

# 빅데이터 솔루션을 활용한 스마트미터 전력량 예측 및 분석

죽음의 불4조  
오진영, 이희철, 최준혁



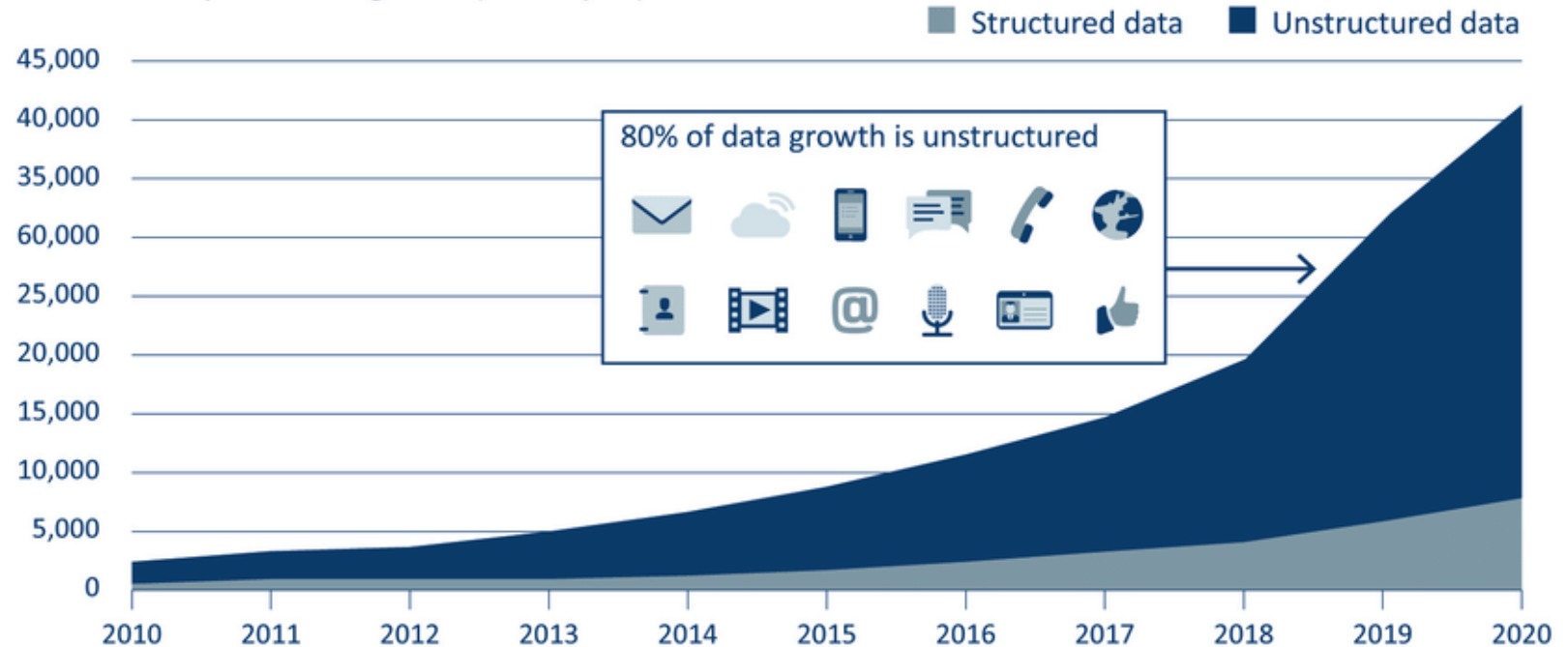
1. 프로젝트 배경
2. 프로젝트 도메인
3. 시스템 아키텍처
4. 솔루션 아키텍처
5. 결론 및 한계



# 프로젝트 배경

## Massive growth in unstructured content

Worldwide corporate data growth (in exabytes)



Source: The Digital Universe

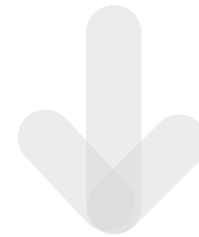
- 4차 산업 이후 최근 2년간 발생한 데이터는 전 세계 데이터의 80% 차지
- 2020년 - 40,000 이상 엑사바이트 증가 -> 80% 비정형 데이터
- 비정형데이터(SNS, IoT, 이미지, 음성, 비디오)는 분석이 필요

