

스마터 시티(Smarter City)의 메시징 인프라

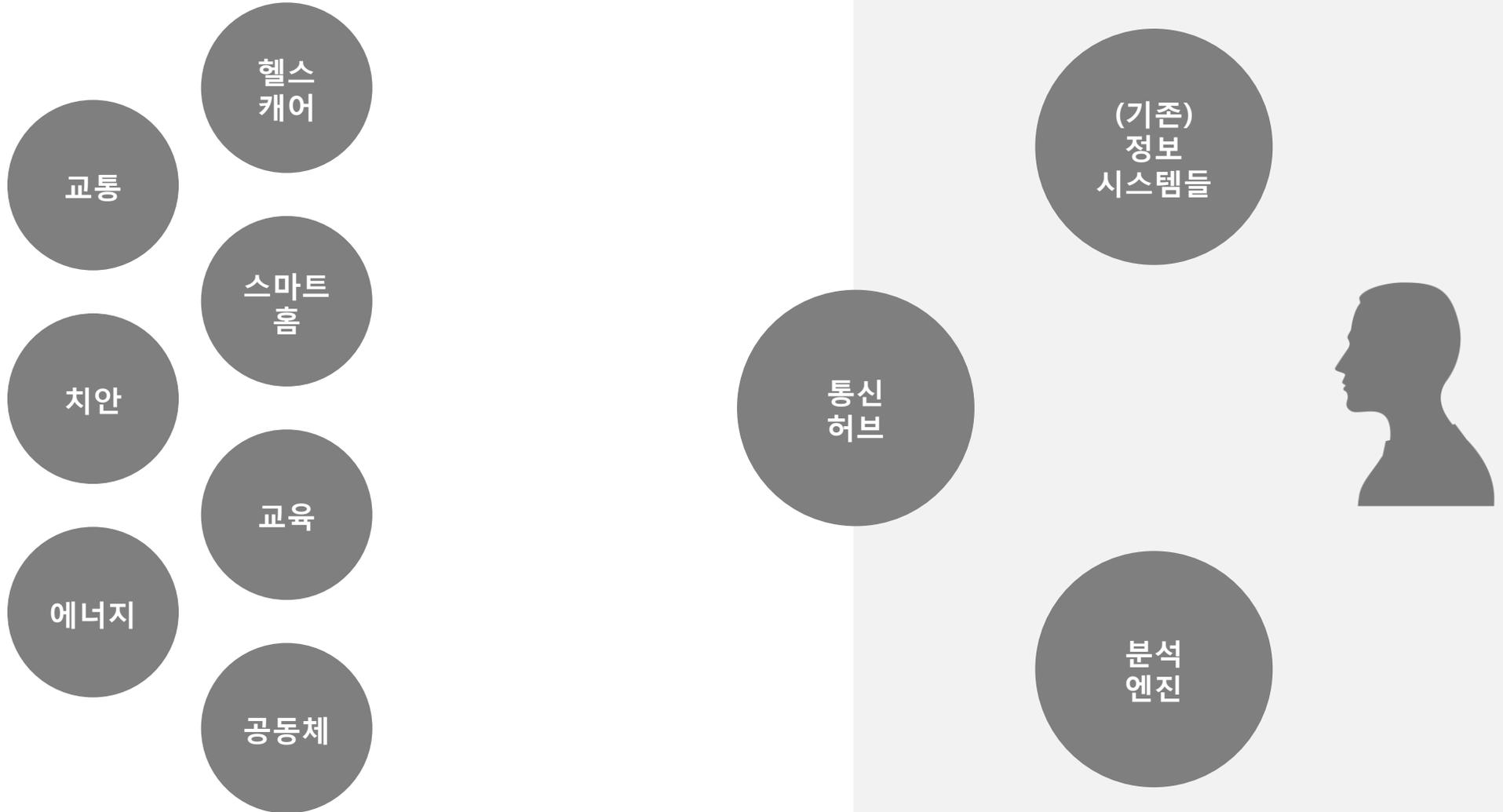
한국 IBM 소프트웨어 사업부
지용득

스마터 시티: IoT 적용 서비스들의 종합

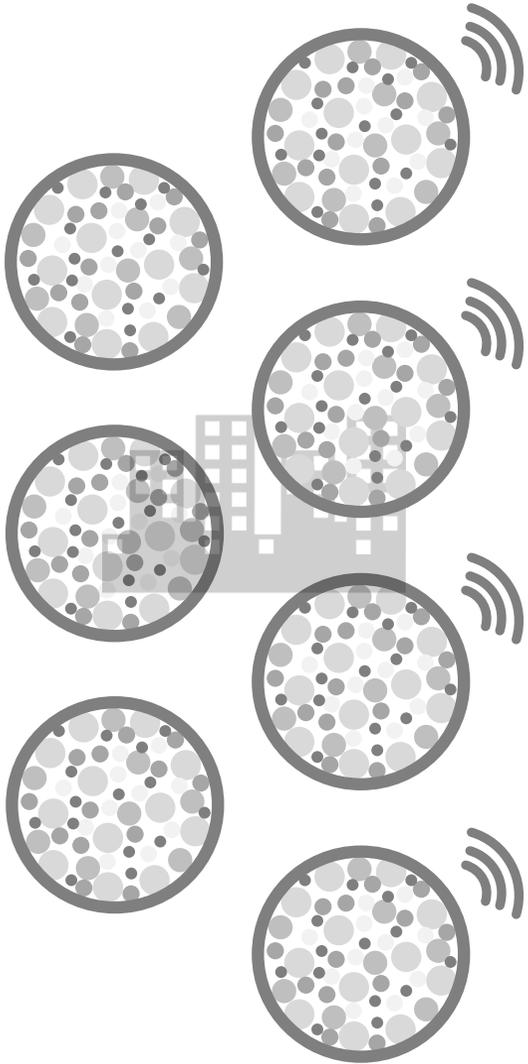


그밖에 지능적인 도시를 만들 수 있는 다양한 서브시스템들을 떠올려 보십시오.

