


1차년도 주요 결과물

(과제명) 대규모 분산 에너지 저장장치 인프라의 안전한 자율운영
및 성능 평가를 위한 지능형 SW 프레임워크 개발
(과제번호) 2021-0-00077

- 결과물명 : 웹 기반 에너지 저장 장치 데이터 공개SW 기술
설계서
- 작성일자 : 2021년 12월 1일

과학기술정보통신부 SW컴퓨팅산업원천기술개발사업
“1차년도 주요 결과물”로 제출합니다.

수행기관	성명/직위	확인
한국전자기술연구원	최효섭/책임연구원	

정보통신기획평가원장 귀하

사 용 권 한

본 문서에 대한 서명은 한국전자기술연구원 내부에서 본 문서에 대하여
수행 및 유지관리의 책임이 있음을 인정하는 것임.

본 문서는 작성, 검토, 승인하여 승인된 원본을 보관한다.

작성자 : 임홍휘

일자 : 2021. 09. 30

검토자 : 김창우


일자 : 2021. 11. 01

승인자 : 최효섭

일자 : 2021. 12. 01


제 · 개정 이력

버전	변경일자	제·개정 내용	작성자
1.0	2021-09-30	최초 등록	임홍휘


	웹 기반 에너지 저장 장치 데이터 공개SW 기술 설계서	
	과제명	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW프레임워크 개발

목 차

1. 개요	1
1.1 목적	
1.2 범위	
1.3 시스템 개요	
1.4 관련 계획 및 표준	
2. 요구사항 추적 매트릭스.....	3
3. SW 아키텍처 설계.....	4
3.1 SW 아키텍처 개요	
3.2 SW 아키텍처 구성도	
3.3 SW 아키텍처 구성 요소	
4. 시스템 아키텍처 설계.....	6
4.1 하드웨어 구성도	
4.2 네트워크 구성도	
5. 주요 요구사항별 기술 설계.....	7
6. 용어.....	11
7. Reference.....	12

	웹 기반 에너지 저장 장치 데이터 공개SW 기술 설계서	
	과제명	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW프레임워크 개발

			위한 데이터를 수집하여 전주기별 에너지 데이터셋, 에너지 분석 및 학습 인터페이스를 일반 사용자에게 공개하는 웹 기반의 공개 SW 서비스 개발 요구사항을 정의한 문서
가이드	-	행정•공공기관 웹사이트 구축•운영 가이드(2021-03, 행정안전부)	행정기관 및 공공기관의 웹사이트 구축(개발)을 추진하는 발주자와 웹사이트를 운영하는 관리자가 지켜야 할 다양한 기준과 관련 사항(규정)들을 정리한 웹사이트 운영관리 활용 안내서
기술	-	RESTAPI	웹 시스템을 위한 소프트웨어 아키텍처의 한 형식으로 HTTP를 활용하는 많은 API가 REST 원리를 따라 제공된다.
기술	-	OAS(OpenAPI Specification)	RESTful 웹 서비스를 설명, 생성, 소비 및 시각화하기 위한 기계 판독 가능 인터페이스 파일의 사양(구글, 마이크로소프트, IBM 등의 많은 기업에서 사용중)

	웹 기반 에너지 저장 장치 데이터 공개SW 기술 설계서	
	과제명	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW프레임워크 개발

2. 요구사항 추적 매트릭스

ESS 공개 SW 서비스 요구사항정의서에 기반한 요구사항 목록을 기록한다.

NO	요구사항 명
F-1	전주기별 에너지 데이터셋 제공
F-2	에너지 분석 및 학습 OpenAPI 제공
F-3	인프라 응용별(ESS, EV) 관리-운영 서비스 배포 기술
F-4	데이터의 유형에 따른 데이터 시각화 제공
F-5	지능형 안전 SW 프레임워크 및 공개 SW 서비스 연동
F-6	사용자 관리
F-7	다중 클라이언트 허용 처리 기능
S-1	데이터셋 민감정보 비식별화
P-1	공개 SW 서비스 요청 처리 속도
P-2	페이지 응답 시간

KETI	웹 기반 에너지 저장 장치 데이터 공개SW 기술 설계서	
	과제명	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW프레임워크 개발

