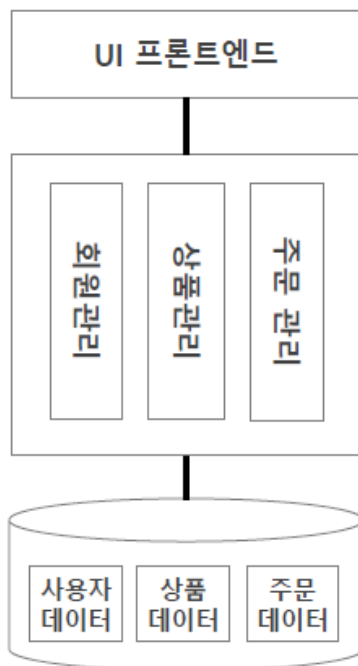
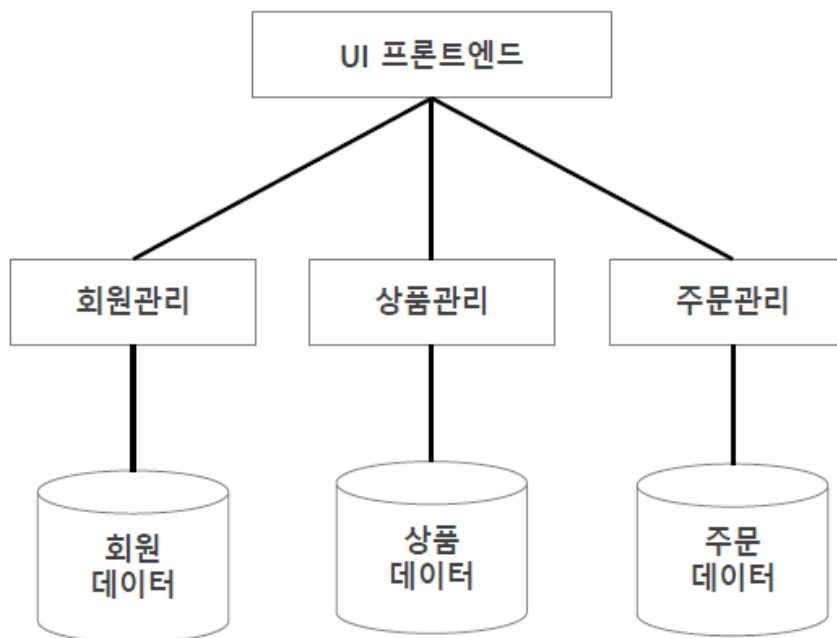


- 업무상의 기능 또는 역할을 하나의 기능 묶음으로 개발된 컴포넌트 → 한 가지의 역할만 수행
- REST API 등을 통하여 서비스들의 기능을 제공하고 사용
- 데이터를 공유하지 않고 서비스별로 독립적으로 가공하고 저장함

Monolithic Architecture

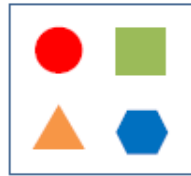


Microservice Architecture



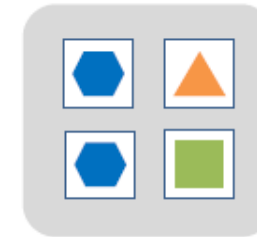
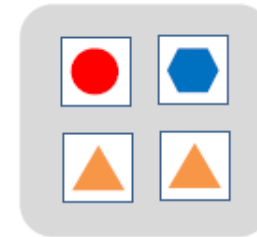
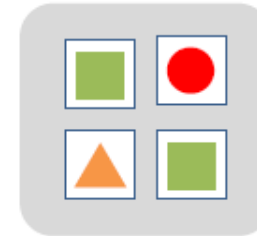
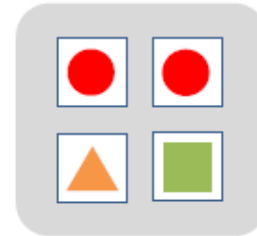
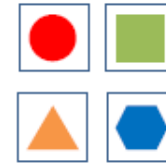
표준 API/Protocol을 통한 데이터 통신