간략하게 도식화한 전체 구조



대화하는 사물들



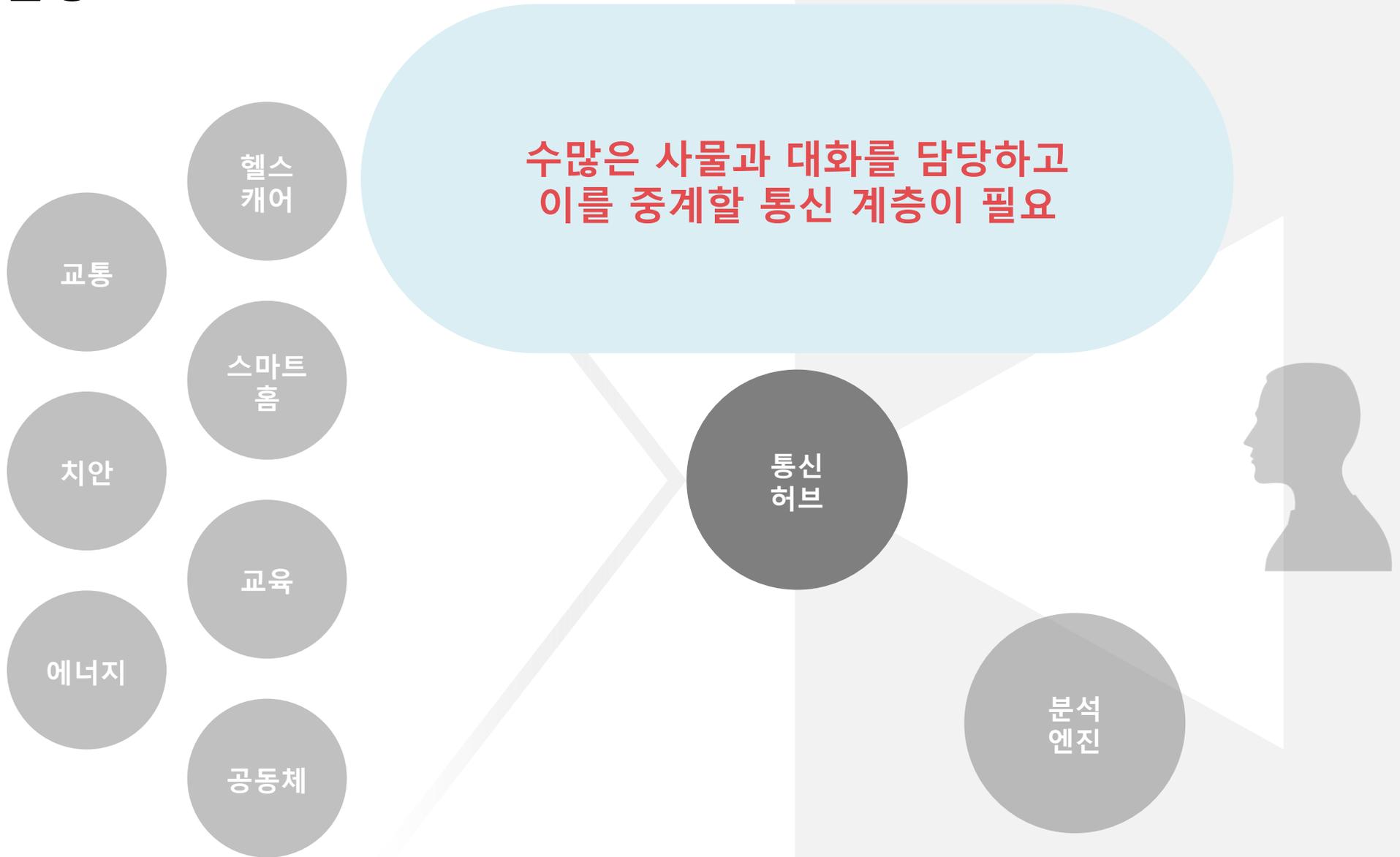
항상 동작하는 수많은 사물들이
지능형 도시의 근간

(기존)
정보
시스템들

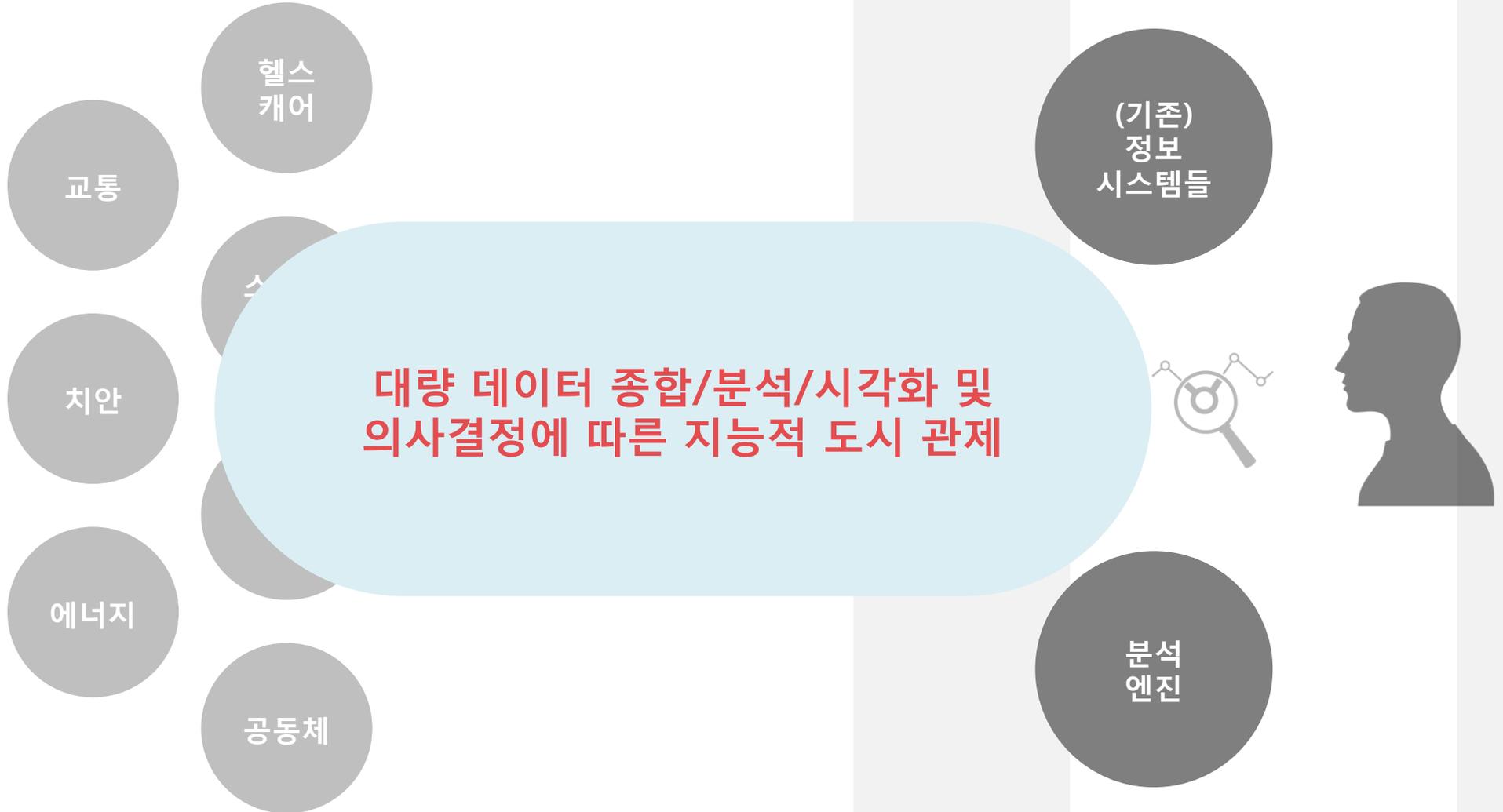
분석
엔진



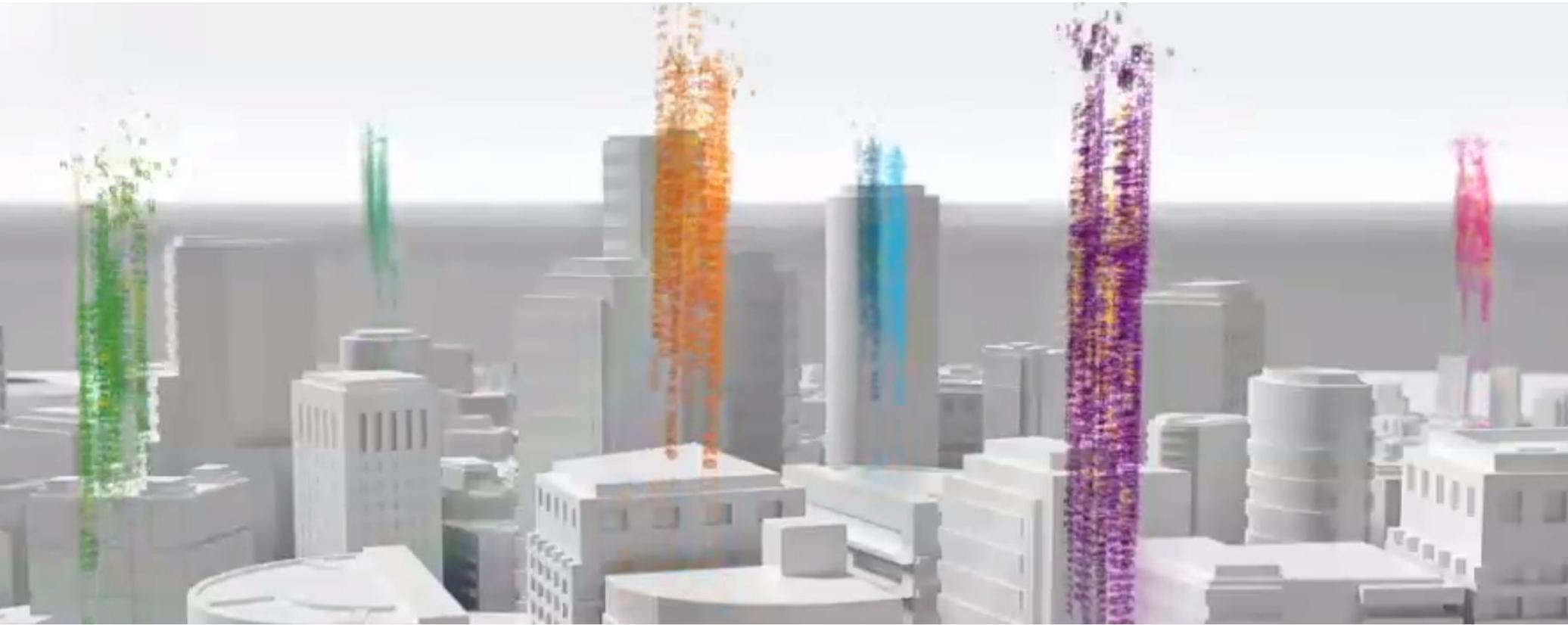
연결성



도시를 똑똑하게 만드는 City Command Center



결국 데이터, 그것도 대량의 데이터가 핵심



도시 곳곳에서 일어나는 일들이 수백만의 사물들이 전하는 데이터로 표현되고
이를 이해하고 지능적으로 대응하기 위해 노력하는 총체적 시스템
'결국 데이터의 문제'

도전: 사물들과 대화



사물들은,

- 전통적인 컴퓨터가 아님
- 제한된 컴퓨팅 환경
- 무언가 감지하는 능력
- 감지된 내용을 기반으로 통신
- 장치로서 외부의 제어 명령을 수행
- 휴대용 기기인 경우 제한된 전력에 의존
- 대개 무선 네트워크 상에서 통신

우리가 흔히 접하는 애플리케이션 실행이 어려운 환경,
사물들에게 적절한 기술은 무엇?

도전: 통신의 퀄리티



심장 박동을 모니터링 하는 기기가 감지한 이상 징후가 네트워크의 불안정으로 전달되지 않았다면?

...

화재를 감지한 장치가 유실을 방지하기 위해 화재 발생 신고를 중복해서 한다면? 또는 너무 늦게 도달한다면?

...