# 프로젝트 배경

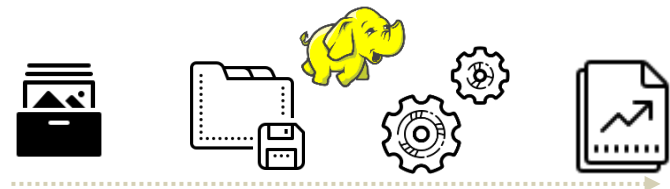


다양한 분야에서 발생하는 데이터는 서버 혹은 문서로 저장-기록 되고 모든 것이 데이터화

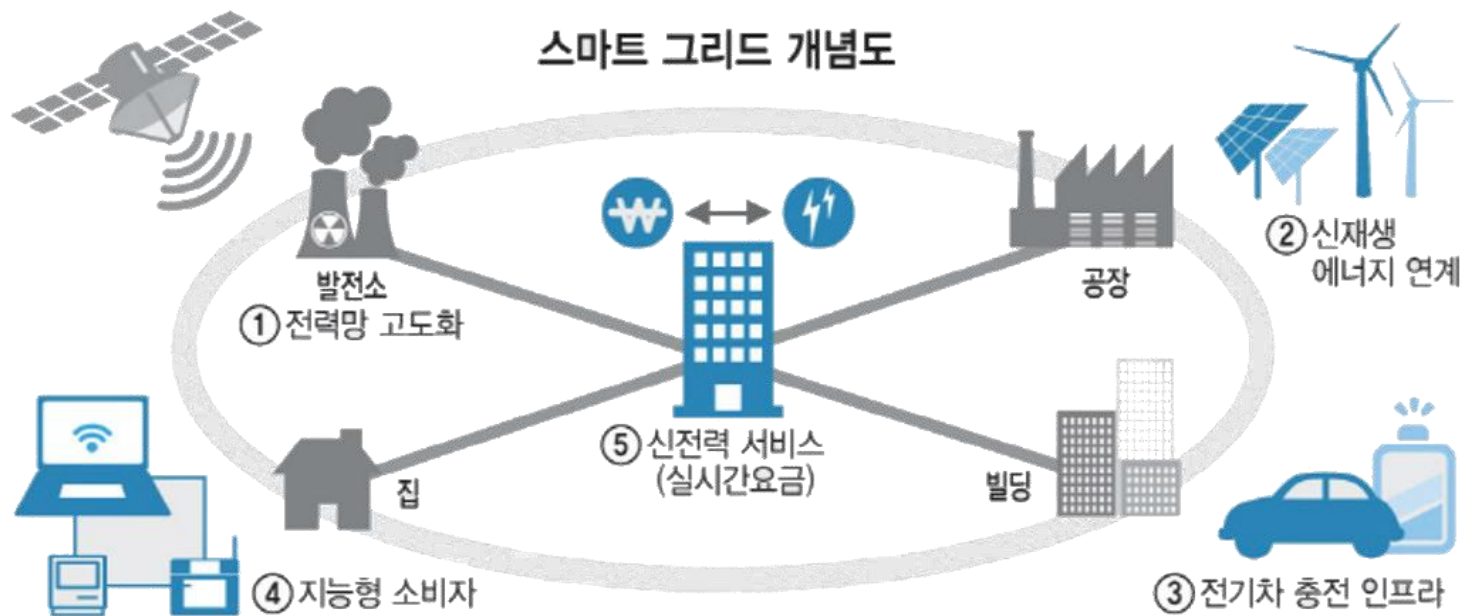
모든 데이터가 처음부터 분석에 맞는 형태로 존재하거나 기록되지 않기에 쌓여가는 데이터의 저장 및 처리 기술이 매우 중요



원천 데이터의 수집부터 분석까지  
“빅데이터 솔루션”



