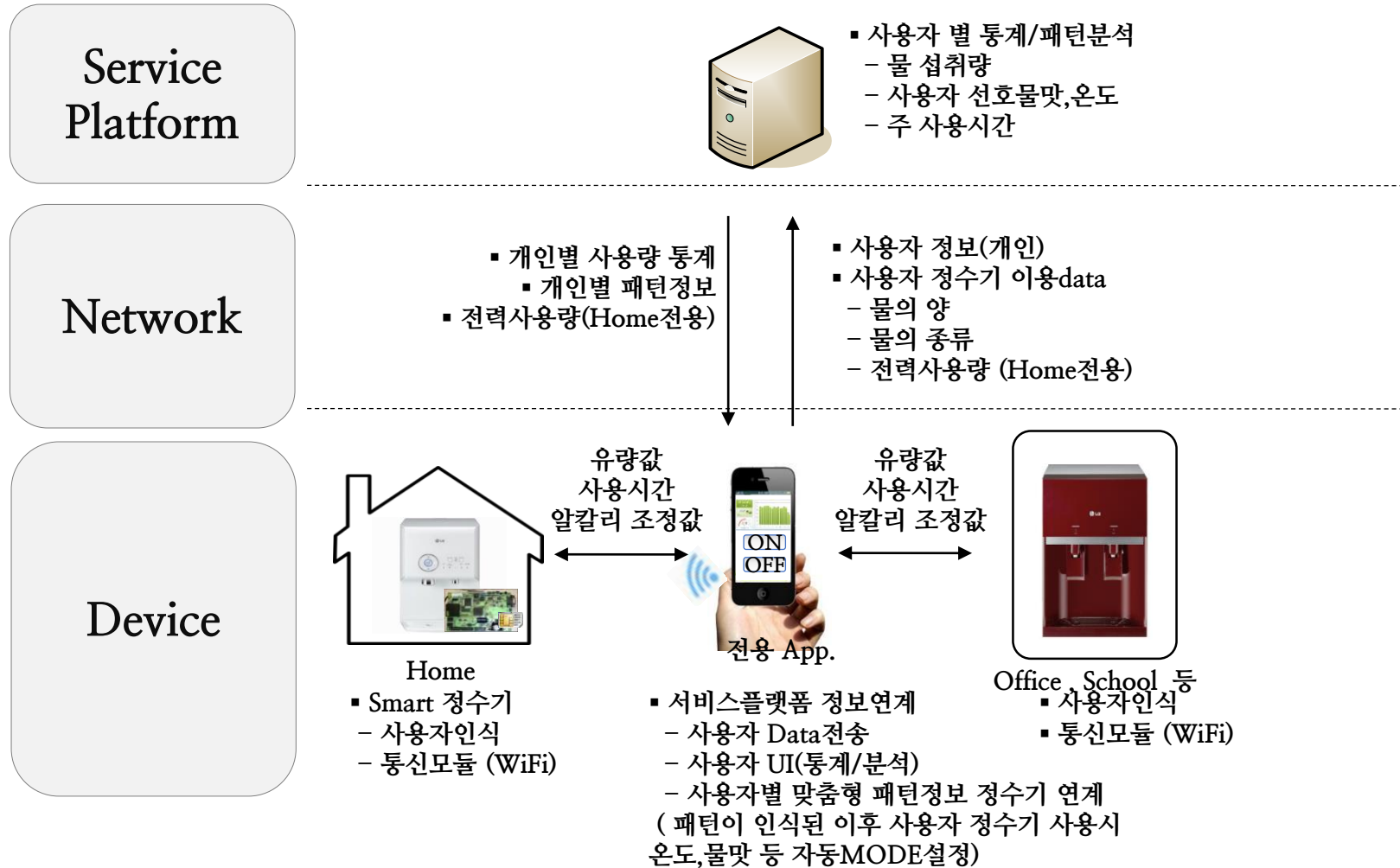


Smart water diary 서비스플랫폼 제안

2014/09/04

(주) 엠앤엘솔루션

서비스 개요



서비스 개요 :: 고려사항

스마트워터 다이어리를 위한 고려사항

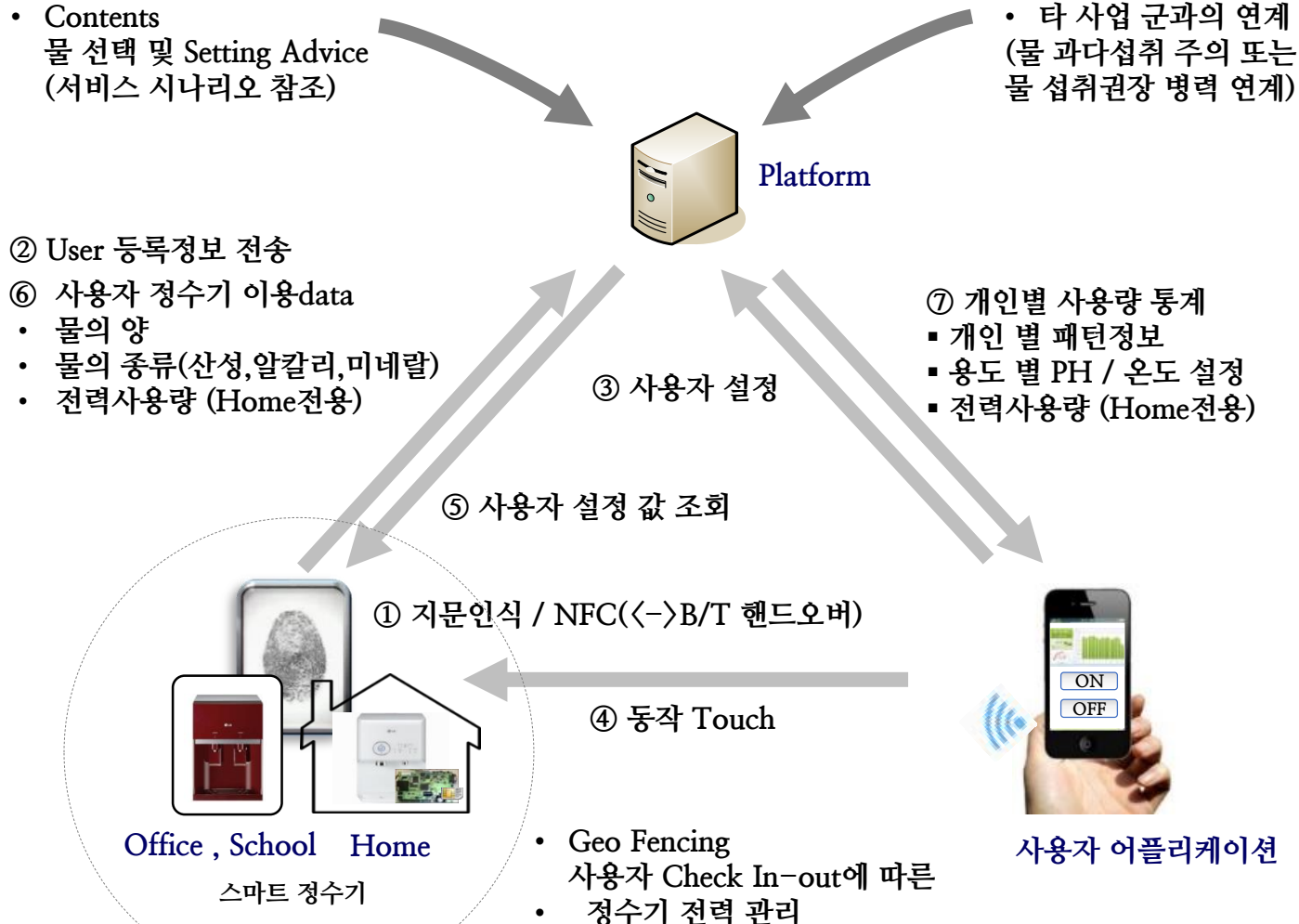
- 하루 물 섭취 량: 체중 x 33 (ml) => 50Kg의 여성은 1일 1.6리터 이상
- 성인 1.6 ~ 2.0리터 권장 (WHO는 3리터 권장)
- 1회 섭취 적정 량: 200 ml을 한모금씩 끊어서 섭취 (1회 500ml 이상은 건강에 좋지 않음)