통신의 속도와 퀄리티에 대한 고려 필요

대응: 관찮은 사물 인터넷 메시징 프로토콜을 개발?

TCP 소켓 = 데이터 스트림 ≠ 메시징 프로토콜

- **경량성:** 사물이라는 제한된 연산 환경에서 동작하도록...
- **보편성:** 센서, 미터, 설비, 스마트 기기, 일반 컴퓨터에서도 동작하도록...
- **안정성:** 일관적이지 못한 무선 네트워크의 문제를 극복하도록...
- **신뢰성:** 유실이나 중복이 허용되지 않는 영역에도 쓰일 수 있도록...
- **유연성:** 사물 - 컴퓨터 간 양방향 통신이 가능하도록...
- **신속성:** 오버헤드 없이 빠르게 통신하도록...



MQTT(MQ Telemetry Transport): IoT 메시징 프로토콜

IBM이 개발한 사물 인터넷 + 모바일에 적합한 양방향 통신 메시징 규약

보편성, 유연성 – 오픈되어 있고 어디서든 사용 가능

- www.mqtt.org에 스펙 및 클라이언트 라이브러리 공개
- 2012년 Eclipse 오픈소스 M2M Working Group 내 M2M 프로젝트(Paho)에 MQTT 클라이언트 라이브러리 공여
- 다양한 환경의 클라이언트 라이브러리 지원(C/C++, Java, JS/Node.js, Object C, .Net, Delphi, Perl, PHP, Python...) & 오픈소스
- 다수의 사물과 효율적 통신을 위한 양방향 Pub/Sub 메시징 패러다임

경량성, 신속성 – 최소화된 프로토콜 오버헤드