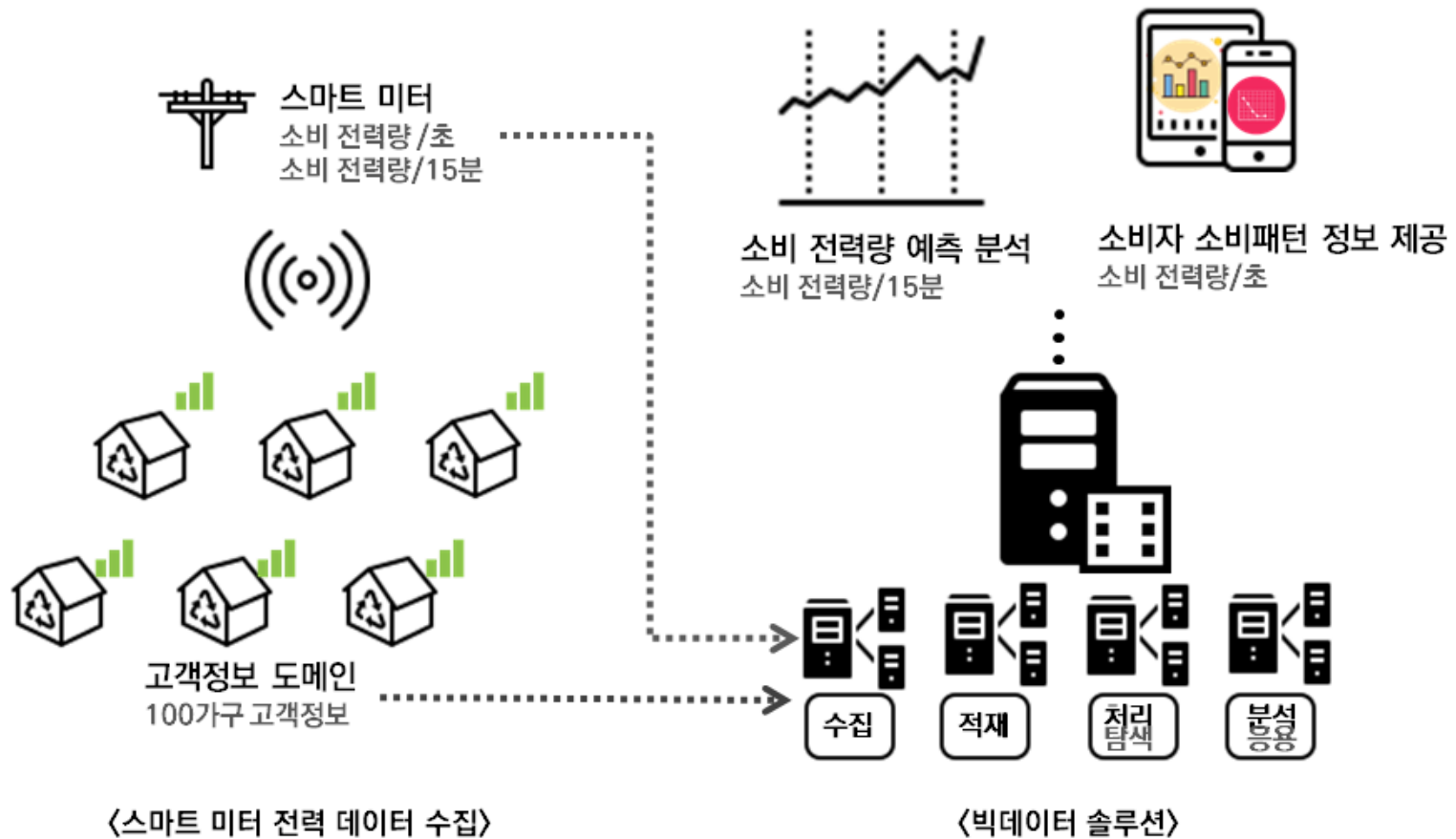
## 프로젝트 배경



- 기존의 전력시스템 - 일방향 전송 = 15% 예비전력 확보
- 스마트 그리드 - 공급자 ↔ 수요자 양방향 정보
- 정확한 수요파악은 효율적 전력생산이 가능

※ 가정/주택 스마트 미터 -> 전력 에너지 빅데이터 수집 - 빅데이터 솔루션 적용

# 프로젝트 도메인



- 스마트 미터 전력데이터는 전력사용 로그 시뮬레이터 활용
- 생성된 로그데이터는 빅데이터 솔루션 처리 과정 수행
- 데이터는 DW(Data Warehouse), DM(Data Mart) 가공 및 정제
- 정제된 데이터는 데이터 형태에 따라 사용자 정보제공 및 예측분석에 활용