- 1시간에 1번씩 마시는 것 권장
- 적어도 1일 8회 권고 (기상 전 1회, 3회 x {식사 전 30분, 식후 1시간 뒤}, 취침 전 1회)
- 고온환경에 노출되어 있거나 운동 후에는 4리터 정도 권장
- 물 섭취와 관련된 질환에 유의 (섭취권장 또는 자제를 요하는 질병 군)

=> 고객의 동적/정적 정보 요구

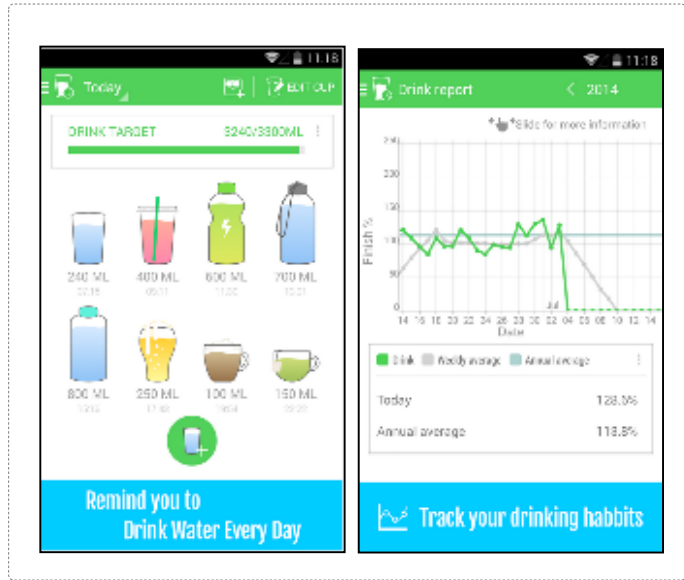
정적 정보: 고객구분, 나이, 성별, 체중, 병력

동적 정보: 위치/동선 (날씨/온도, 운동량), 시간, 물섭취정보(온도, 유량, PH)