3. SW 아키텍처 설계

3.1 SW 아키텍처 개요

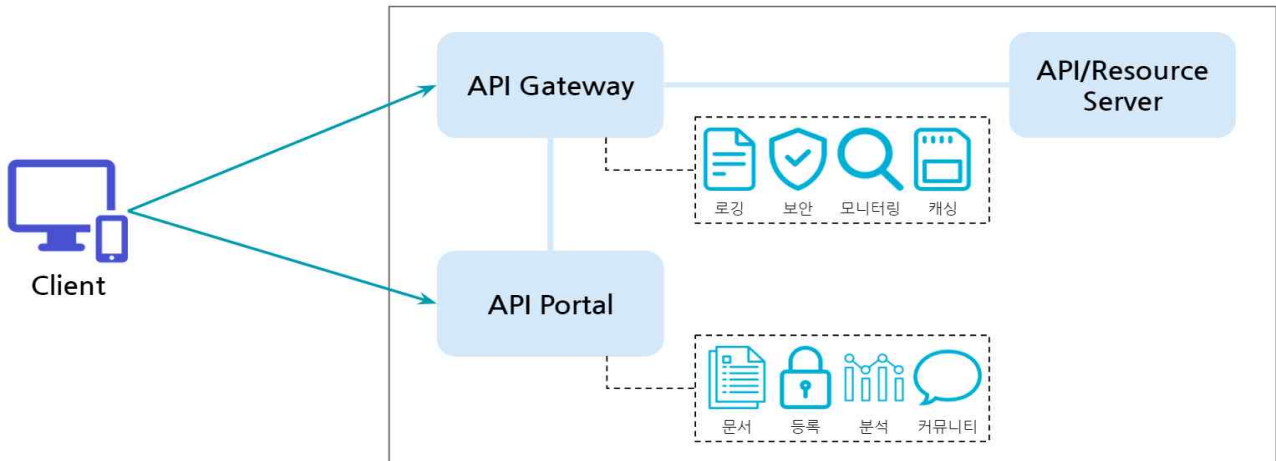


그림 2 공개 SW 서비스 SW 아키텍처

시스템은 크게 API/Resource Server, API Gateway, API Portal 3개의 시스템이 존재한다. API/Resource Server는 API를 직접 생성하거나, 리소스를 제공하는 비즈니스 로직이 포함된 시스템이다. API/Resource Server와 클라이언트가 바로 통신하는 API 생산-소비 모델을 설계할 수 있지만 API 로깅, 인증, 모니터링 및 캐싱 등의 API 관리 기능을 제공하는 API Gateway를 API Server 앞 단에 두는 일종의 Proxy 패턴을 활용해 클라이언트-API Gateway-API Server에 이르는 API 생산-관리-소비 모델을 설계한다. API Portal은 클라이언트가 API를 활용할 수 있게 API Gateway에 인증 설정, API 스펙 정보 확인 및 API 사용 현황을 알 수 있는 정보 등을 게시하는 시스템이다.

3.2 SW 아키텍처 구성도

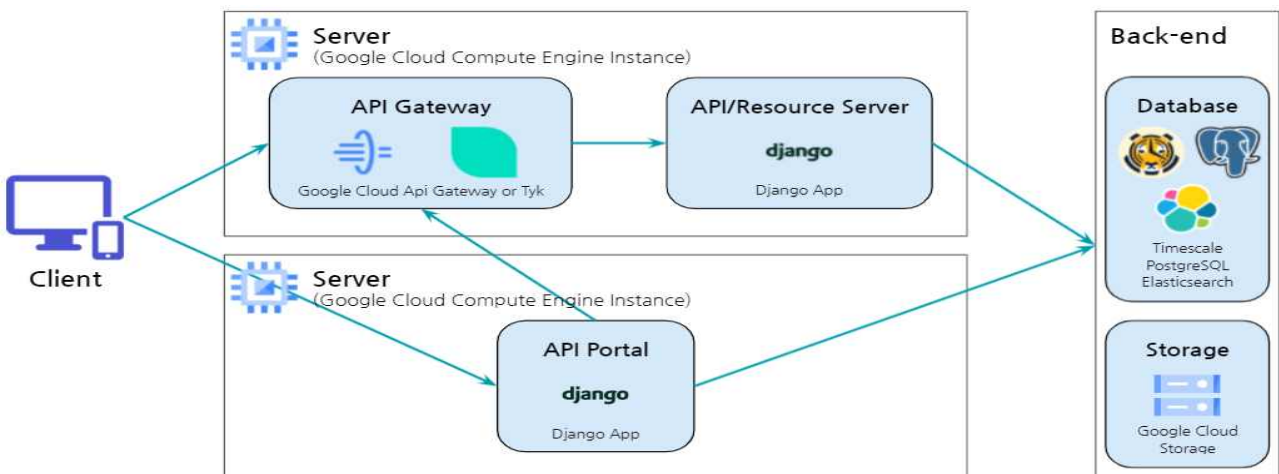



그림 3 공개 SW 서비스 SW 아키텍처 최종 구성도

	웹 기반 에너지 저장 장치 데이터 공개SW 기술 설계서	
	과제명	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW프레임워크 개발