Monolithic Architecture



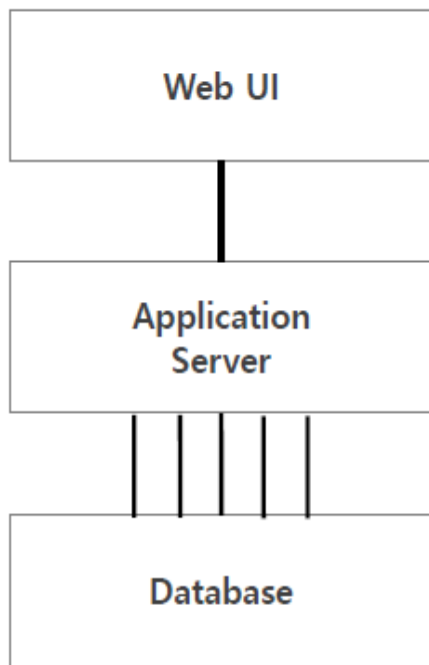
모든 기능의 전체 프로세스 복제

Microservice Architecture



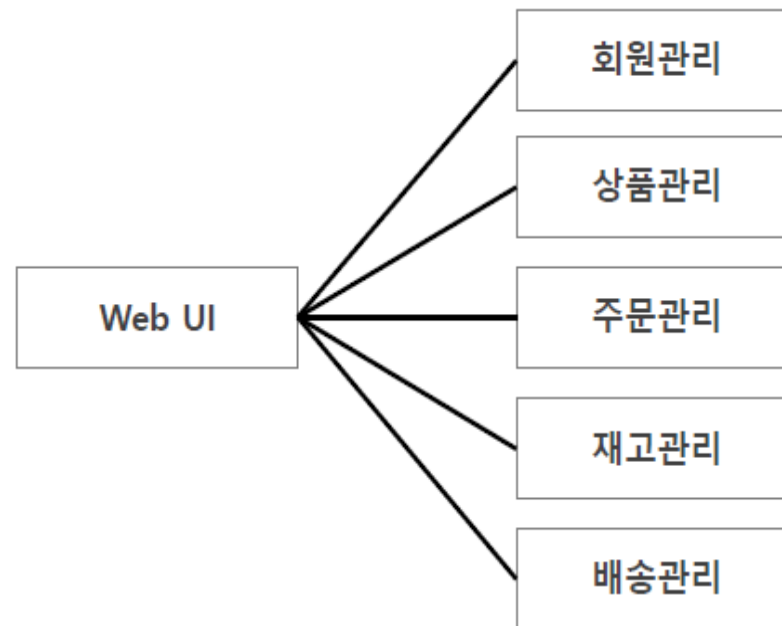
필요한 프로세스만 복제

Monolithic Architecture



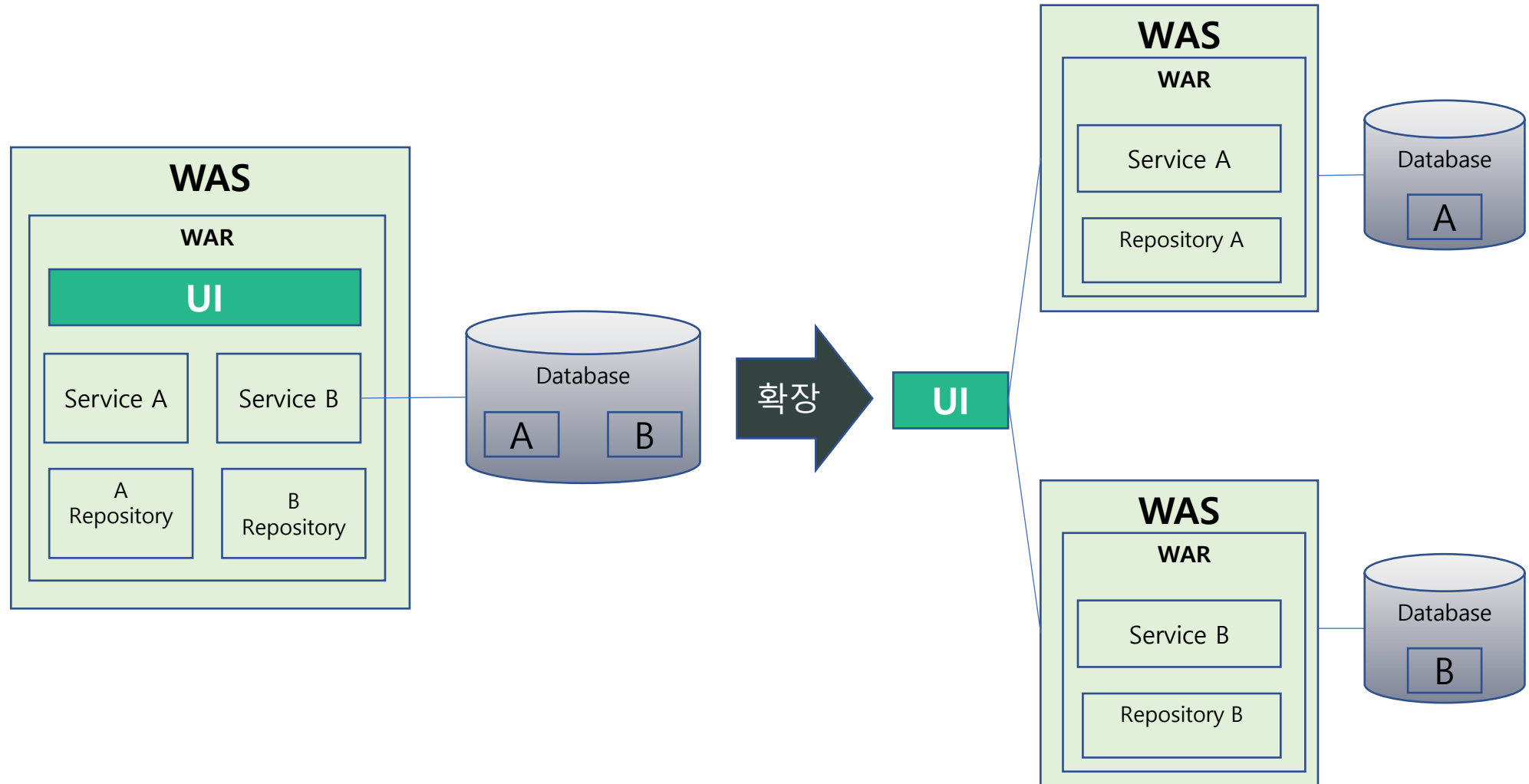
- 단순한 아키텍처
- 특정 서비스 변경 시 모든 코드의 수정 필요

Microservice Architecture

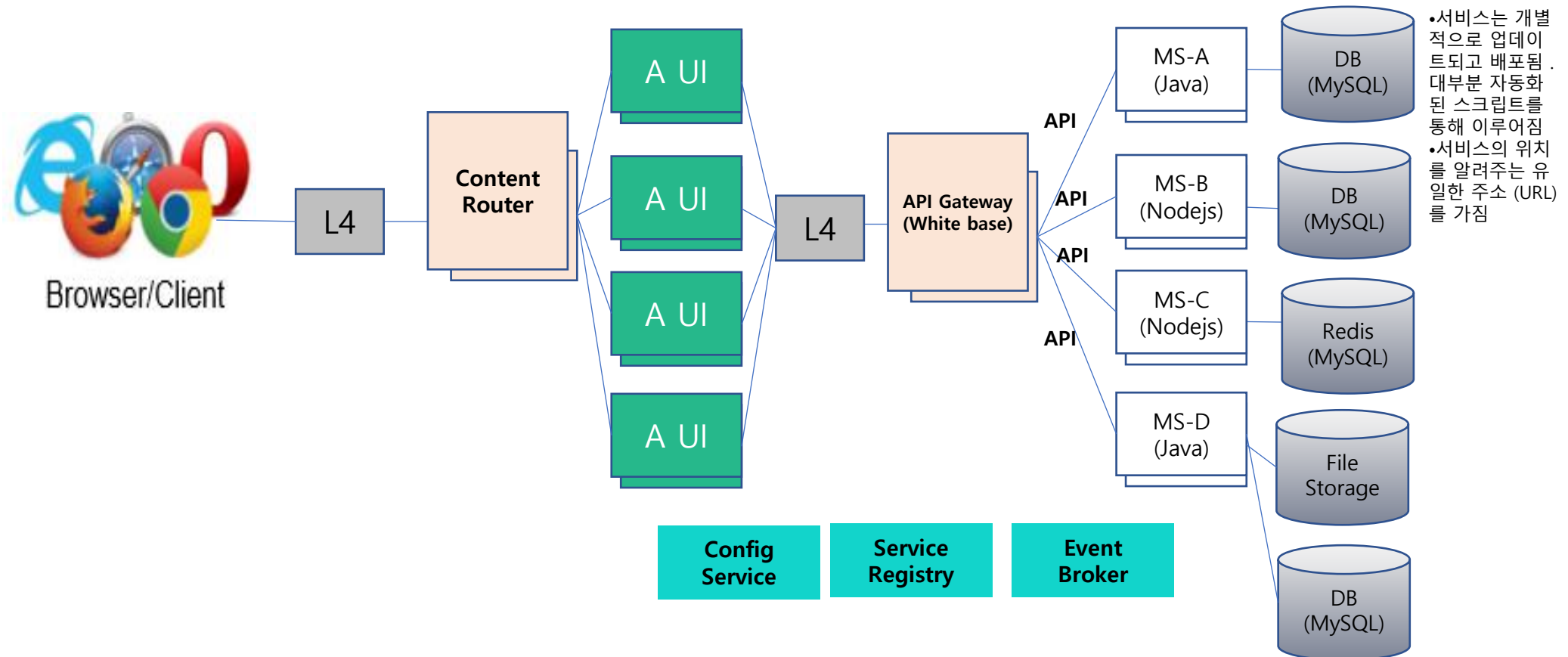


- Web UI 클라이언트가 모든 서비스의 호스트 정보를 필요
- 서비스 개수가 늘어나면 응답 속도가 늦어짐
- 요청 처리를 위한 클라이언트의 코드가 매우 복잡해짐
- 모든 서비스의 프로토콜이 서로 일치하지 않을 수 있음