서비스 흐름도



1. IoT, 정수기관련 라이프로그 수집
2. Smart 절전 서비스



[물 섭취 다이어리]

서비스 시나리오

구분	상세 기능
Contents 정보 제공 시나리오	
날씨 정보 연동 맞춤 정보	<p>아침 물 먹을 때 지역 날씨 정보 연동 통해 물 휴대 권고</p> <ul style="list-style-type: none"> . 금일 기온은 섭씨 OO도 햇빛이 강합니다. 물을 챙겨 가세요. . 금일 습도는 OO%로 건조한 날씨입니다. 물 챙겨 드셔서 수분 유지 하세요. . 기온이 영하 OO도입니다. 외출하실 때에는 따뜻한 물을 챙겨 가세요. . 미세먼지가 많은 날이었습니다. 물을 많이 챙겨 드세요.
스마트 폰 통한 운동량 체크 후 권장 수분 섭취량 맞춤 정보 제공	<p>스마트 폰 운동 체크 앱과 연동하여 운동 시간, 운동량 데이터 분석 통한 권장 수분 섭취량 제공</p> <ul style="list-style-type: none"> . 운동 시간 O 시간, 운동량은 OO Kcal입니다. OOO ml의 물을 섭취하세요
일일 권장 수분 섭취량 체크 프로그램	스마트 정수기 물 섭취량 자동 기록 + 개인 수동 기록 지원 통한 수분 섭취 관리 프로그램.
정수기 제어	
선호 물 온도 및 PH 값 setting	전용 App에 선호 설정 값 셋팅
사용자 상황 別 PH 값 정보 제공	상황 별 적정 온도 / PH 값 정보 제공
전원 관리 (절전)	
정수기 절전 시나리오	<p>해당 정수기 등록 사용자 지오 펜싱 통한 절전 프로그램 구동.</p> <ul style="list-style-type: none"> . 가족 구성원 전부 외출 중일 때 정수기 절전 모드 동작