3.3 SW 아키텍처 구성 요소

API/Resource Server로 활용되는 Django App은 ESS 관련 전주기별 데이터셋을 제공한다. Python의 데이터 타입을 직렬화(Serializer)를 통해 JSON, XML, CSV 등과 같은 다양한 형식의 데이터 오픈 포맷 형식으로 변환하는 기능을 지원한다. API Gateway로 활용되는 Google Cloud Api Gateway 또는 Tyk은 API 활용에 대한 로깅, 보안, 모니터링 등을 제공해주는 기능을 포함한다. 인프라 응용별(ESS, EV 등.) 설계된 API를 관리 및 운영하기 위해 필요한 공통적인 기능을 API Gateway 한 곳에서 조정할 수 있고, 효율적인 분산 시스템 아키텍처를 구축할 수 있다. 자주 요청하는 API에 대한 리소스를 캐싱하여 요청 처리 속도를 높일 수도 있다. 이렇게 API Gateway는 Client App과 API Server 중간에 위치하여 API에 대한 요청과 응답을 중개한다. API Portal로 활용되는 Django는 사용자의 API에 대한 접근 인터페이스를 웹 화면에 노출한다. 이것은 API Gateway에서 제공하는 API를 API Portal에서 활용하여 사용자에게 API에 대한 가시성을 제공한다.

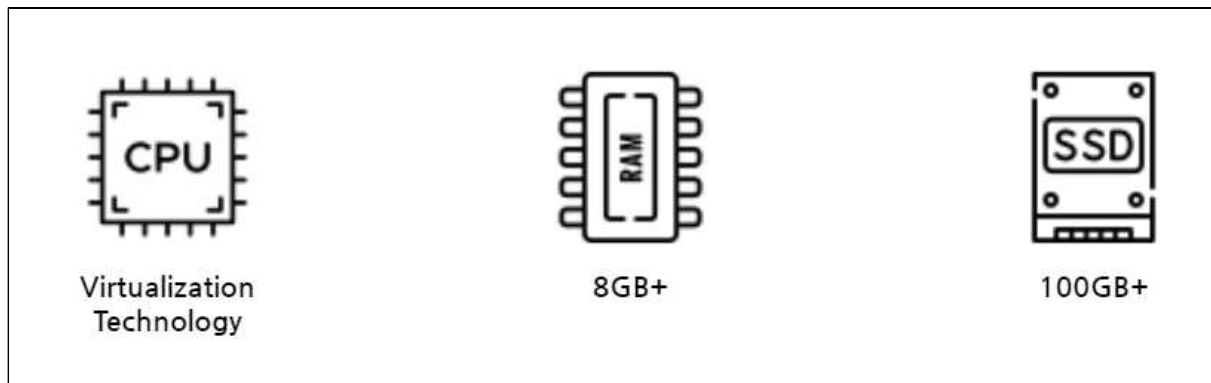
Back-end 환경에서는 Database로 Timescale, PostgreSQL, Elasticsearch 사용한다. ESS 운영 데이터가 시계열 데이터인 것을 고려해 시계열 데이터 처리에 적합한 Timescale을 활용해 API/Resource Server에 데이터를 제공한다. 공개 SW 서비스를 사용하는 사용자는 시스템 관리자와 일반 사용자로 크게 분류할 수 있다. 그래서 사용자 관리를 위한 설계를 관계형 모델로 정해 Full SQL을 지원하는 PostgreSQL 사용해 API Portal과 API/Resource Server에 데이터를 제공한다. ESS 운영 데이터를 모니터링하기 위한 이벤트 로그 데이터를 생성하여 Elasticsearch에서 로그 데이터를 관리하여 API/Resource Server에 데이터를 제공한다. 데이터베이스에서 관리하기 힘든 이미지, 동영상 및 파일 등의 리소스를 Google Cloud Storage에서 관리함으로써, API Portal과 API/Resource Server에서 해당 리소스를 제공 받을 수 있도록 한다.