- 가장 작은 메시지 사이즈는 2 byte까지 가능: 가변 길이 MQTT 헤더 + 애플리케이션 Payload
- Payload 데이터에 중립적: 별도의 다른 애플리케이션 헤더 불필요
- 클라이언트 라이브러리: C 버전은 30Kb, Java 버전은 100Kb 내외, 제한된 리소스를 갖는 사물에서도 무리없이 동작
- 고성능 메시징: 휴대용 장비에서 초당 천 단위의 메시지 교환 실현

MQTT(MQ Telemetry Transport): IoT 메시징 프로토콜

신뢰성 – 세 가지 메시징 신뢰성을 위한 QoS 레벨 제공

- 반드시 전달되어야 하는 중요 메시지에 대한 전달 보장 및 중복 방지
- 0 – 메시지가 최대 1번 전달, 유실 가능성 있음
- 1 – 메시지가 최소 1번 전달, 중복 전달 가능성 있음
- 2 – 메시지가 단 한 번, 정합성 있게 전달

안정성 – 연결을 잃었을 때 이를 보정하기 위한 자체 기능

- **Last Will and Testament:** 클라이언트가 불시에 연결을 잃을 경우 이벤트가 서버에서 발생, 서버 측에서 연결의 유실 여부를 인지
- **Durable Subscription:** 서버에 클라이언트의 구독(subscription) 정보 저장됨, 세션 종료 후 재접속 시에도 재작업 없이 Pub/Sub 유지
- **Clean Session 기능:** 연결 해제 후 다시 연결되었을 때의 이전 세션 유지/삭제 선택