서비스 시나리오 :: 고객구분

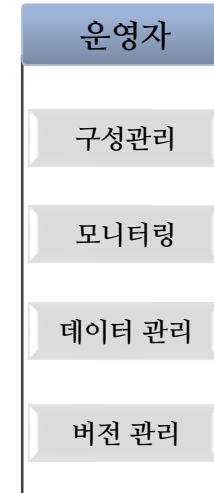
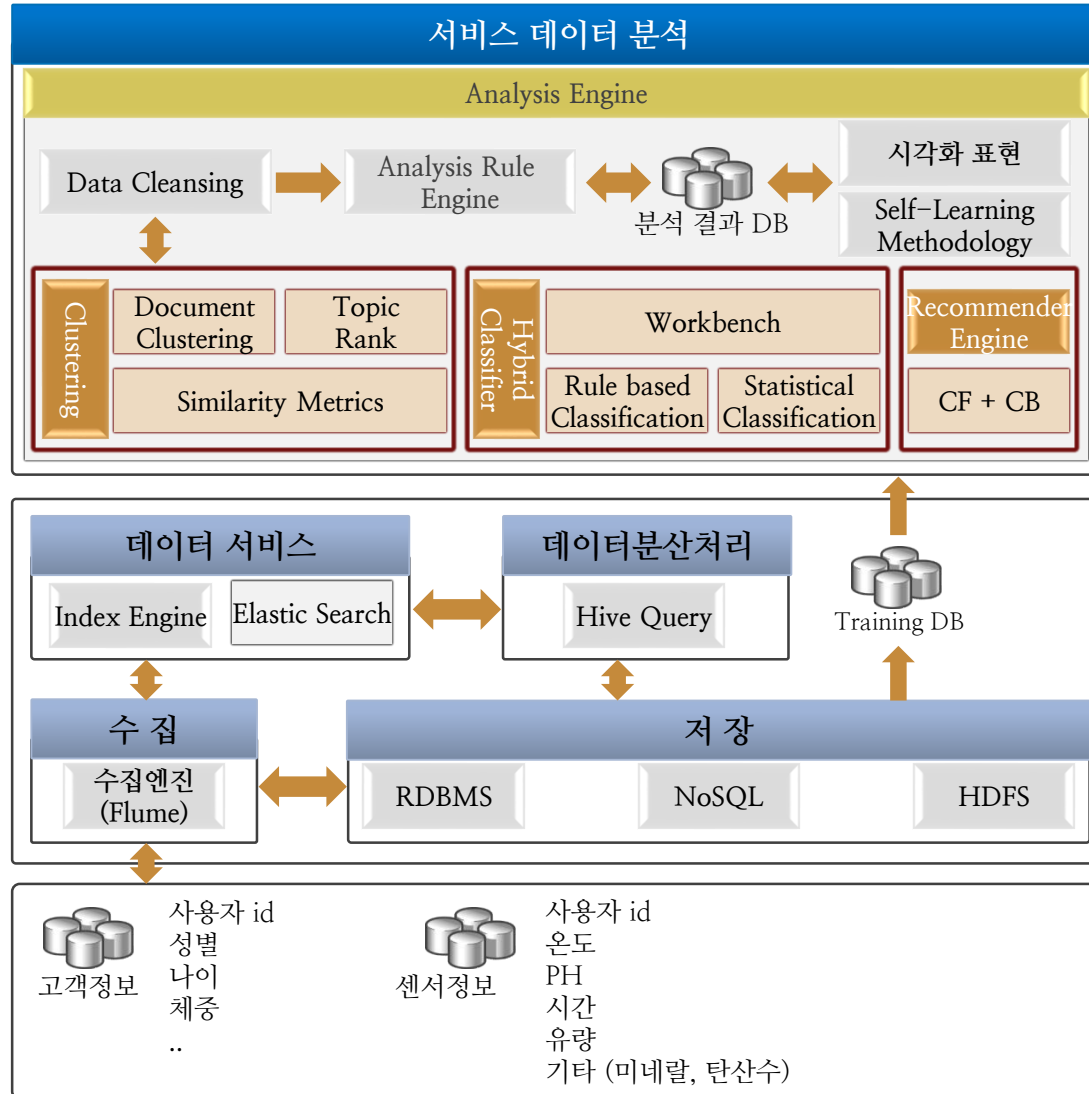
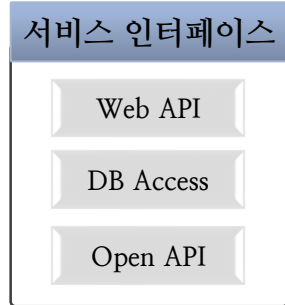
지문인식 (안)

- 지문인식을 통하여 개인에 대한 구분 제공
- 공동환경에서 위생문제, 인식속도, 지문등록과 개인정보 mapping 절차가 복잡해짐.