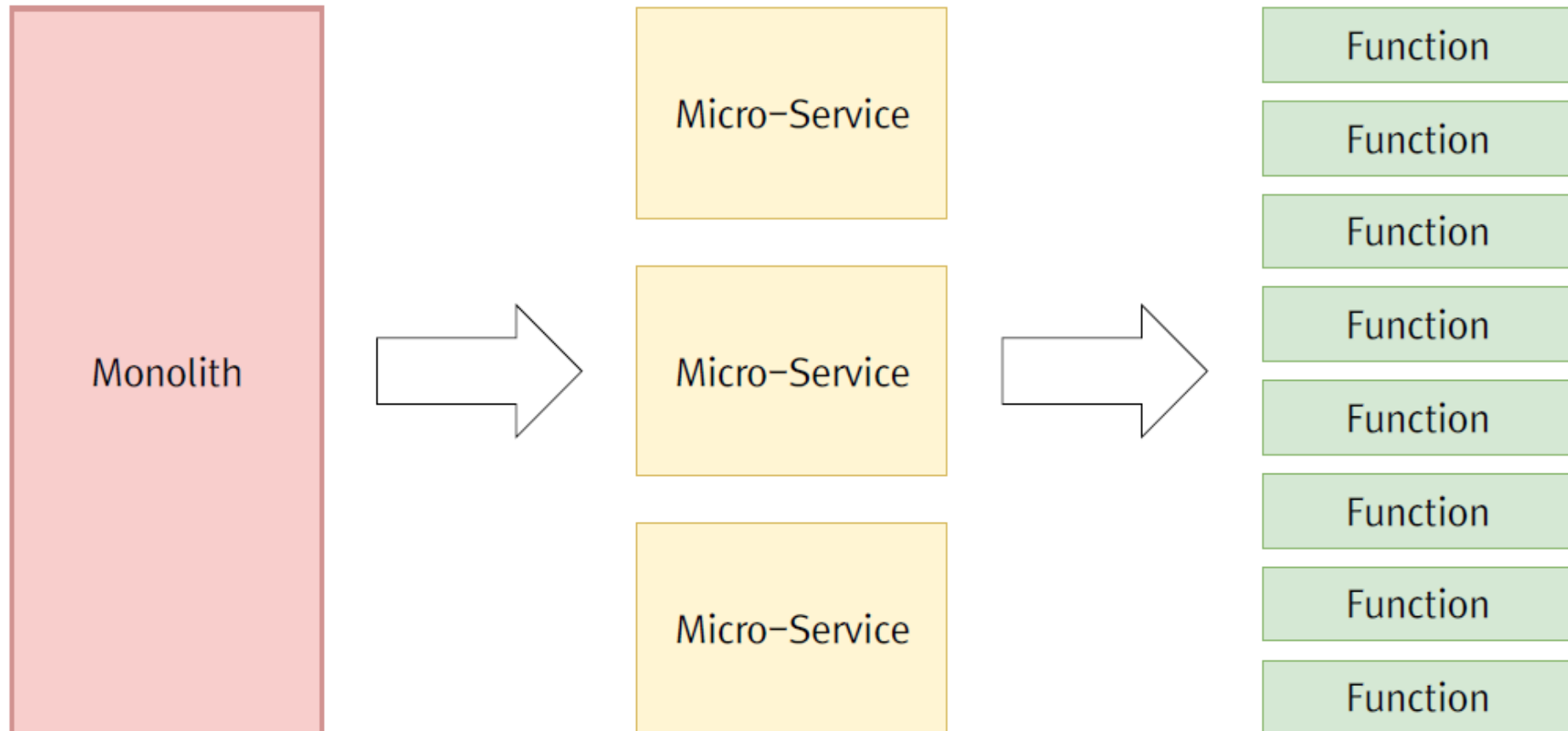
서비스를 분할하고 서비스별로 별도의 데이터베이스 구축



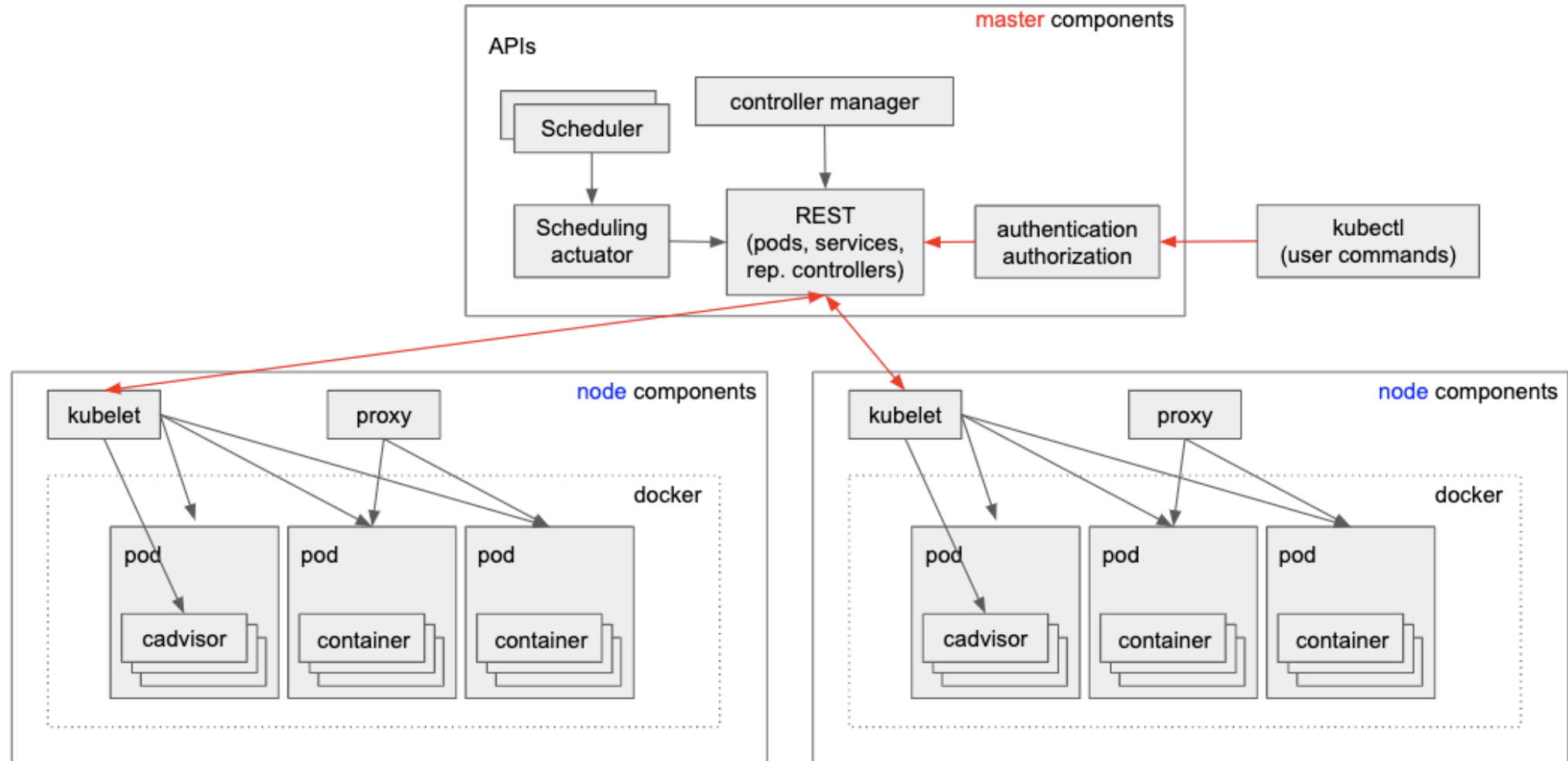
- 독립된 서비스들은 표준화된**API**를 통해서 서로 통신/결합→ 다른 관련 서비스를 바꾸지 않고도 원하는 특정 서비스만 변경
- 구축 시 중점 사항 : 느슨하게 결합된 구성요소들, 확장성, 코드의 분리(partitioning), 업그레이드와 변경을 쉽고 빠르게 하고 유연성을 보장하는 상태



- 마이크로서비스는 기능(Function) 단위로 분화되고, 향후 서버리스 컴퓨팅 시대가 도래할 것으로 예상됨

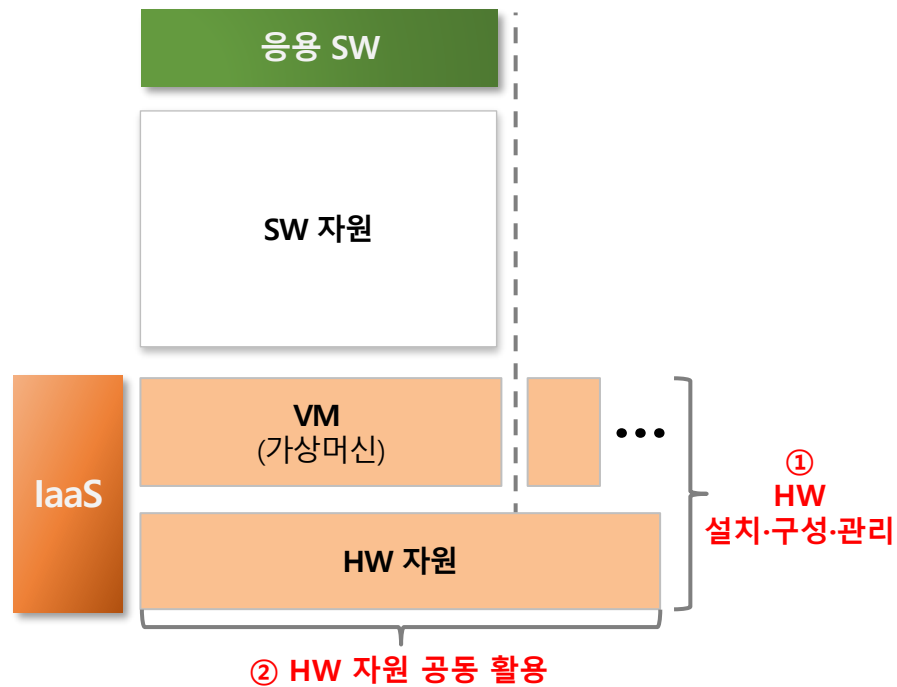


- 마이크로서비스는 컨테이너에 독립적으로 탑재되어 실행됨. 마이크로서비스의 개수가 많아지면 부하분산과 컨테이너 오케스트레이션 메커니즘 필요 -> 최근 쿠버네티스가 각광 받음