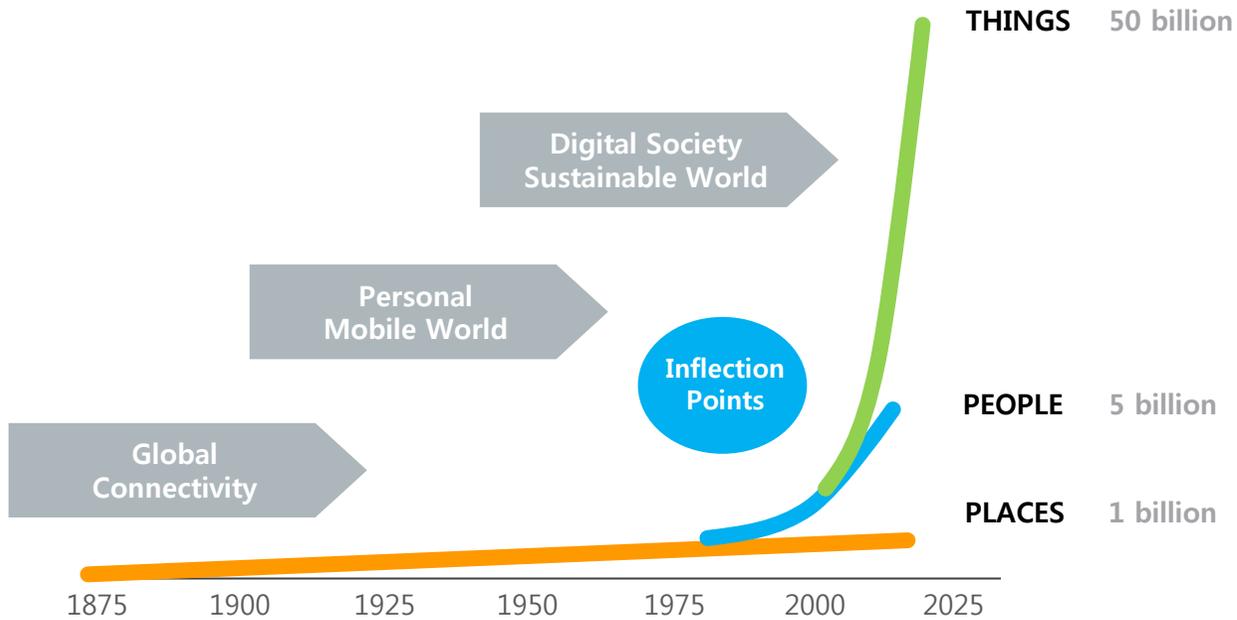
사물과의 양방향 통신의 혜택

Polling/Pulling을 하지 않기 때문에 사물의 전력, 데이터 사용에 매우 효율적
수백만의 사물이라는 규모로 바라보면 매우 큰 차이를 초래

- **93x** 더 많은 무선 네트워크 상의 메시지 송수신
- **11.89x** 더 적은 송신 시 전력 소모
- **170.9x** 더 적은 수신 시 전력 소모
- **8x** 더 적은 무선 대역 사용
- **50 μ s** 고속망 상에서 송수신 속도

	HTTPS	MQTT
수신 메시지 / 시간	1,708	160,278
% 배터리 / 메시지	0.01709	0.00010
송신 메시지 / 시간	1,926	21,685
% 배터리 / 메시지	0.00975	0.00082

도전: 사물들의 양적 팽창



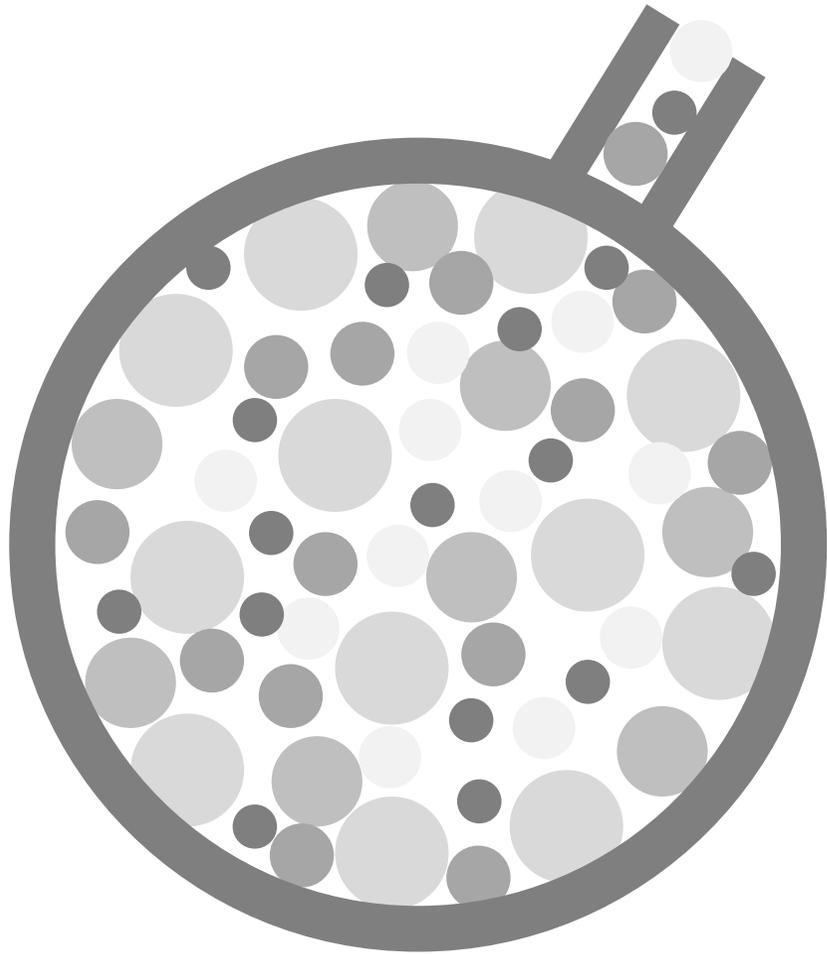
Ericsson AB, "Infrastructure Innovation – Can the Challenge be Met?" Sept 2010