# 프로젝트 도메인

## 전력사용량 로그데이터 발생주기 : 1초

- 100세대 가정용 스마트 미터  
정보 수집

- 스마트 미터는 1초 실시간  
전력 사용량 생성

- 데이터 발생 일시, 고객번호,  
mac address 생성

1초 단위 실시간 전력 사용량 데  
이터는 데이터는 일정 구간(시간)  
안 이상치를 필터링하여 소비자에  
게 정보 제공을 위한 데이터 셋

요구사항	각 세대 스마트미터 발생 로그 파일 수집 후 전력사용량 상태 점검(실시간)	
데이터의 종류	1인~9인 가구 100세대 전력량	
데이터 발생 주기	1초	
데이터 수집 주기	실시간	
데이터 수집 규모	100세대/초 (1일 수집규모 : 약 1GB)	
데이터 타입	텍스트(UTF-8), 반정형	
데이터 분석 주기	분/시간/일/주	
데이터 처리 유형	실시간	
데이터 구분자	coma(,)	
데이터 스키마	발생 일시	20191014045403(2019년 10월 14일 4시 54분 03초)
	고객 번호	H000~H100
	전력(kw)	실시간 전력사용량/1초
	mac address	00:00:00:00
예시	20190101000001, H001, 0.001162, a1:b1:c1:d1 정보 : 2019년 1월 1일 00시 00분 1초에 고객정보 H001의 전력량 0.001162발생 고유 주소는 a1:b1:c1:d1	

# 프로젝트 도메인

## 전력사용량 로그데이터 발생주기 : 15분

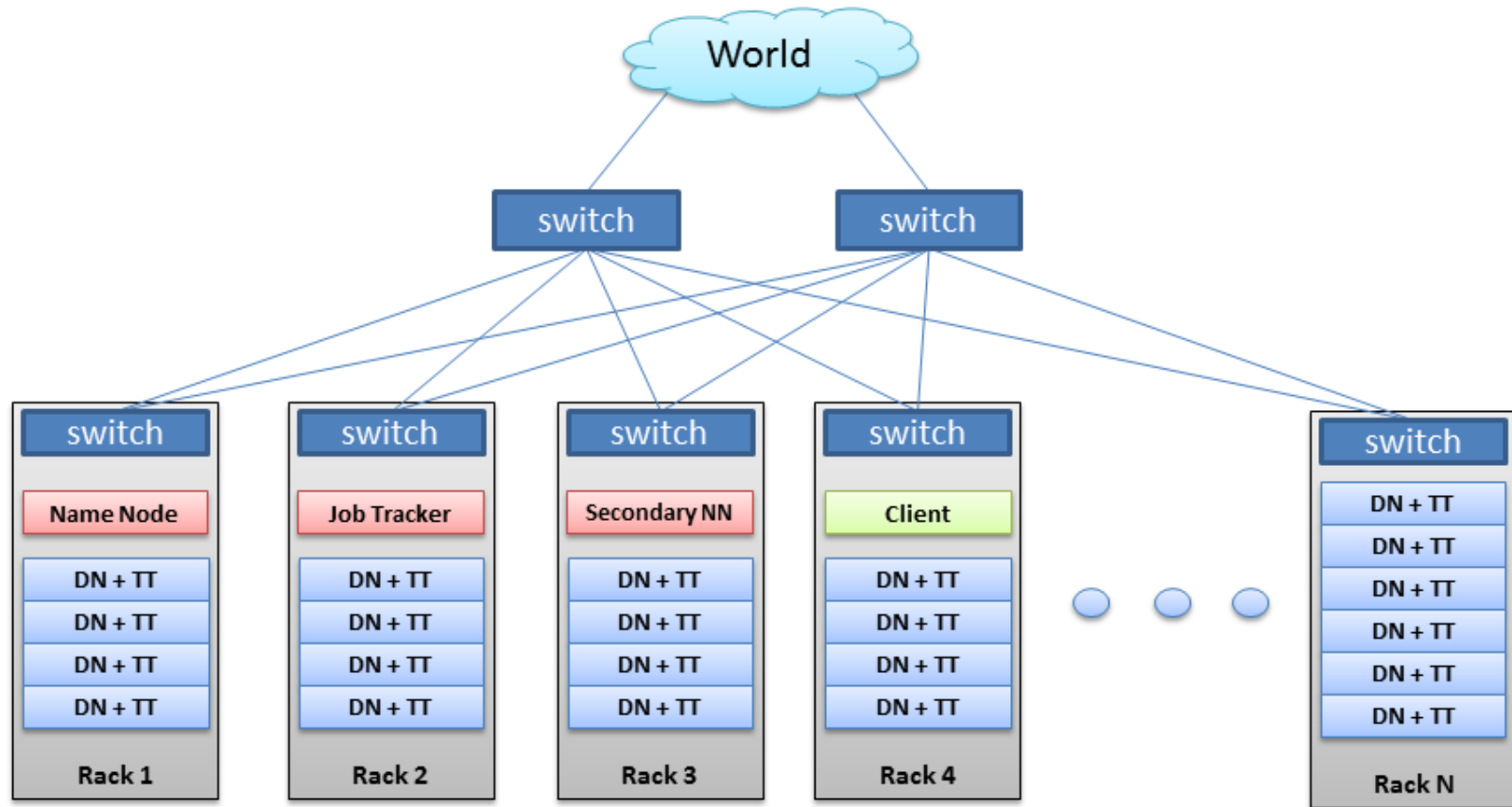
- 100세대 가정용 스마트 미터  
정보 수집
- 스마트 미터는 15분 주기 누적  
전력 사용량 생성
- 데이터 발생 일시, 고객번호,  
mac address 생성