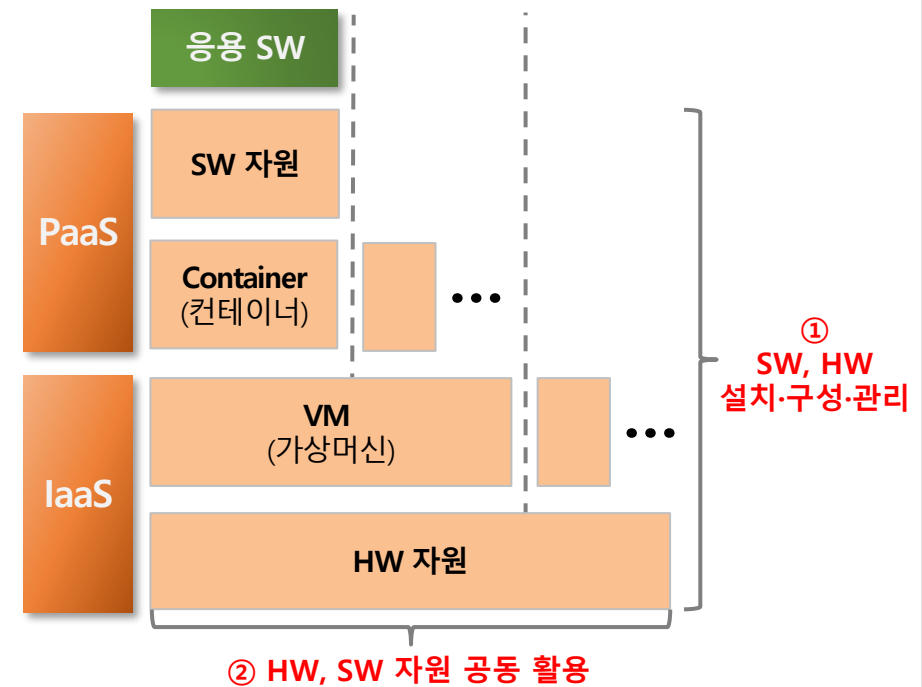
클라우드 플랫폼(PaaS) 기술을 통한 [HW 및 SW 자원의 효율적인 관리]

IaaS 환경 응용 SW 개발



- HW 자원의 공동 활용 ▶ IT 비용 절감
- HW 설치·구성 자동화 ▶ 개발 생산성 향상

PaaS 환경 응용 SW 개발



- HW, SW 자원의 공동 활용 ▶ IT 비용 절감 극대화
- HW, SW 설치·구성 자동화 ▶ 개발 생산성 극대화

파스-타 주요 개발 내용



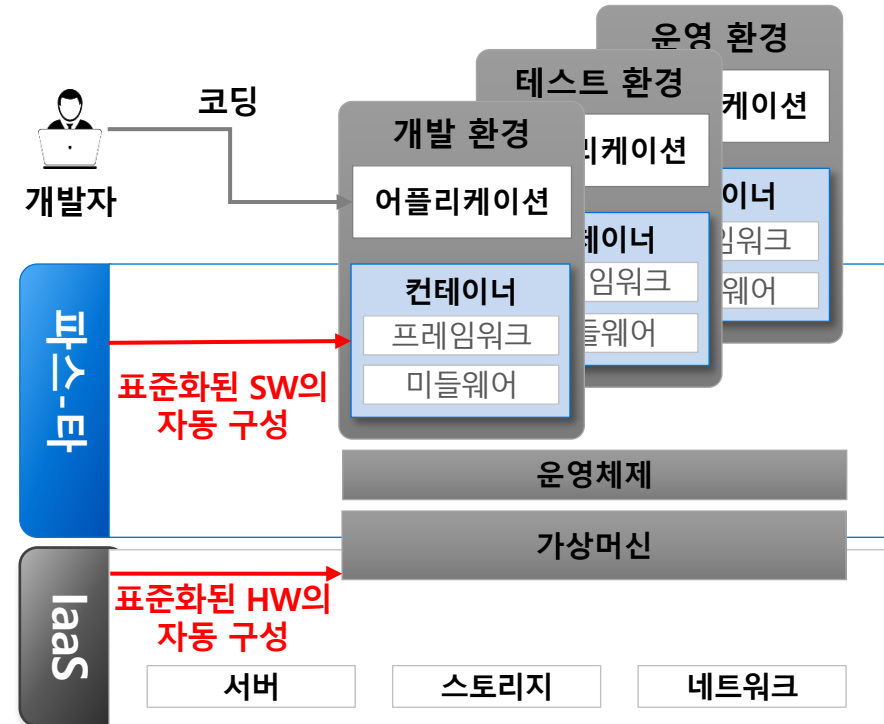
표준화된 HW 및 SW 개발·실행 환경을 수분 내로 자동화함으로써 응용 SW의 신속한 개발·테스트