GSMA는 소위 "Connected Life"를 구성할 사물의 시장이 2020년 경에는 약 \$4.5조 규모에 이를 것으로 전망*

- 2012년: 70억 개의 연결된 사물
- 2015년: 150-200억 개의 연결된 사물
- 2020년: 400-500억 개의 연결된 사물

* <http://www.gsma.com/newsroom/gsma-announces-the-business-impact-of-connected-devices-could-be-worth-us4-5-trillion-in-2020>

도전: 스마트 시티의 서버 측에는 높은 확장성이 필요



수백만의 사물의 동시 연결을 처리하며 동시에 다음과 같은 TASK들을 수행

- Pub/Sub 허브로서 중계
- 메시징 신뢰도 구현
- 연결 관리
- 메시징 보안(암호화/인증)
- 분석/제어 플랫폼 연계

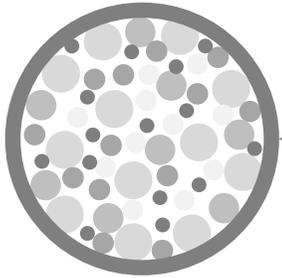


그리고 위 같은 일을 다음과 같이 하고자 함

- 낮은 지체 시간으로
- 높은 단위 시간 당 처리량으로
- 필요 시 쉽게 확장하며
- 운영과 관리도 용이하게
- 서비스의 중단도 없이

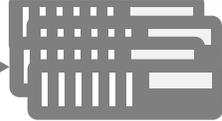
대응: 서버 측 확장성 확보 방안

수백만의 사물
동시 처리



HTTP

웹 서버 X 수십 클러스터*

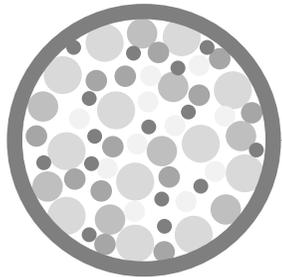


?

(기존) 정보 시스템들

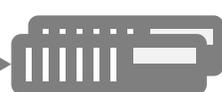


분석 엔진



MQTT

확장성 높은 MQTT 서버 X 2개**



💡

(기존) 정보 시스템들



분석 엔진

*Apache HTTP 서버의 경우 동시 처리량은 대략 25,000 정도로 알려져 있습니다.
**고가용성을 고려하여 최소 2개의 노드를 추천하기 때문에 2개로 표현되었습니다.

IBM의 상용 MQTT 서버

IBM의 상용 MQTT 서버 구현은 대규모의 스마트 시티/IoT를 겨냥한 고용량, 고속 솔루션으로 구성되어 있습니다.

소프트웨어

WebSphere
MQTelemetry

어플라이언스



MessageSight

지능형 도시의 규모에 필요한 IoT 메시징 서버

소프트웨어

WebSphere
MQTelemetry

- 인스턴스 당 수십만 이상의 사물 동시 연결 처리
- 초당 수천 이상의 메시지 송수신
- 메시지 당 매우 적은 리소스 사용
- WebSphere MQ를 통한 용이한 백엔드 연계
- 명령/설정 한 번으로 N개의 클러스터 구성

어플라이언스



MessageSight

- IoT를 위해 처음부터 설계된 특정 목적 하드웨어
- 천만 단위의 동시 연결, 초당 메시지 송수신
- 마이크로초 단위의 메시징 처리 능력
- 어플라이언스 특유의 비교할 수 없는 관리성과 확장성
- DMZ 구간에 최적화된 높은 보안성

스마트 시티 사례: 일본 기타큐슈 스마트 전력 관리 시스템



배경

- 2010년 일본 경제산업성에서 차세대 에너지 혁신 지역 사회 시스템의 파일럿 프로그램을 발족
- 2014년까지 탄소 배출량을 반으로 줄이고자 했던 기타큐슈 시가 선정
- 탄소 발생 감축을 위해 기존 발전 설비 외에 태양광 발전 설비의 비중이 높아짐
- 기상 상태에 영향이 높은 태양광 발전의 특성 상 새로운 전력 수급 및 과금 시스템을 요구