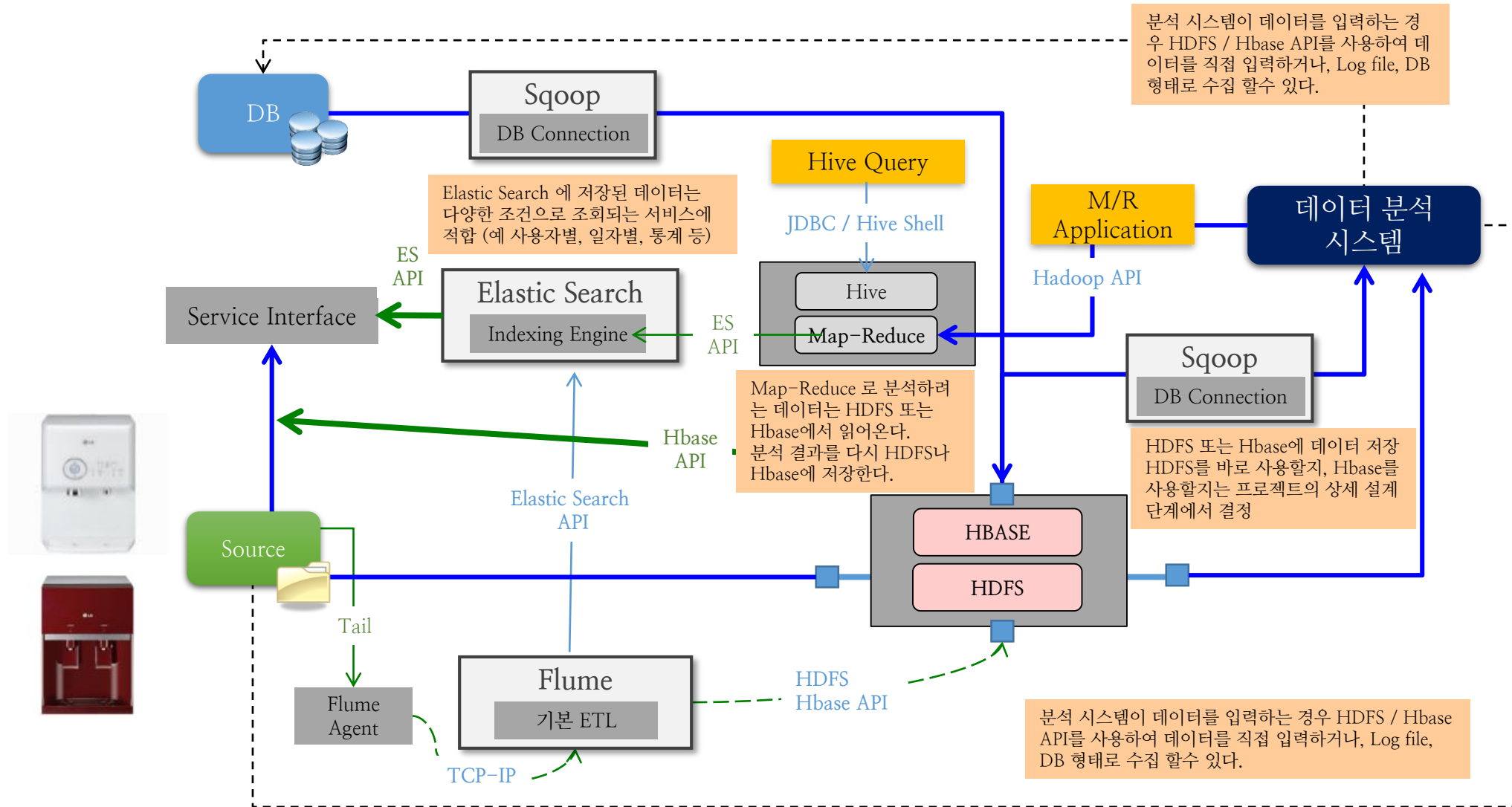
NFC 탑재-폰 활용

- NFC 직접통신 방식 대신, NFC 태깅 사용 (= NFC ⇔ B/T 핸드오버; 오디오에서 많이 사용)
 - 개인의 정적 정보를 스마트-앱에 등록(uuid 발급)하면 플랫폼에 동기화
 - NFC를 태깅하면, 고객정보(=uuid)를 정수기에 전달하고
 - 정수기는 해당 uuid를 통하여 플랫폼과 통신하여 프로필 자동설정 기능이 동작
- => 항상 스마트폰을 들고다녀야 라이프로그(물 섭취 정보) 수집이 가능