IaaS 도입 시



- IaaS는 서버·스토리지·네트워크 등 표준화된 HW의 설치 및 구성을 자동화함

파스-타 도입 시

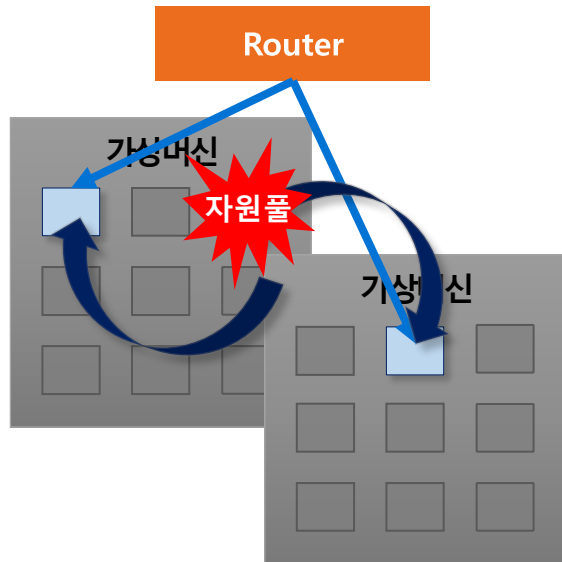


- 파스-타는 어플리케이션 개발에 필요한 프레임워크·미들웨어·OS 등 표준화된 SW의 설치 및 구성을 자동화함

리소스 자원 감시를 통한 자동 감지·확장 기능



어플리케이션 자원 감시·확장



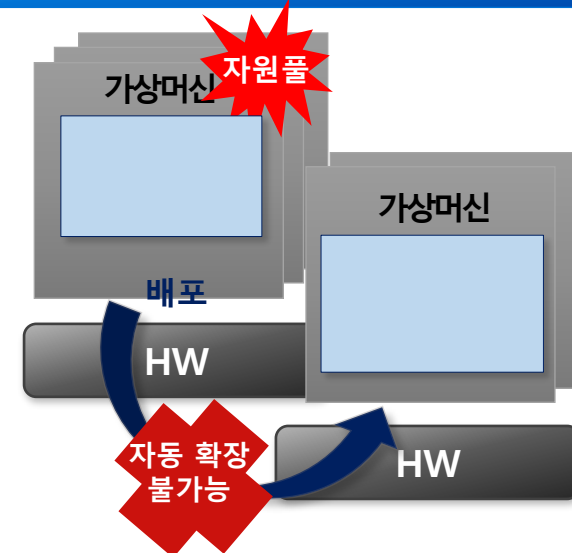
- 어플리케이션 수준의 자원 풀
- 컨테이너 모니터링을 통한 자동 확장

가상 머신 자원 감시·확장



- VM 수준의 자원 풀
- VM 모니터링을 통한 자동 확장

Only IaaS 경우



- VM 수준의 자원 풀
- 새로운 VM 생성, 어플리케이션 배포 및 Load Balance를 위한 3rd Party 제품 필요

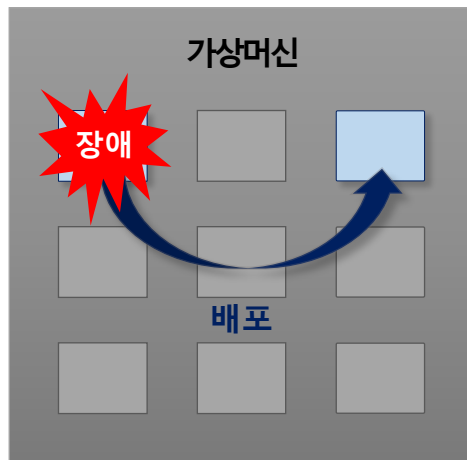
주요 특징점

- CPU, Memory, Disk 모니터링을 통한
 - ▶ 어플리케이션 자동 자원 확장
 - ▶ VM 자동 자원 확장

장애유형의 자동 감지·복구를 통한 가용성 강화

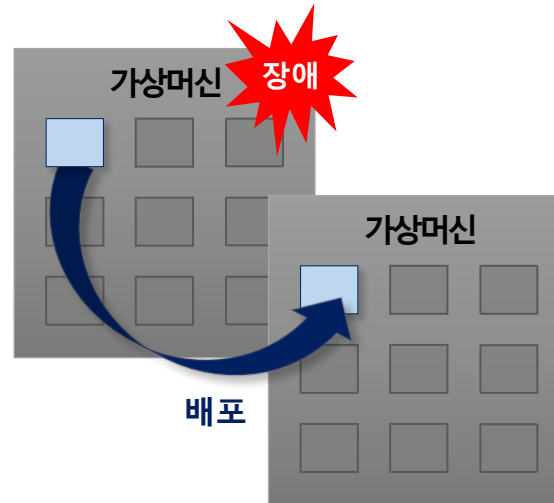


어플리케이션 장애 복구