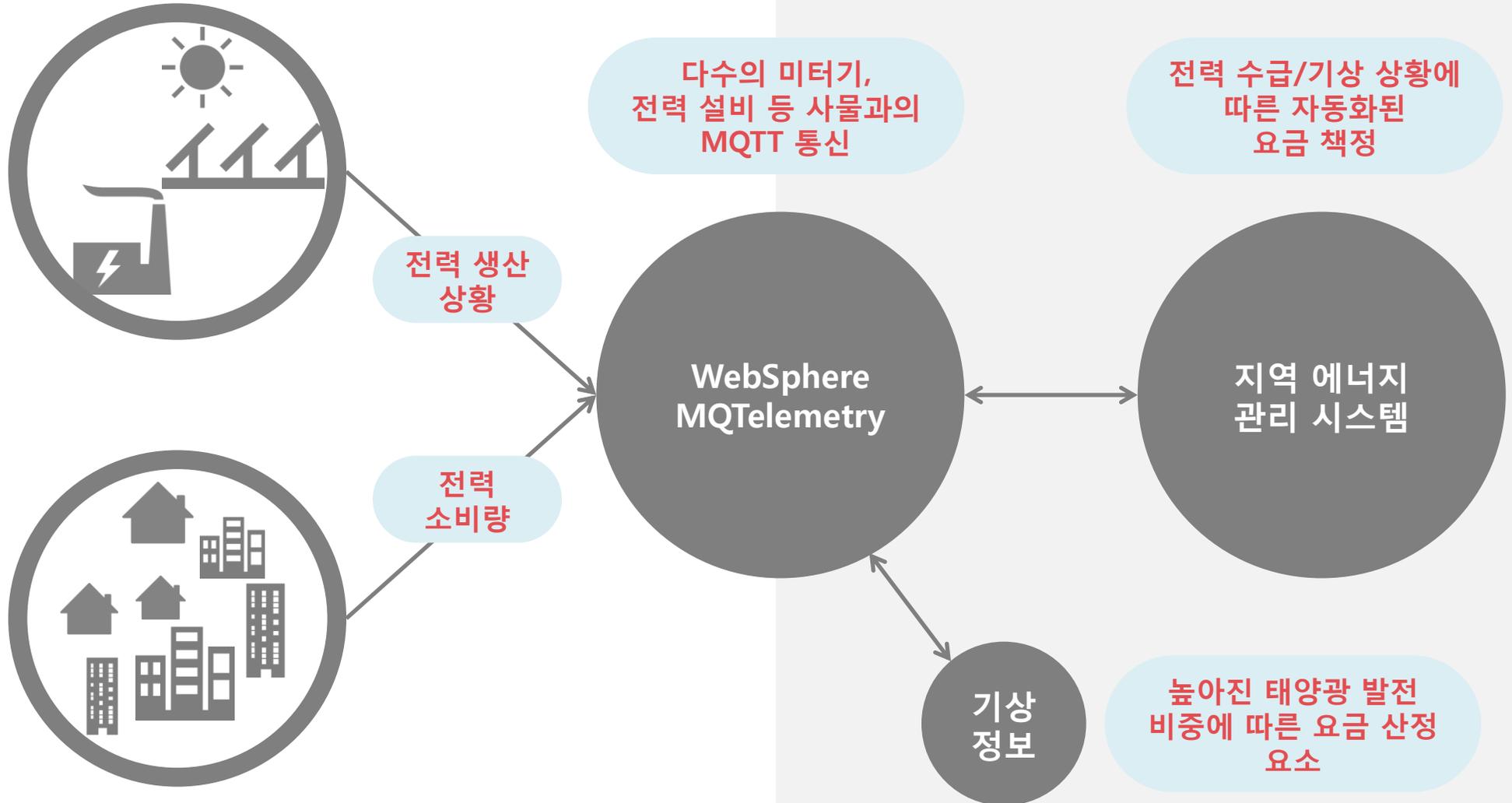
솔루션

- 주택과 건축물의 전력 소비 상황을 집계하기 위해 MQTT 프로토콜 적용
- 발전 설비로부터 얻어지는 전력 생산 현황 또한 MQTT 프로토콜로 집계
- 미터기, 설비 등과의 통신을 담당하는 MQTT 서버인 WebSphere MQ Telemetry 배치
- 전력 수급 상태에 따라 동적으로 전기 요금이 변하는 지역 에너지 관리 시스템으로 집계 데이터 연동

효과

- 수급 상황에 따라 동적으로 과금되는 새로운 요금 체계 수립
- 지역 소비자들에게 합리적 전기 소비에 대한 정보를 제공함에 따라 전체적인 전력 소비량 10~15% 감소
- 장치/사물들과 백엔드 시스템들과의 연동을 단일 솔루션 - WebSphere MQ Telemetry로 수행하여 IT 투자에 대한 효율화

스마트 시티 사례: 일본 기타큐슈 스마트 전력 관리 시스템



스마트 시티 사례: Italy Bolzano, “Living Safe” 프로젝트



배경

- 인구의 22.5%가 75세 이상의 고령인구이고 향후 급속한 노령화가 예상됨
- 경제 활동 인구의 감소로 세수는 정체 또는 감소
- 기존 공공 서비스 부문에 대한 유지보수 및 투자가 필요하나 투자 여력이 없음
- 고령층에 대한 보건 서비스 등을 확충해야 하나 위의 이유로 비용에 대한 이슈가 있음

