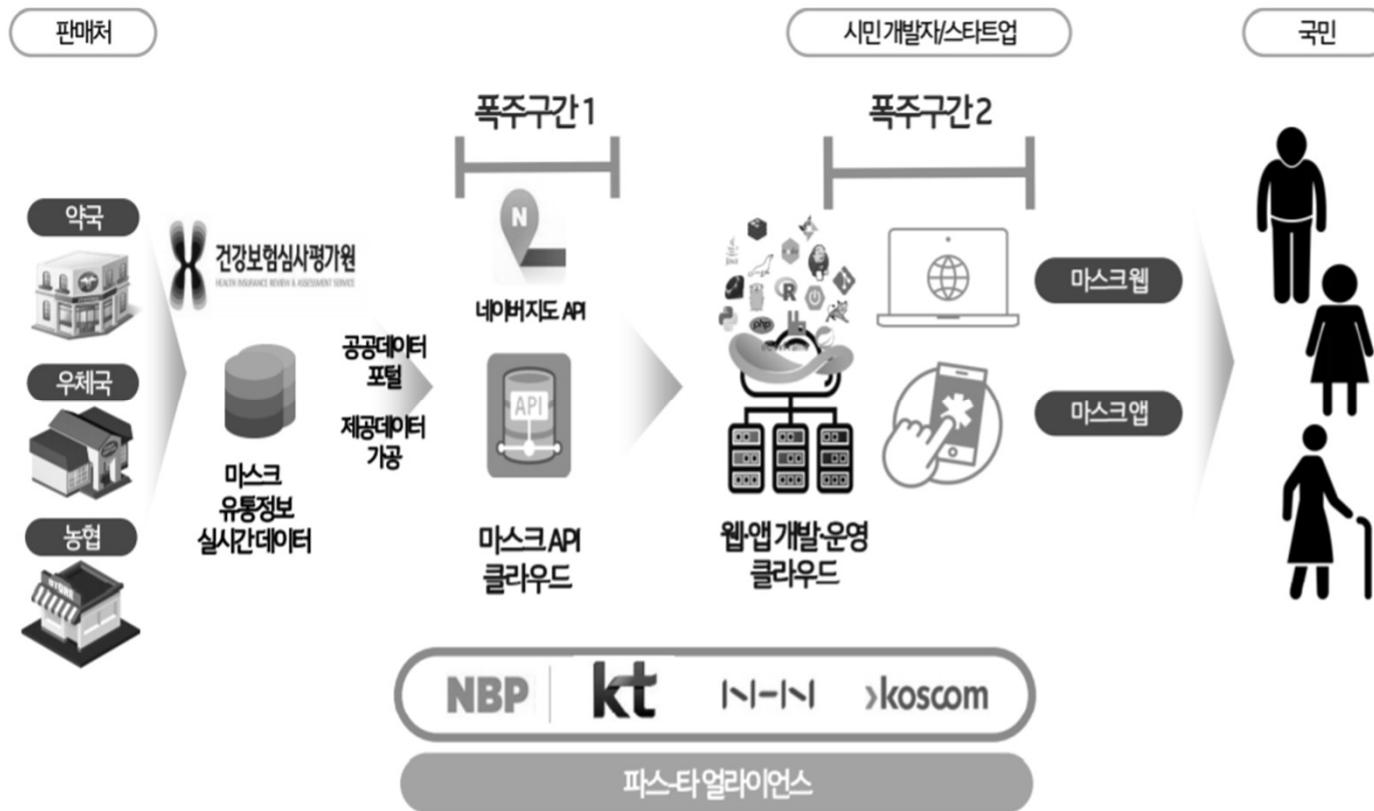
1) 대표사례



세종사이버대학교

📍 마스크 웹/앱

▶ 파스-타 기반으로 구축 : 한국정보화진흥원 주관



1) 대표사례



세종사이버대학교

마스크 웹/앱

▶ 파스-타 기반으로 구축 : 한국정보화진흥원 주관(NIA, KT 제공)

신속한 서비스 개발 및 런칭 가능(부하 문제 해결)