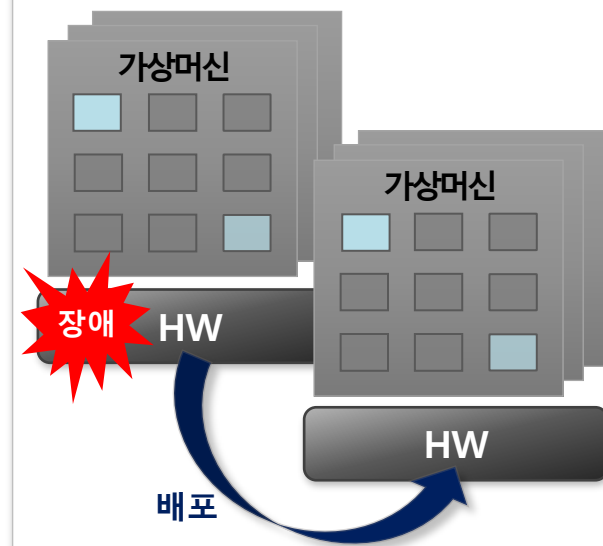
- 어플리케이션 수준의 장애 발생 시 동일한 가상머신의 컨테이너로 이동

가상머신 장애 복구



- 가상머신 수준의 장애 발생 시 다른 가상머신의 컨테이너로 이동

HW 장애 복구



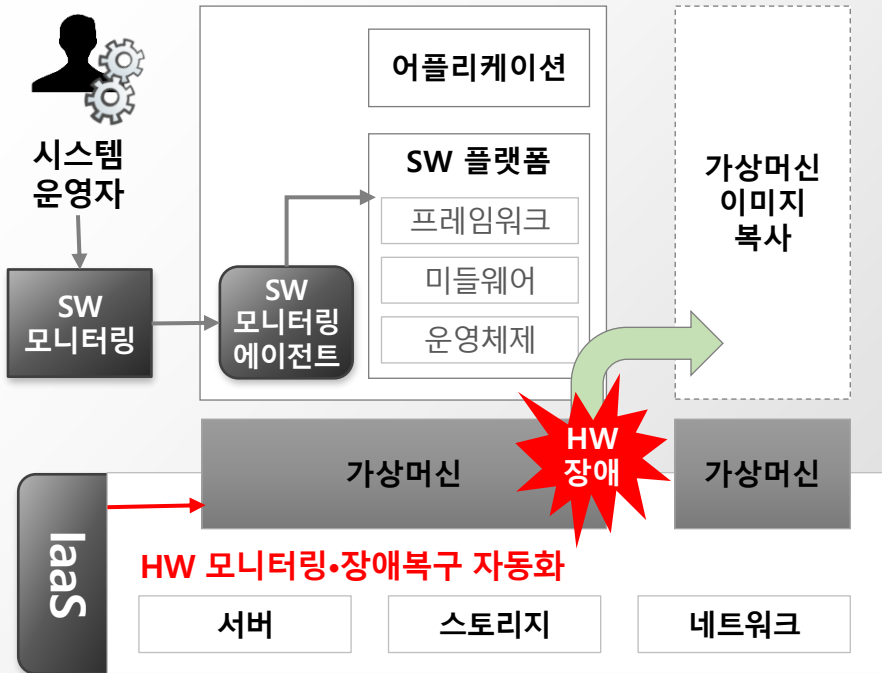
- HW 수준의 장애 발생 시 다른 HW 가상머신의 컨테이너로 이동

주요 특징점

- 다양한 유형의 장애를 자동으로 감지하고 복구하는 기능 제공
 - ▶ SaaS의 Down-time 최소화

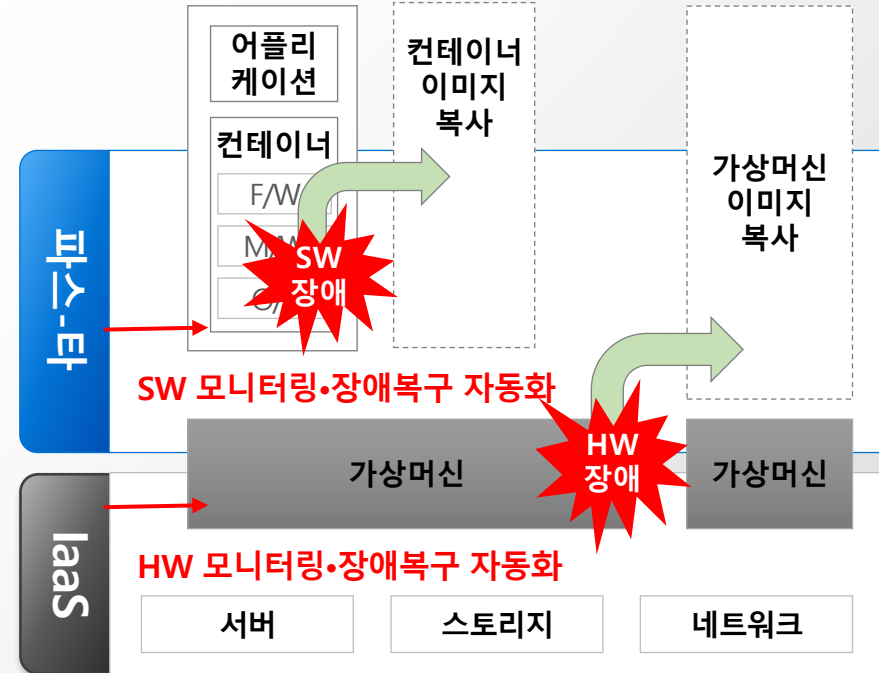


laaS 도입 시



- IaaS는 가상머신에 대한 모니터링 및 장애대응 자동화를 통해 H/W 수준의 고가용성을 보장함

파스-타 도입 시



- 파스-타는 가상머신 위에서 동작하는 컨테이너에 대한 모니터링 및 장애대응 자동화를 통해 **SW 수준의 고가용성을 보장함**

파스타 구축 사례: 지방행정 클라우드 플랫폼 구축



서울시 차세대 업무관리시스템

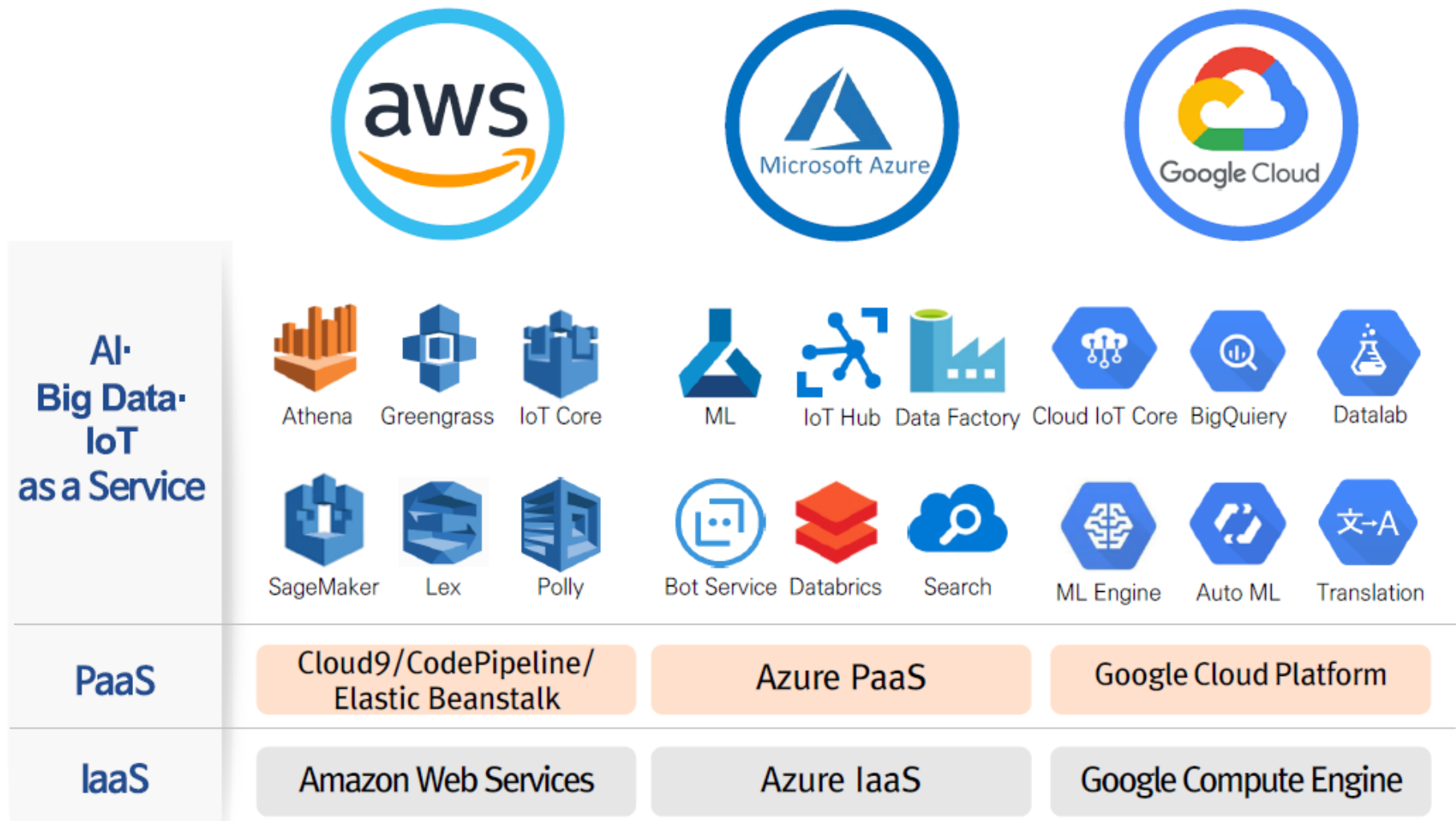


전자결재, 과제관리 등 행정업무시스템
서울시 및 8개 자치구 공무원 2만여명 사용
(월 평균 68건 이상의 결재)