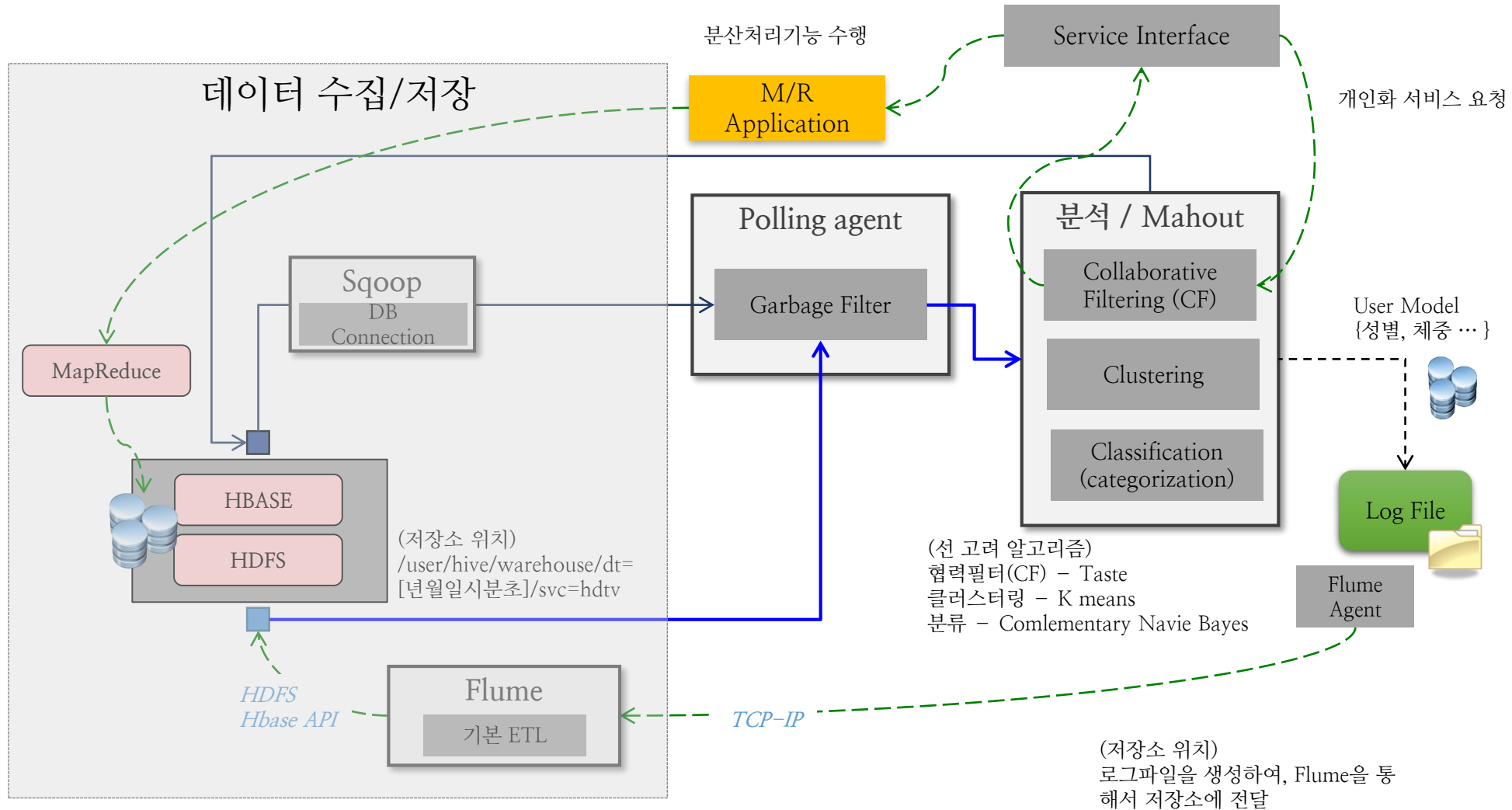
서비스 플랫폼 :: 시스템 전체 아키텍처



서비스 플랫폼 :: 수집/저장 시스템



서비스 플랫폼 :: 분석 시스템



서비스 플랫폼 :: 분석 알고리즘

수집된 데이터에 대해서는 다음과 같은 분석기능(hadoop mahout)이 사용되며, 서비스의 유형 또는 데이터 셋의 성격에 따라 세가지 분석 절차를 다르게 수행

Classification

Distributed Naive Bays

Complementary Naive Bays

각 사건 들이 서로 영향을 주지 않는 독립적인 관계 (Independence) 라고 가정한 가장 단순한 확률 분류기

$$p(C|F_1, \dots, F_n) = \frac{p(C) p(F_1, \dots, F_n|C)}{p(F_1, \dots, F_n)}$$

현재의 항목에 별도의 레이블을 지정하여 분류

Clustering

K-means

Fuzzy K-means

이전 반복의 중심 또는 중앙으로부터 항목이 떨어져 있는 거리를 기반으로 항목을 k개의 클러스터로 클러스터링한다.

Canopy

빠른 클러스터링 알고리즘이며 다른 클러스터링 알고리즘의 초기 시드를 작성하는 데도 사용

Dirichlet

가능성이 있는 여러 모델을 기반으로 하는 클러스터가 제공되므로 특정 클러스터 보기를 조기에 결정하지 않아도 된다는 장점

Mean Shift

클러스터 수를 미리 알 필요가 없으며 임의 형태의 클러스터를 생성할 수 있는 알고리즘

다량의 데이터 셋이 존재할 경우 유사한 항목끼리 그룹화

Collaborative Filtering

User Item Model

Slope-one

Taste

유사성추출 (추천)

Euclidean Distance
Person Correlation
Manhattan distance
Cosign similarity

사람들 중에 나와 비슷한 기호를 가지고 있는 사람이 추천점수를 순위를 매겨서 활용

서비스 플랫폼 :: 노드 별 SW 스택

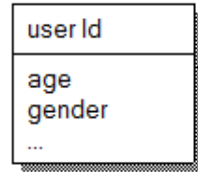
App Server	DB Server	Data Node	Name Node	Jobtracker Node	Mgmt Node
Analysis Logic (Mahout)	MySQL	Hbase	ZooKeeper	ZooKeeper	ZooKeeper
Polling Agent	RHEL	Task Tracker	NFS Client	Hbase master	JAVA
iBatis		JAVA	NameNode	JobTracker	RHEL
Apache/Tomcat		RHEL	JAVA	Task Tracker	
Spring			RHEL	JAVA	
RHEL				RHEL	

각 node의 물리 구성은 수집된 센서크기에 따라 가변적으로 구성됨.

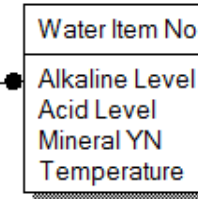
- 데이터 node를 제외한 모든 node는 2+0 구성
- 데이터 node는 성능을 고려할 경우 많을수록 좋으며, node의 수는 내부 스토리지 크기에 영향을 크게 받으며, storage의 크기는 수집/저장(누적)할 센서정보 볼륨 x 3 / 10으로 산출됨 (3 copy에 압축율 1/10 적용 시)

서비스 플랫폼 :: 스마트워터다이러리 / 추천

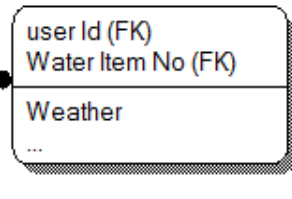
User Profile



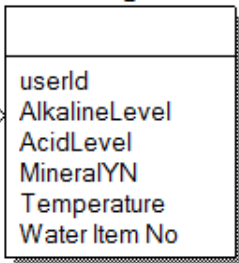
Water Item



Recommend Water



Water Log Data



```

    (1,0),(14,1),(10,0),(16,0),(10,2),(2,1),(2,2),(9,0),(12,0),(9,0),(11,2),(8,2),(5,2),(4,0),(5,2),(1,0),(9,1),(1,1),(9,1),(10,0),(14,1),(11,0),(11,2),(9,0),(2,0),(11,2),(15,1),(6,0),(4,0),(10,1),(2,1),(3,1)
    ...
  