KETI	웹 기반 에너지 저장 장치 데이터 공개SW 기술 설계서	
	과제명	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW프레임워크 개발

4. 시스템 아키텍처 설계

4.1 하드웨어 구성도

Cloud 기반 솔루션을 사용하므로, 물리적인 하드웨어 구성도는 없다. 시스템이 동작하기 위한 최소 사양은 Docker와 같은 가상화 솔루션이 동작하기 위해 CPU 가상화 기능이 필요한 CPU가 필요하며, 메모리(RAM)는 8GB 이상이어야 한다. Ubuntu 20.04 LTS 버전 이상의 Linux 환경에서 개발과 운영이 이루어질 것이며, 관련 소프트웨어를 저장할 최소 용량은 100GB 이상의 자체 스토리지 용량이 필요하다.



4.2 네트워크 구성도

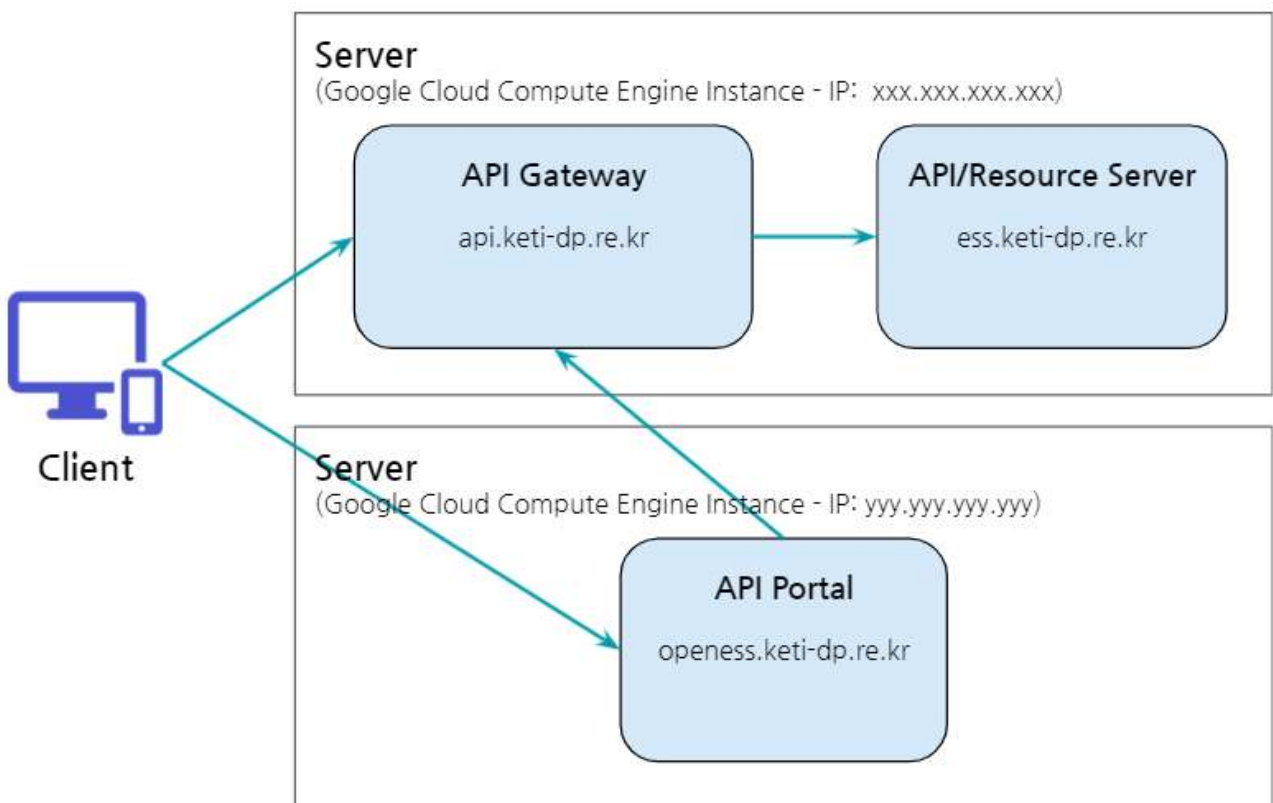


그림 4 공개 SW 서비스 네트워크 구성도

5. 주요 요구사항별 기술 설계

5.1 전주기별 에너지 데이터셋(F-1), 에너지 분석 및 학습 OpenAPI 제공(F-2)