15분 단위 누적 전력 사용량 데이터는 데이터 수집 후 솔루션 저장되어 이후 데이터 분석을 위한 데이터 셋

요구사항	각 세대 스마트미터 발생 로그 파일 수집 후 전력사용량 상태 점검(15분)	
데이터의 종류	1인~9인 가구 100세대 전력량	
데이터 발생 주기	15분	
데이터 수집 주기	15분	
데이터 수집 규모	100세대	
데이터 타입	텍스트(UTF-8), 반정형	
데이터 분석 주기	시간/일/주/월/년	
데이터 처리 유형	배치	
데이터 구분자	콤마(,)	
데이터 스키마	발생 일시	201901011215(2019년 1월 1일 12시 15분)
	고객 번호	H000~H100
	전력(kw)	전력사용량/15분
	mac address	00:00:00:00
예시	20190101000001, H001, 0.1425, a1:b1:c1:d1 정보 : 2019년 1월 1일 00시 00분 1초에 고객정보 H001의 전력량 0.1425발생 고유 주소는 a1:b1:c1:d1	

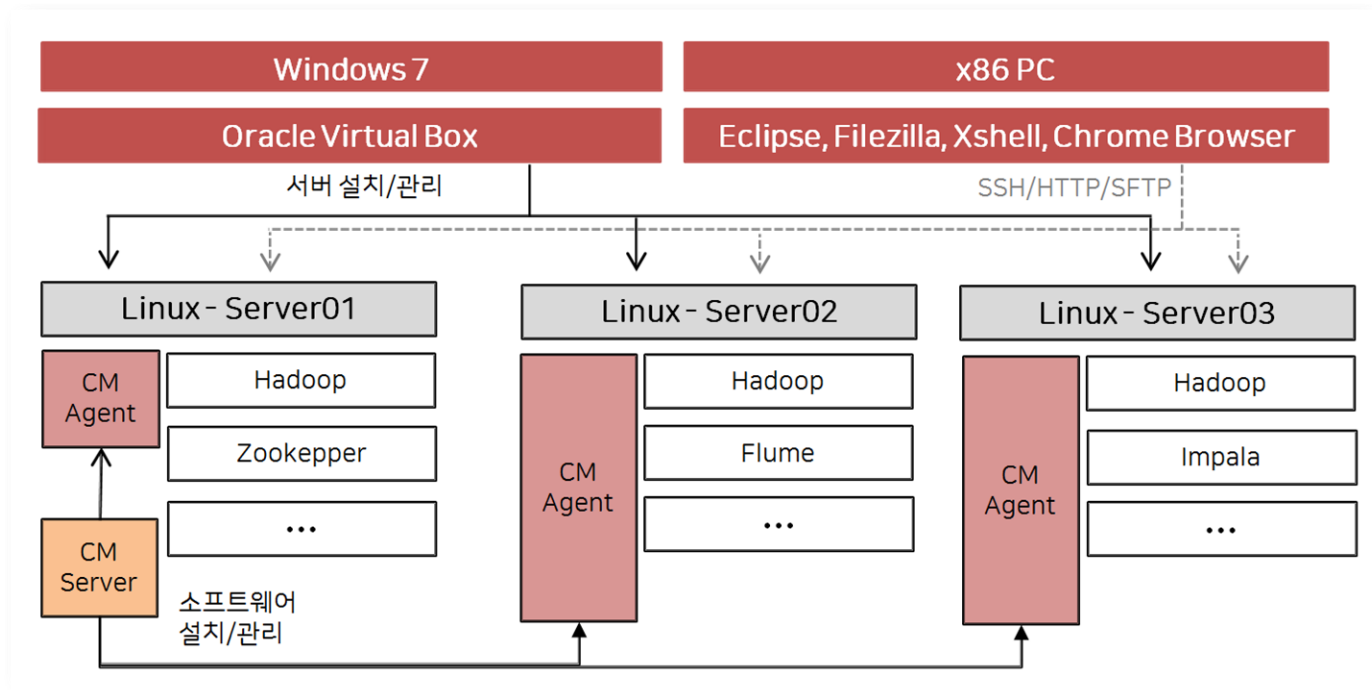


# Hadoop Cluster



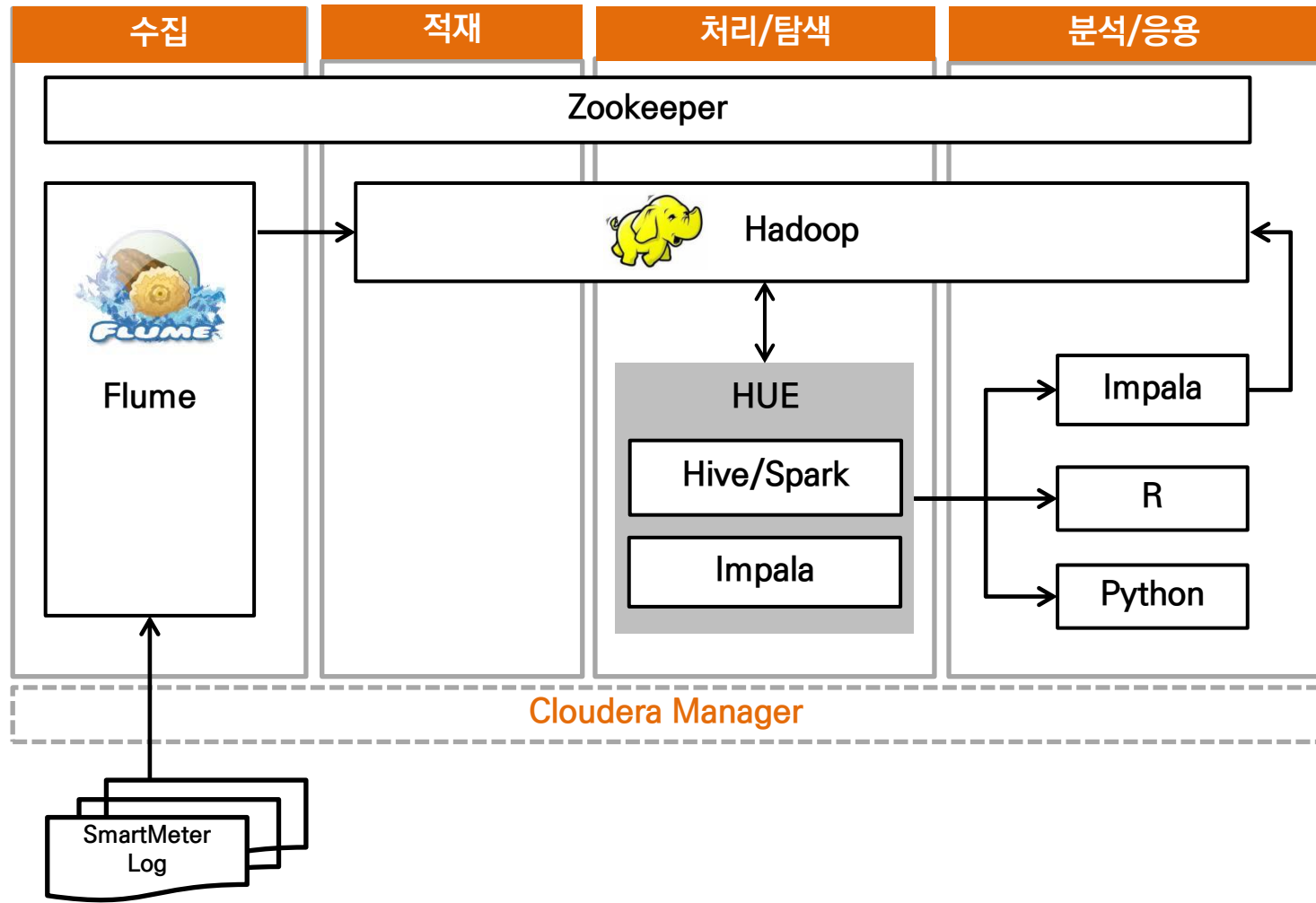
출처 : BRAD HEDLUND .com

# 시스템 아키텍처



- SW/HW 아키텍처를 구축/구현을 위해 3대의 Linux VM에서 진행
- 가상 환경이지만 빅데이터 모든 기술 요소를 완벽히 갖춘 시스템으로 수작업으로 설치와 설정을 하면 많은 시간과 노력이 필요
- 빅데이터 자동화 툴인 클라우드라의 Cloudera Manage(CM)을 이용해 소프트웨어를 설치/관리(모니터링)합니다.
- 개발 및 배포환경에서는 Eclipse, SFTP, SSH를 사용하며 특히, CM 웹 관리환경에서는 반드시 크롬 브라우저 사용