```

```

    center : 5 {0:4.346590909090909,1:0.6566558441558441}, count : 7185
    center : 0 {0:1.3839373163565132,1:0.8452497551420176}, count : 3850
    center : 6 {0:3.0476758045292014,1:1.7592371871275327}, count : 4425
    center : 7 {0:6.747419132828631,1:1.1348933241569168}, count : 8118
    center : 3 {0:15.62136832239925,1:0.7919400187441424}, count : 5139
    center : 2 {0:10.507549361207898,1:0.5017421602787456}, count : 4305
    center : 1 {0:14.015711645101664,1:1.4279112754158965}, count : 5818
    center : 8 {0:12.406444906444907,1:0.840956340956341}, count : 4598
    center : 4 {0:10.050473186119874,1:2.0}, count : 3382
    center : 9 {0:8.522968197879859,1:0.5147232037691402}, count : 3180
  
```

```

    wt: 1.0 distance: 0.9283614659488414 vec: [0:1.000] belongs to cluster 0
    wt: 1.0 distance: 0.42819962099453085 vec: [14:0.00, 1:0.00] belongs to cluster 1
    wt: 1.0 distance: 0.7136886922627611 vec: [0:10.000] belongs to cluster 2
    wt: 1.0 distance: 0.8777989180736293 vec: [0:7.000] belongs to cluster 3
    wt: 1.0 distance: 0.0504731861198735 vec: [10:0.00, 2:0.00] belongs to cluster 4
    wt: 1.0 distance: 0.6352014392786118 vec: [2:0.00, 1:0.00] belongs to cluster 0
    wt: 1.0 distance: 1.074984336378057 vec: [2:0.00, 2:0.00] belongs to cluster 6
    wt: 1.0 distance: 0.7017829555726863 vec: [0:9.000] belongs to cluster 9
    wt: 1.0 distance: 0.9340262466171093 vec: [0:12.000] belongs to cluster 8
    wt: 1.0 distance: 0.7017829555726863 vec: [0:9.000] belongs to cluster 9
    wt: 1.0 distance: 0.9495268138801345 vec: [11:0.00, 2:0.00] belongs to cluster 4
    wt: 1.0 distance: 1.5222905075549995 vec: [8:0.00, 2:0.00] belongs to cluster 7
    wt: 1.0 distance: 1.4938262821102337 vec: [5:0.00, 2:0.00] belongs to cluster 5
    wt: 1.0 distance: 0.7425107109856945 vec: [0:4.000] belongs to cluster 5
    wt: 1.0 distance: 1.4938262821102337 vec: [5:0.00, 2:0.00] belongs to cluster 5
    wt: 1.0 distance: 0.9283614659488414 vec: [0:1.000] belongs to cluster 0
    wt: 1.0 distance: 0.6804799109408459 vec: [9:0.00, 1:0.00] belongs to cluster 9
    wt: 1.0 distance: 0.4139510854855279 vec: [1:0.00, 1:0.00] belongs to cluster 0
    wt: 1.0 distance: 0.6804799109408459 vec: [9:0.00, 1:0.00] belongs to cluster 9
    wt: 1.0 distance: 0.7136886922627611 vec: [0:10.000] belongs to cluster 2
    wt: 1.0 distance: 0.42819962099453085 vec: [14:0.00, 1:0.00] belongs to cluster 1
    wt: 1.0 distance: 0.7030311707513005 vec: [0:11.000] belongs to cluster 2
    wt: 1.0 distance: 0.7017829555726863 vec: [11:0.00, 2:0.00] belongs to cluster 4
    wt: 1.0 distance: 0.7017829555726863 vec: [0:9.000] belongs to cluster 9
    wt: 1.0 distance: 1.0459351694754582 vec: [0:2.000] belongs to cluster 0
    wt: 1.0 distance: 0.9495268138801345 vec: [11:0.00, 2:0.00] belongs to cluster 4
  