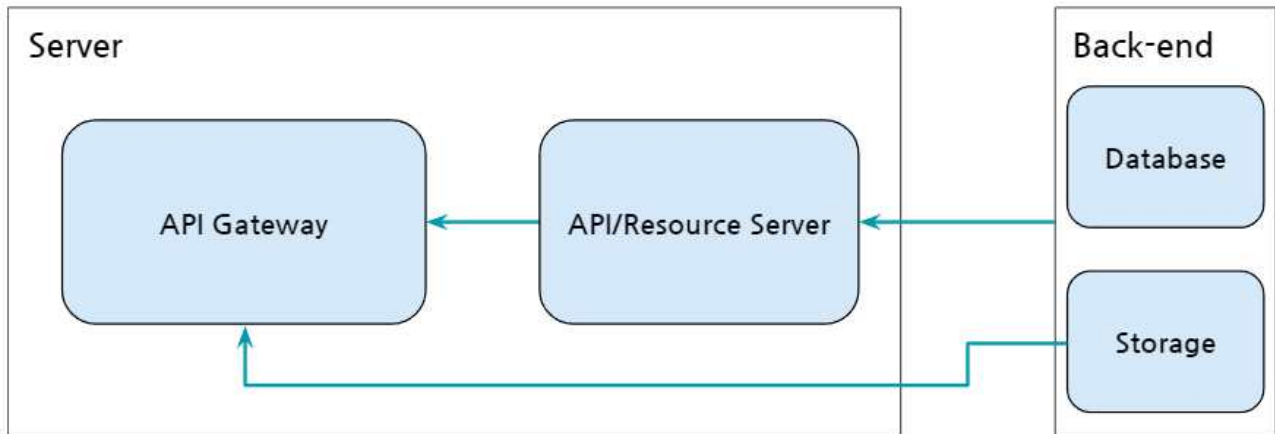


그림 5 공개 SW 서비스 데이터 흐름도

전주기별 에너지 데이터셋, 에너지 분석 및 학습을 통한 결과물들은 모두 Back-end 시스템에 저장된다. 결과물의 유형은 크게 CSV, TXT와 같은 파일, Table 형식의 각각 Row로 저장되어 호출할 수 있는 쿼리, JSON, XML과 같은 데이터 포맷일 수 있다. 이것은 대부분 API/Resource Server에서 API 형태로 최종 가공되어 API Gateway에 전달한다. Storage에 저장되어 API로 호출할 수 있는 최종 데이터일 경우는 API/Resource Server를 거치지 않고, 바로 API Gateway에 전달한다(API Gateway-Storage 통합). API Gateway는 여러 시스템에서 전달된 API를 사용자가 활용하기 쉬운 형태의 API 형태로 가공하는 Endpoint 시스템으로 데이터 및 API를 제공한다.

5.2 인프라 응용별(ESS, EV) 관리-운영 서비스 배포 기술(F-3)

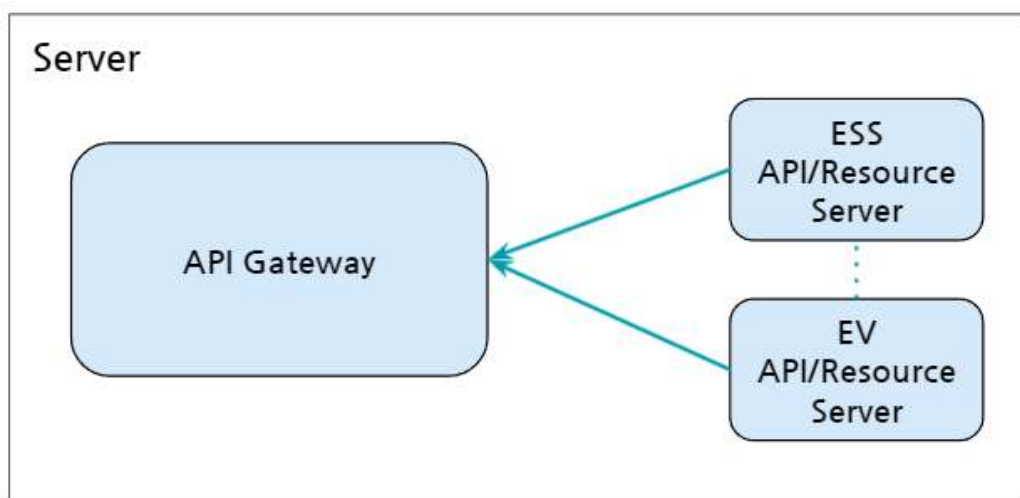


그림 6 공개 SW 서비스 인프라 응용별 데이터 흐름도

KETI	웹 기반 에너지 저장 장치 데이터 공개SW 기술 설계서	
	과제명	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW프레임워크 개발

ESS, EV는 API/Resource Server를 각각 분리하여 시스템을 관리한다. 기본적으로 API Gateway는 여러 API/Resource Server의 API를 통합 관리한다. ESS API/Resource Server는 EV API/Resource Server와 내부적으로 공개된 API를 호출하여 서로 시스템을 연동할 수 있다.