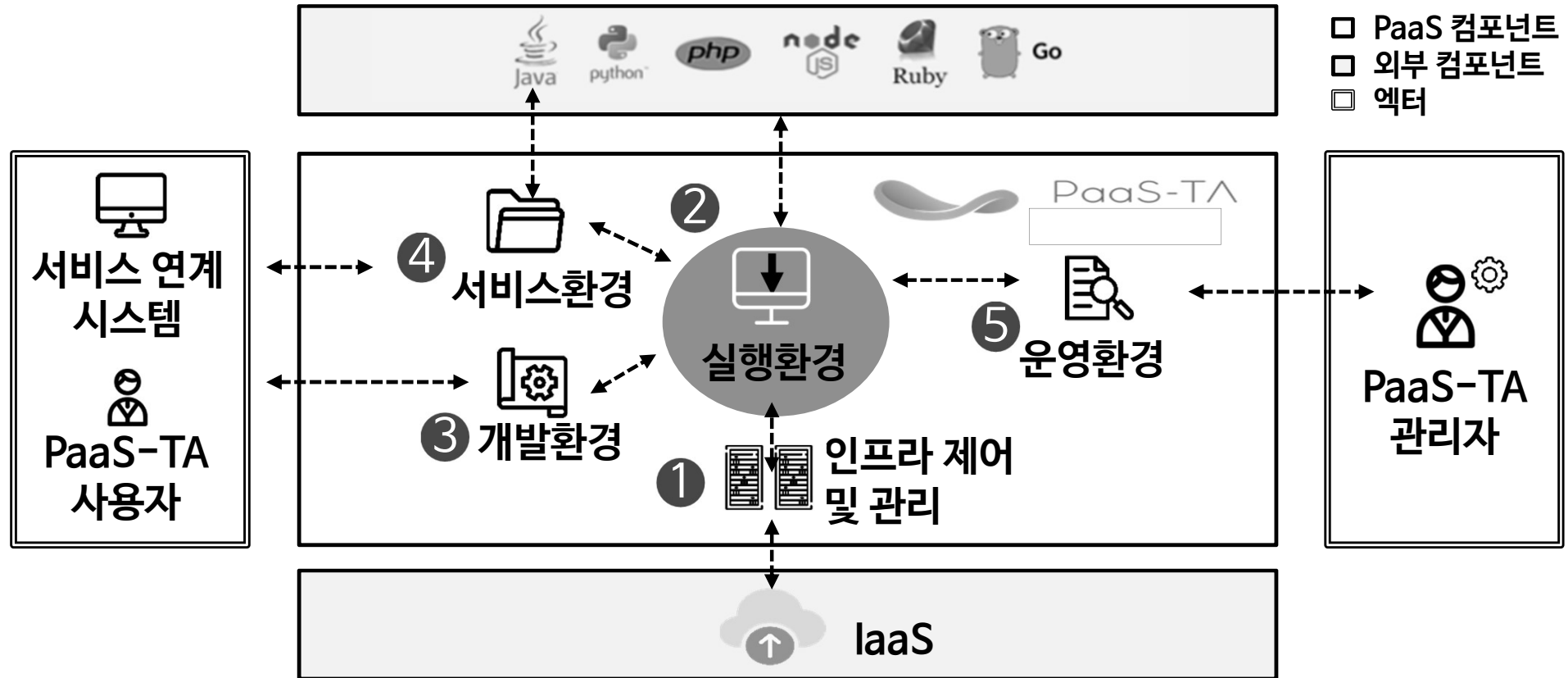
코로나19 극복을 위한 파스-타 무상 제공			
파스-타 얼라이언스			
코로나19 정보 제공을 위해 클라우드가 필요한 공익성 웹/앱 서비스 개발 지원(20.03) 공적 마스크 서비스를 클라우드 제공 후 4일만 서비스 런칭			
마스크 API 공개 지원			
3.6(금)	3.7(토)	3.8(일)	3.9(월)
마스크 API 개발	마스크API 클라우드 테스트	마스크API 클라우드 베타 운영	마스크API 클라우드 베타운영 공개
앱·웹 개발 · 운영 환경 제공			
3.6(금)	3.7(토)	3.8(일)	
무상지원 공지 및 자원확보	개발자·기업에 클라우드 제공 및 개발 시작	개발완료 웹·앱 대상 자동확장되는 운영 클라우드 이관	

클라우드를 통해 전국민의 요청을 안정적으로 소화(3월 11일 기준)

시간	공적 마스크 서비스 이용 현황		
	호출수	평균응답시간(ms)	초당 호출수
07:00~08:00	2,865,695	418	796
08:00~09:00	7,109,320	5,759	1,975
09:00~10:00	8,327,624	14,498	2,313
10:00~11:00	5,492,678	7,759	1,526
11:00~12:00	7,079,548	2,028	1,967
12:00~13:00	7,633,692	843	2,120
13:00~14:00	6,862,595	951	1,906
14:00~15:00	4,877,089	2,041	1,355
15:00~16:00	4,448,163	825	1,236
16:00~17:00	5,268,025	838	1,463
17:00~18:00	6,116,495	845	1,699
18:00~19:00	4,949,179	696	1,375
합계	71,030,103		



2) 파스-타 플랫폼 도구



도구를 통해 사용량 기반 미터링, 상품 카탈로그, 구매, 관리 등 가능



세종사이버대학교