```

스마트워터다이러리를 위한
고객의 동적/정적 정보
{데이터셀}

고객성향분석을 위한 군집(clustering) 예시
⇒ 현재의 사용패턴이 어느성향에 속하는지를 판단

신규사용자에게 유사한 정적 프로필을 가진 사용자의
패턴을 제시함

- 클러스터링, 추천에 사용예정

서비스 플랫폼 :: 정수기 전원관리

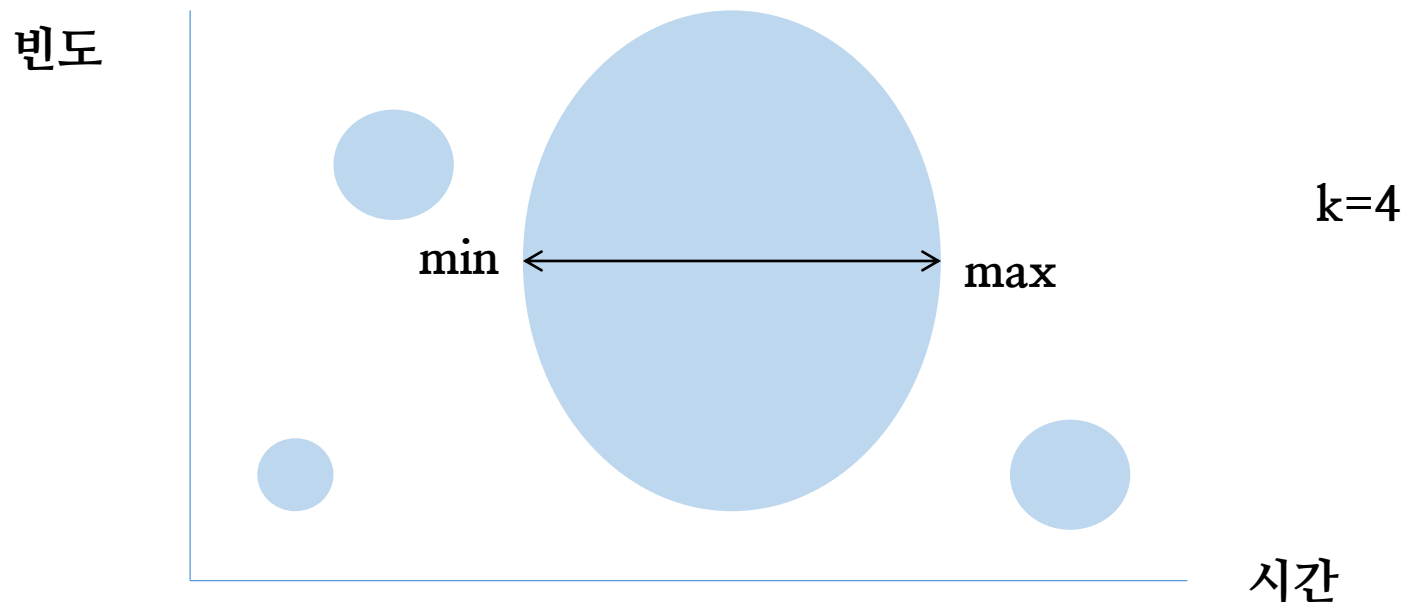
정수기 別 센서 정보 수집 - { 기기 id, 시간대 }

수집된 센서정보를 기기 id 별로 clustering 적용 (mahout / k-means 적용)

K-means의 적정 값, k는 여러 차례 시험데이터를 산출한 후 적정 k 값(cluster 수)을 도출

가장 많이 사용하는 패턴의 cluster의 1일 시간값 (min, max)를 도출하고

해당범위시간 外 를 해당 정수기 가정/사무실 환경에서의 전력 사용 프로필로 설정 제안



서비스 플랫폼 구축일정

구분		M+0	M+1	M+2
요구분석	요구분석	■		
	데이터 셀/모델링/입.출력정의	■		
기능개발	Hadoop core 구축	■		
	센서정보 수집 (Flume/Sqoop)	■		
	인덱스 (Elastic-Search)	■		
	저장관리(HDFS/HBase)	■		
	분석 (Mahout)	■	■	
	M/R(Map/Reduce)		■	
	가시화		■	■
	통계/RDB 저장		■	■
	WAS/OpenAPI		■	■
	시험			■

서비스 플랫폼 구축비용

구분		공급단가	개수	공급비용
하드웨어	X86 서버 (2P16C 16GB MEM) <ul style="list-style-type: none"> - Name node 2 - Data node 4 - App node 2 - DB node 2 - Job-Tracker node 2 - Mgmt node 2 	10,000,000	14	140,000,000
개발비용	기획(웹) 1명	7,000,000	3 M/M	21,000,000
	디자인 1명	6,000,000	3 M/M	18,000,000
	개발 (8명) <ul style="list-style-type: none"> - Sqoop/Flume 수집 1명 - HDFS/HBase/Hive/RDB 저장 1명 - MR programming 1명 - Mahout 분석 2명 - 분석.가시화(Visualization) 1명 - Web/WAS/통계/OpenAPI 2명 	8,500,000	24 M/M	204,000,000
합계				383,000,000

감사합니다.