5.3 데이터의 유형에 따른 데이터 시각화 제공(F-4)

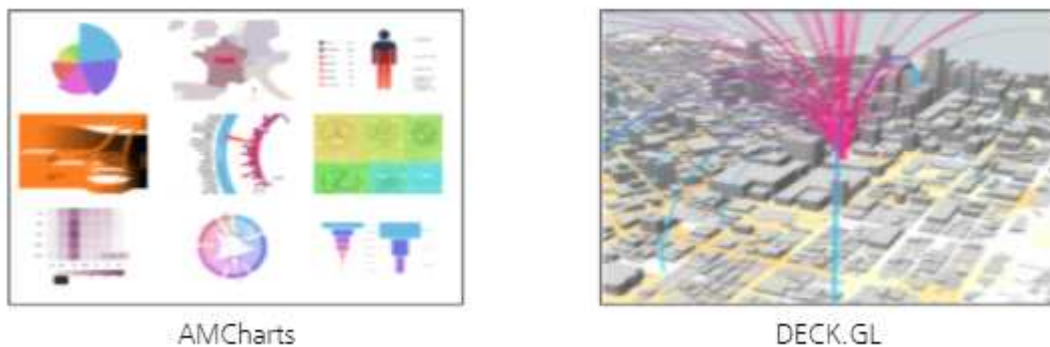


그림 6 Javascript 데이터 시각화 라이브러리(AMCharts와 DECK.GL)

'AMCharts'는 Bar, Pie, Donut, Line, Area, Bubble, Scatter, Candle, Gant, Radar, Map 등의 다양한 유형의 데이터 시각화 차트를 제공한다. 차트 또는 데이터를 JPEG, PNG, PDF, XLSX, CSV 또는 JSON과 같은 형식으로 출력할 수도 있다. 그리고 툴팁, 추세선, 애니메이션 등의 차트 옵션을 활용해 커스텀 차트를 개발할 수 있기 때문에 전문적인 데이터 시각화가 필요한 작업에 적합하다. 'DECK.GL'은 Map 특화 데이터 시각화 라이브러리이다. CSV, JSON, Array, Image, GeoJson, 3D Tile 등의 다양한 형식의 데이터를 로드할 수 있고, 국내 지역 시각화도 가능하다. API/Resource Server와 API Portal에서 API Gateway에서 제공하는 모니터링, 로그 데이터와 구축한 API를 결합하고, Javascript 데이터 시각화 라이브러리를 활용해 데이터를 시각화한다.

5.4 지능형 안전 SW 프레임워크와 연동(F-5)

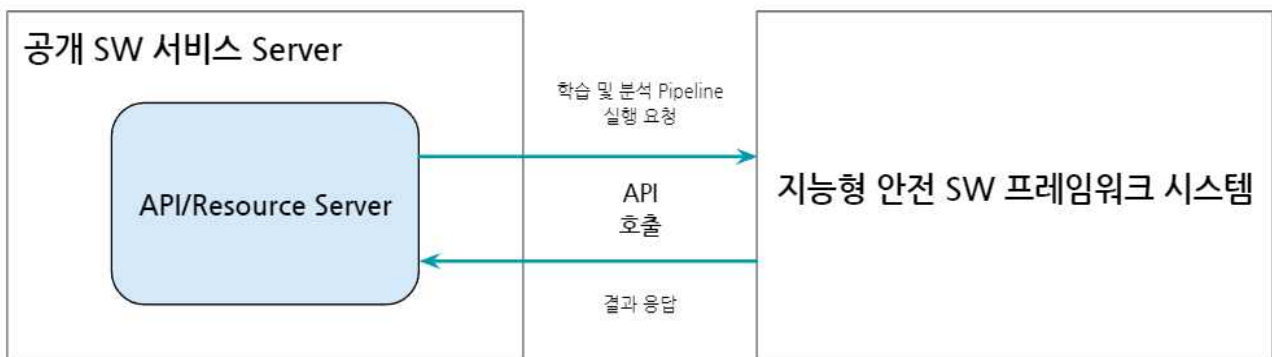


그림 6 공개 SW 서비스와 지능형 안전 SW 프레임워크 연동

KETI	웹 기반 에너지 저장 장치 데이터 공개SW 기술 설계서	
	과제명	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW프레임워크 개발

지능형 안전 SW 프레임워크 시스템과 API 호출을 통해 연동한다. 대상 시스템은 API/Resource Server이다. 지능형 안전 SW 프레임워크 시스템은 에너지 학습 및 분석에 필요한 모듈이 각각 컴포넌트 형태로 존재하고, 필요한 구성에 따라 컴포넌트들을 차례대로 연결하는 Pipeline을 구성한다. API/Resource Server는 학습 및 분석 Pipeline 실행 요청을 통해 지능형 안전 SW 프레임워크 시스템에서 데이터셋 또는 분석 결과 데이터와 같은 응답을 받은 후 API로 가공한다.