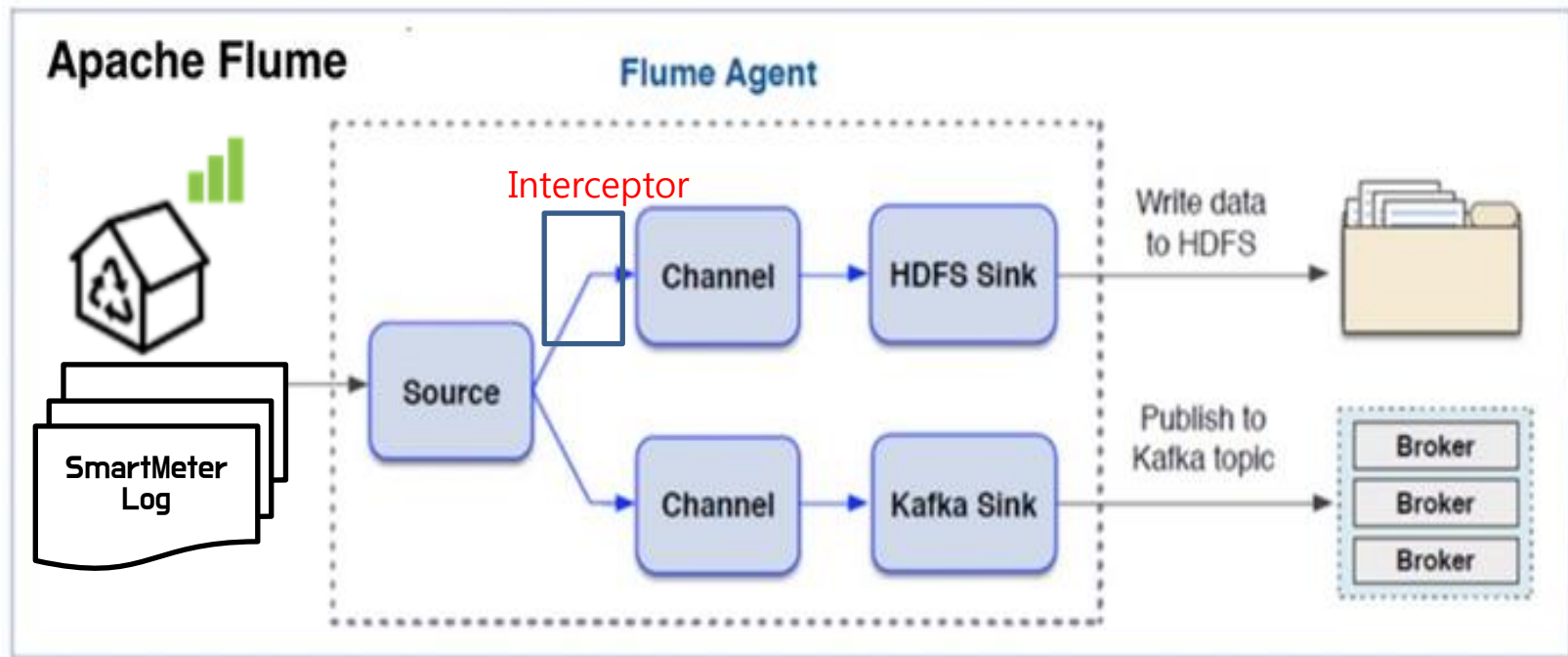
# 솔루션 아키텍처



# 솔루션 아키텍처 - 수집 레이어

원천 데이터

적재 레이어



# 솔루션 아키텍처 - 플럼 메커니즘

```
SmartMeter.sources = HourSource      RealTimeSource
SmartMeter.channels = HourChannel     RealTimeChannel
SmartMeter.sinks    = Hour_hdfsSink   RealTimeSink
```

```
SmartMeter.sources.HourSource.type = spoolDir
SmartMeter.sources.HourSource.spoolDir = /home/workspace/smartmeter/working/flume_spool
SmartMeter.sources.HourSource.deletePolicy = immediate
SmartMeter.sources.HourSource.batchSize = 1000
SmartMeter.sources.HourSource.interceptors = timeInterceptor typeInterceptor collectDayInterceptor
filterInterceptor
SmartMeter.sources.HourSource.interceptors.timeInterceptor.type = timestamp
SmartMeter.sources.HourSource.interceptors.timeInterceptor.preserveExisting = true
SmartMeter.sources.HourSource.interceptors.typeInterceptor.type = static
SmartMeter.sources.HourSource.interceptors.typeInterceptor.key = logType
SmartMeter.sources.HourSource.interceptors.typeInterceptor.value = flume_spool
SmartMeter.sources.HourSource.interceptors.collectDayInterceptor.type = bigdata.CollectDayInterceptor
$Builder
SmartMeter.sources.HourSource.interceptors.filterInterceptor.type = regex_filter
SmartMeter.sources.HourSource.interceptors.filterInterceptor.regex = ^\\d{14}
SmartMeter.sources.HourSource.interceptors.filterInterceptor.excludeEvents = false
```