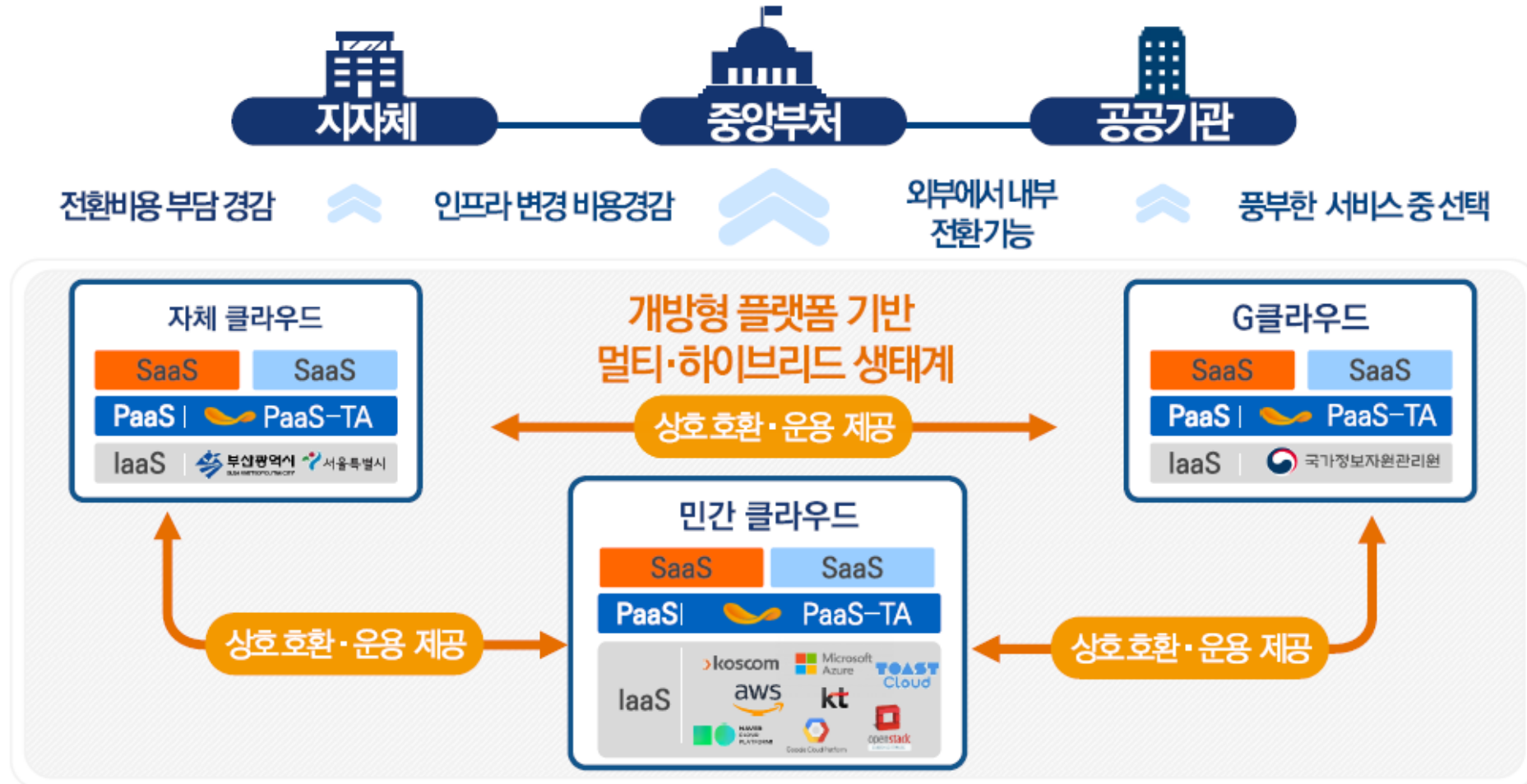
부산시 클라우드 플랫폼 구축



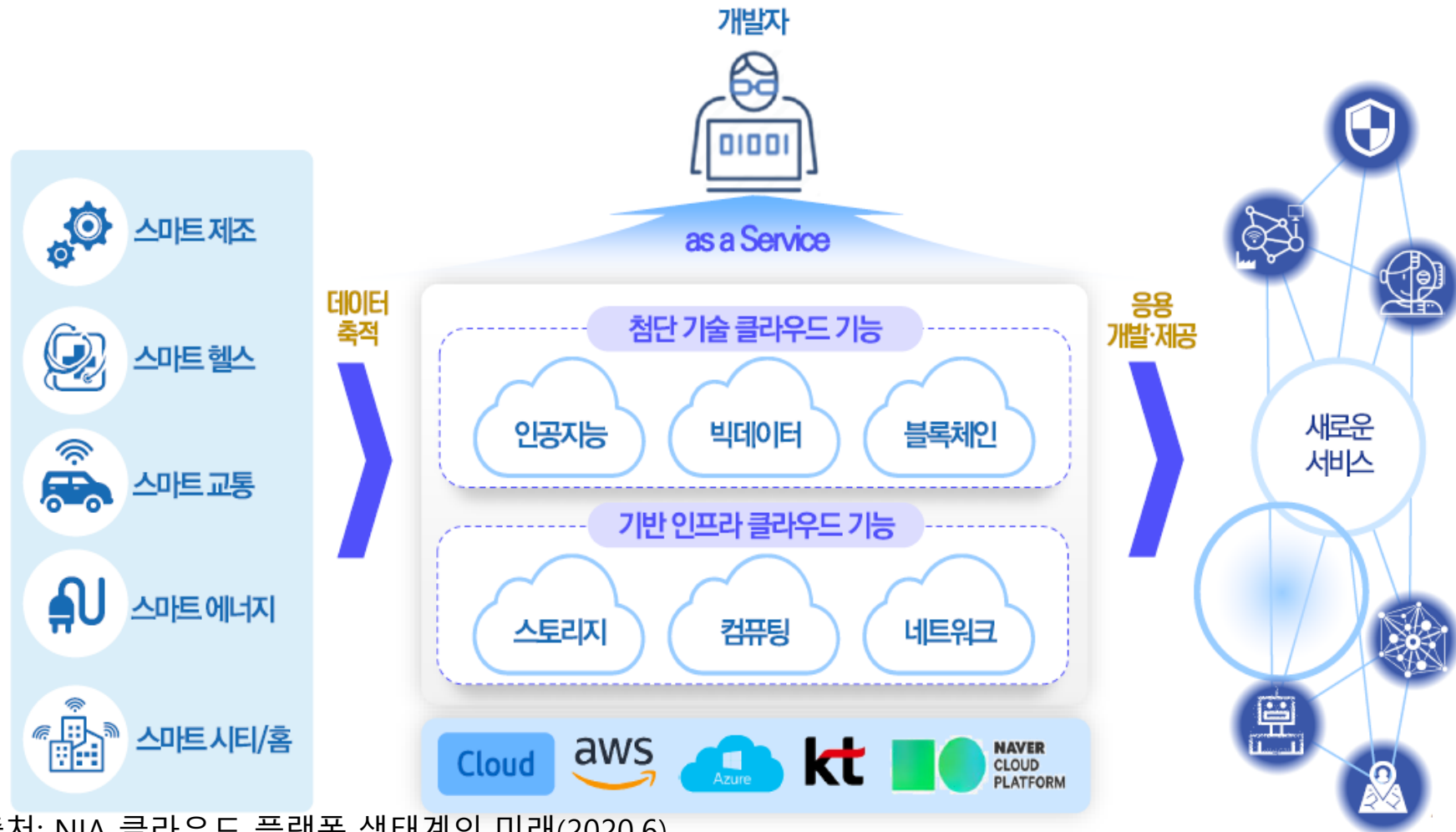
파스타 기반 클라우드 플랫폼 구축으로
행정망 및 인터넷망의 203개 시스템 전환