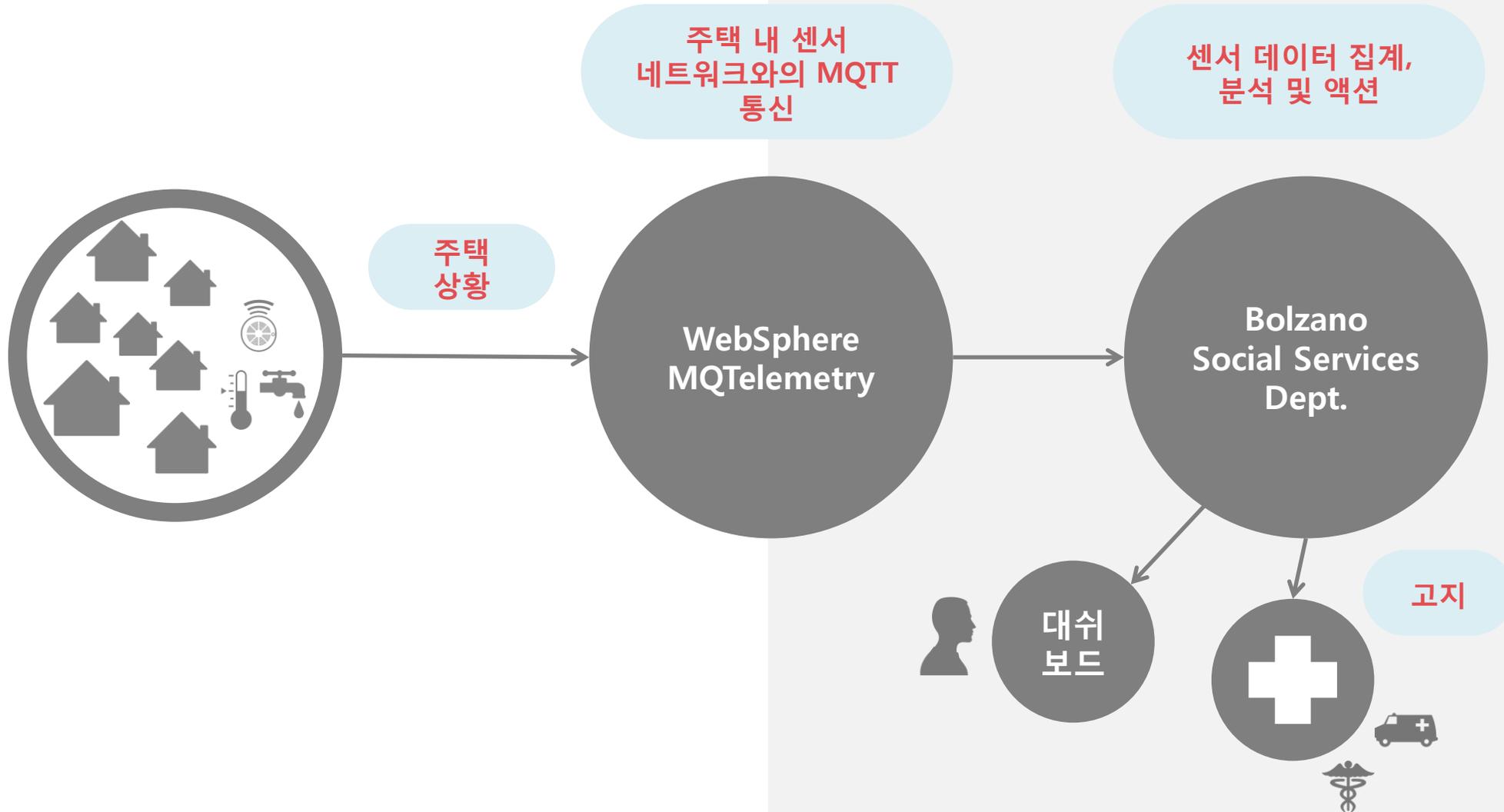
솔루션

- 주로 가택 생활을 하는 고령 인구의 주택 상황(온도, 이산화탄소 농도, 누수 탐지 등)을 모니터링하는 센서 및 통신 유닛 설치
- 주택 모니터링 정보는 실시간으로 Bolzano 사회 복지 부서로 전송되도록 구성하며 Zigbee와 MQTT 프로토콜을 사용하여 집계
- 데이터 전송의 신뢰도를 확보하기 위해 센서를 위한 통신 허브로 WebSphere MQ Telemetry 배치
- 집계 데이터를 분석하여 대쉬보드에 현 상황을 표시
- 이상 징후 감지 시 거주인의 가족, 담당 복지사, 911 등에 다양한 경로로 고지

효과

- 거주지 내 데이터를 실시간 모니터링하고 이를 지능적으로 분석함으로써 적시에 리소스를 투입
- 제한된 운영 예산으로 시의 복지 시스템 운영을 최적화/고도화한 새로운 운영 모델 제시
- 노령 인구에 대한 헬스 케어 지원으로 경제 활동 인구에 부양 부담을 줄임
- 2012년 COMPUTERWORLD HONORS PROGRAM Laureate 선정

스마트 시티 사례: Italy Bolzano, "Living Safe" 프로젝트



감사합니다

Credits

Some of the icons used in this presentation/handout are under the license agreement Creative Commons 3.0(<http://creativecommons.org/licenses/by/3.0/us/>) and attributed to their respective author as listed below:



User by Simon Child from The Noun Project



City by les vieux garçons from The Noun Project



Analytics by Christopher Holm-Hansen from The Noun Project



Temperature by Ashley Reinke from The Noun Project



Traffic Light by Arthur Shlain from The Noun Project



Smoke Alarm by Theresa Stoodley from The Noun Project



Television by Yair Cohen from The Noun Project



Heartbeat by Jardson A. from The Noun Project



Broken Machine by Anton Håkanson from The Noun Project



Like by Shmidt Sergey from The Noun Project



Light Bulb by Chris Brunskill from The Noun Project