5.5 사용자 관리(F-6)

사용자는 크게 관리자와 일반 사용자로 구분한다. 관리자는 API/Resource Server, API Gateway 및 API Portal 시스템을 통틀어 관리한다. 일반 사용자는 API Portal을 활용해 API Gateway에 필요한 API를 요청한다. 이 때 API Portal에선 일반 사용자에게 API를 요청할 수 있는 Key를 발급한다. 일반 사용자는 요청한 API에 관한 모니터링 정보를 자신의 대시보드에서 확인한다. API Gateway는 일반 사용자의 API 요청 할당량을 정하여, 시스템 트래픽을 관리한다.

5.6 데이터셋 민감정보 비식별화(S-1)

민감정보 공개여부	비식별화 방법	예시
필요없음	API 구축 시 민감정보 제외	-
필요	일련의 고유 번호를 사용하여 Key-Value 값으로 대체	ESS-USER-135 → EU1003005
필요	데이터 마스킹 기법 활용	ESS-USER-135 → EXX-UXXX-1XX

그림 7 데이터셋 민감정보 비식별화

데이터셋 민감정보 비식별화는 데이터 보안과 관련있다. 해당 사항은 데이터 보안 정책에 따라 진행한다. 데이터셋의 민감정보를 공개할 필요가 없는 경우 해당 정보를 제외하고 API를 구축하여 데이터셋을 공개한다. 민감정보를 포함하여 데이터셋을 공개할 경우 일련의 고유 번호와 함께 Key-Value 값으로 저장 후 해시 테이블에서 관리하고, 기존 데이터를 일련의 고유 번호로 대체하여 공개한다. 또는 데이터의 기본 틀을 유지한 채 데이터 마스킹 기법을 활용해 데이터셋 민감정보 비식별화를 달성한다.

5.7 다중 클라이언트 허용 처리(F-7), 서비스 요청 처리 속도(P-1), 페이지 응답 시간(P-2)

KETI	웹 기반 에너지 저장 장치 데이터 공개SW 기술 설계서	
	과제명	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW프레임워크 개발