■ 국가 상호 호환·운영 생태계 조성

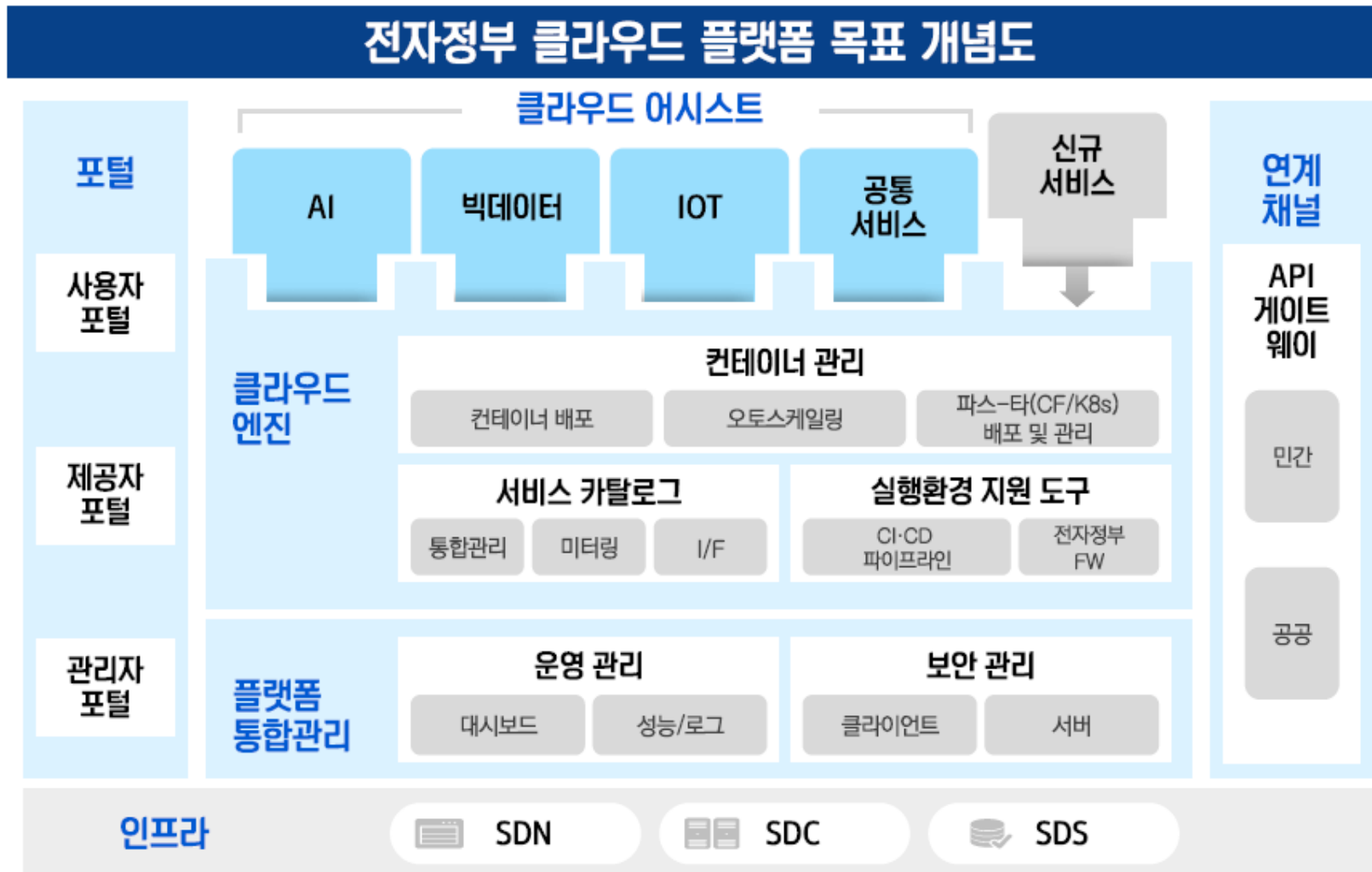


■ 기술 융합을 지원하는 플랫폼으로 진화



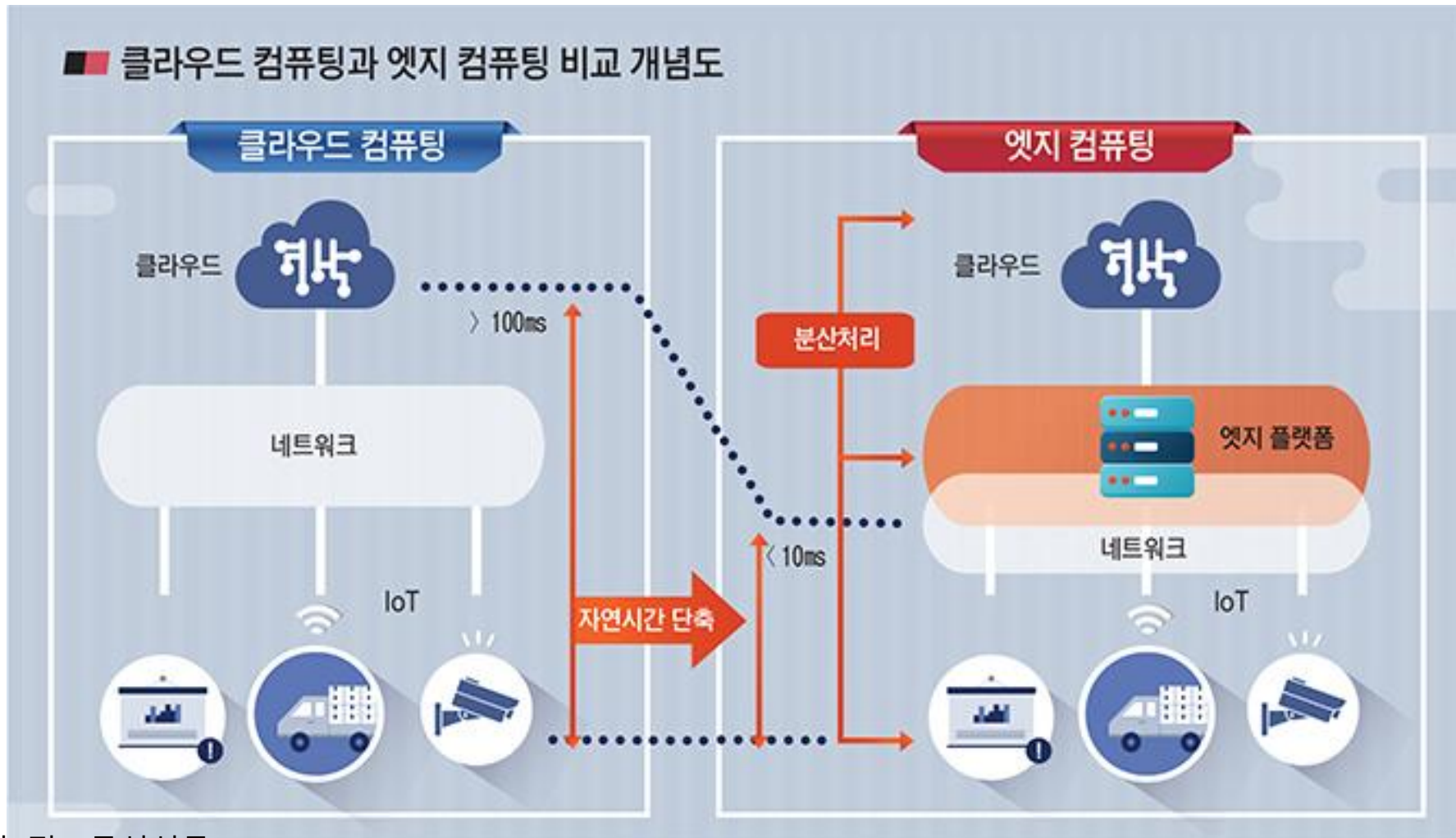
출처: NIA 클라우드 플랫폼 생태계의 미래(2020.6)

- 전자정부 클라우드 플랫폼(행안부: '19-'21)

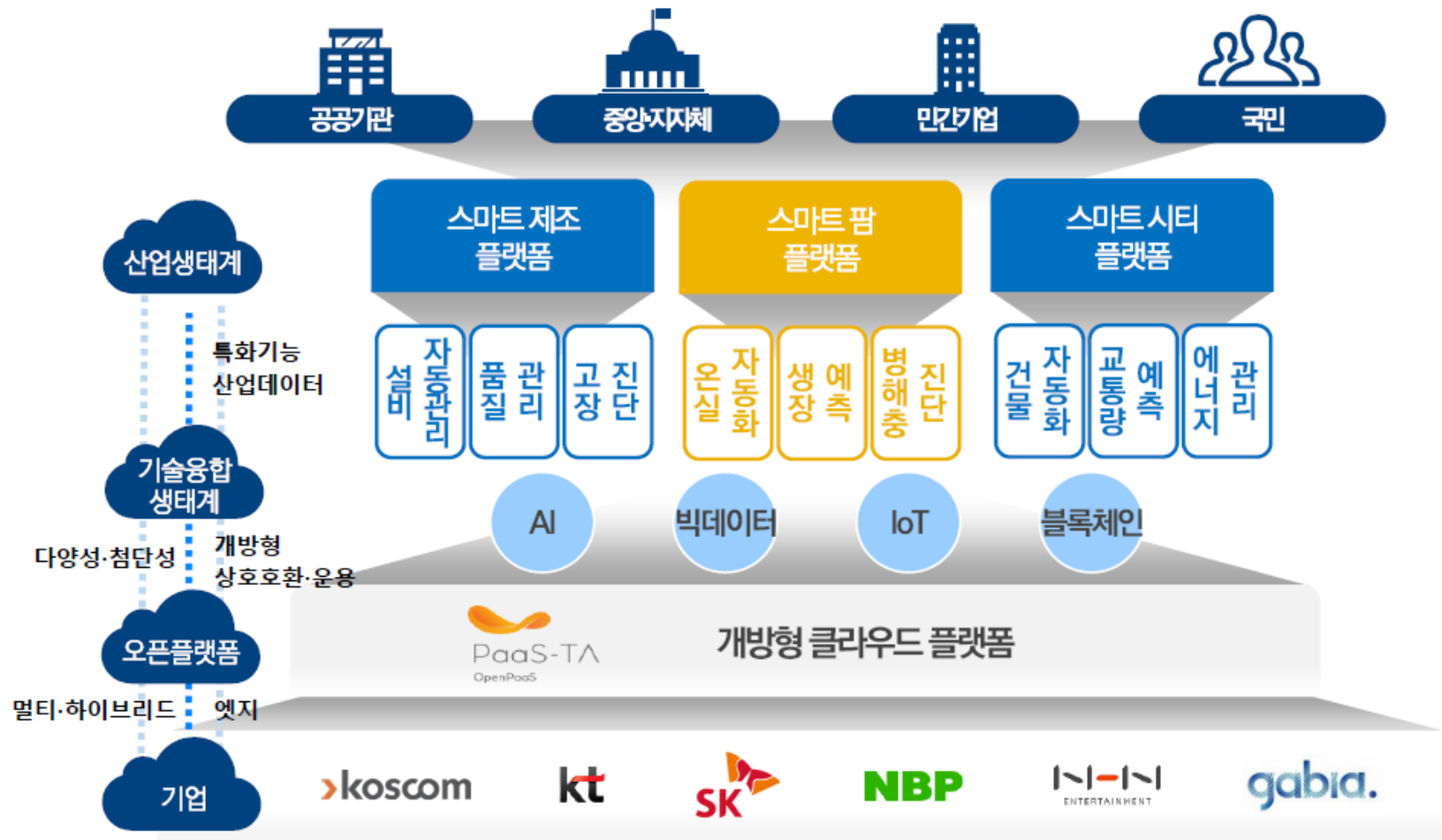


출처: NIA 클라우드 플랫폼 생태계의 미래(2020.6)

- IoT, 자율 주행, 5G 등 기술 개발 영향으로 데이터 폭증 대비: 엣지클라우드



■ 기술융합, 멀티·하이브리드(엣지), 클라우드 네이티브



감사합니다