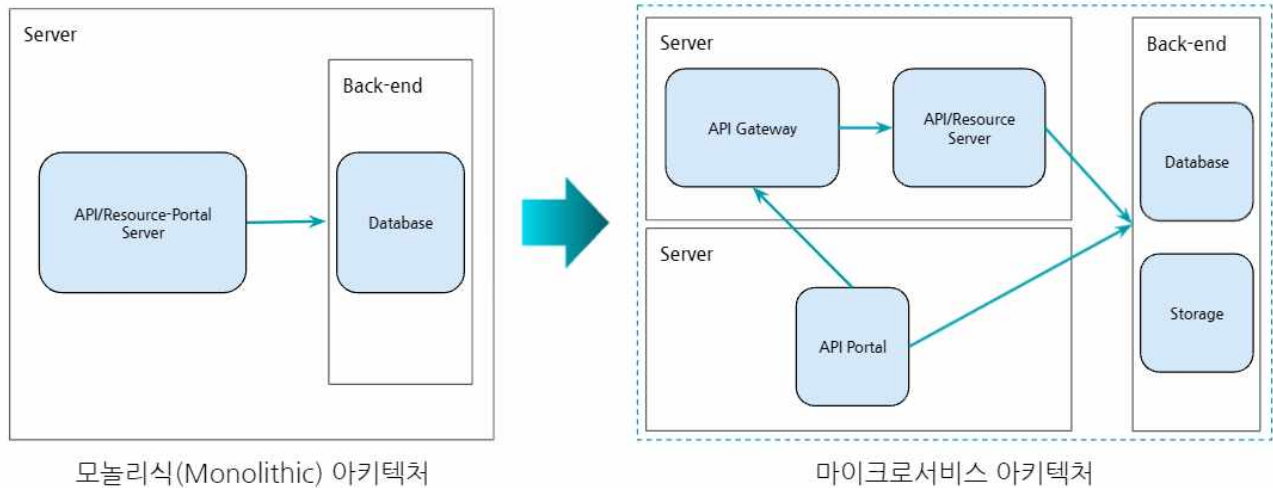




그림 8 공개 SW 서비스 아키텍처 비교 - 모놀리식 VS 마이크로서비스

공개 SW 서비스의 서버 성능을 높이기 위해 마이크로서비스 아키텍처 기반으로 서비스를 설계한다. 모놀리식 아키텍처로 설계할 경우 모든 시스템이 하나의 서버에 구성된다. API/Resource Server와 API Portal 시스템이 하나의 서버로 구성될 경우 API 호출하는 사용자와 API Portal을 이용하는 사용자의 요청이 한 곳으로 몰리는 트래픽 부하가 발생할 것이다. 트래픽 부하가 심해질수록 서비스 요청 처리 속도가 느려지고, 페이지 응답 시간이 길어지는 심각한 전체 시스템의 성능저하를 일으킨다. 그래서 API Portal과 API 제공 시스템을 분리(서로 다른 서버를 사용)하여 여러 클라이언트의 동시 요청을 처리하고, 서버의 응답 시간을 단축한다. 또한 Back-end 환경에 Storage를 도입하여 정적 이미지와, 미디어 파일 및 데이터셋 등을 관리하는 콘텐츠 분산을 통해 트래픽을 분산하여 효율적으로 트래픽 부하를 줄인다.

	웹 기반 에너지 저장 장치 데이터 공개SW 기술 설계서	
	과제명	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW프레임워크 개발

6. 용어

- API Gateway: 서버 최앞단에 위치하여 모든 클라이언트 API 호출에 대한 엔드 포인트를 통합하는 시스템
- API Portal: API를 쉽게 검색하고 액세스할 수 있도록 API Provider와 API Consumer 사이에서 다리 역할을 하는 시스템
- Proxy: 클라이언트와 서버 사이에 중계기 역할을 하여 클라이언트가 간접적으로 다른 네트워크 서비스에 접속할 수 있게 하는 시스템
- RESTAPI: REST는 효율적, 안정적이며 확장가능한 분산 시스템을 가져올 수 있는 소프트웨어 아키텍처로 이것을 따르는 API 설계 모델
- OAS(Openapi Specification): RESTful 웹 서비스를 설명, 생성, 소비 및 시각화하기 위한 기계 판독 가능 인터페이스 파일의 사양
- 직렬화(Serializer): 데이터 구조나 오브젝트 상태를 동일하거나 다른 컴퓨터 환경에 저장하고 나중에 재구성할 수 있는 포맷으로 변환하는 과정
- 분산 시스템: 네트워크에 연결된 여러 시스템들의 처리 능력을 이용하여 거대한 계산 문제를 해결하려는 분산처리 모델
- 시계열 데이터: 일정 시간 간격으로 배치된 데이터들의 수열
- Cloud Storage: 디지털 데이터를 논리 풀에 저장하는 시스템으로 클라우드 컴퓨팅을 통해 작동한다.
- Docker: 프로그램을 프로세스 격리 기들을 사용해 컨테이너로 실행하고 관리하는 기술 중 하나

	웹 기반 에너지 저장 장치 데이터 공개SW 기술 설계서	
	과제명	대규모 분산 에너지 저장장치 인프라의 안전한 자율운영 및 성능 평가를 위한 지능형 SW프레임워크 개발

7. Reference

- 오픈API와 오픈 플랫폼의 구성요소 - <https://www.2e.co.kr/news/articleView.html?idxno=210258>
- Google Cloud Platform API Gateway 아키텍처 - <https://cloud.google.com/api-gateway/docs/architecture-overview>
- 클라이언트-마이크로 서비스 간 직접 통신과 API 게이트웨이 패턴 비교 - <https://docs.microsoft.com/ko-kr/dotnet/architecture/microservices/architect-microservice-container-applications/direct-client-to-microservice-communication-versus-the-api-gateway-pattern>
- OAS Document - <https://spec.openapis.org/oas/latest.html>
- 소프트웨어(SW) 개발방법론(2014, 특허청)
- 행정·공공기관 웹사이트 구축·운영 가이드(2021-03, 행정안전부)

본 문서는 ‘특허청 소프트웨어(SW) 개발 방법론(2014, 특허청)’ 및 ‘SW 산출물 작성 가이드(소프트웨어 자산뱅크)’를 참고하여 작성했습니다.