```
SmartMeter.sources.HourSource.type = spoolDir
SmartMeter.sources.HourSource.spoolDir = /home/workspace/smartmeter/working/flume_spool
SmartMeter.sources.HourSource.deletePolicy = immediate
SmartMeter.sources.HourSource.batchSize = 1000
```

Source

```
SmartMeter.sources.HourSource.interceptors.filterInterceptor.type = regex_filter
SmartMeter.sources.HourSource.interceptors.filterInterceptor.regex = ^\\d{14}
```

Channel

```
SmartMeter.sinks.RealTimeSink.type = hdfs
SmartMeter.sinks.RealTimeSink.hdfs.path = /workspace/smartmeter/collect/$(logType)/real_meter=%Y%m%d%h
SmartMeter.sinks.RealTimeSink.hdfs.filePrefix = $(logType)
SmartMeter.sinks.RealTimeSink.hdfs.fileSuffix = .log
SmartMeter.sinks.RealTimeSink.hdfs.fileType = DataStream
SmartMeter.sinks.RealTimeSink.hdfs.writeFormat = Text
SmartMeter.sinks.RealTimeSink.hdfs.batchSize = 10000
SmartMeter.sinks.RealTimeSink.hdfs.rollInterval = 0
SmartMeter.sinks.RealTimeSink.hdfs.rollCount = 0
SmartMeter.sinks.RealTimeSink.hdfs.idleTimeout = 100
SmartMeter.sinks.RealTimeSink.hdfs.callTimeout = 600000
SmartMeter.sinks.RealTimeSink.hdfs.rollSize = 67108864
SmartMeter.sinks.RealTimeSink.hdfs.threadsPoolSize = 10
```

```
SmartMeter.sink.Hour.hdfsSink.type=hdfs
SmartMeter.sink.Hour.hdfsSink.hdfs.path = /workspace/smartmeter/collect/$(logType)/w가_date=%Y%m%d
```

Sink

## 솔루션 아키텍처 - 플럼 실행

“스마트미터 100대에서 생성되는 정보를 발생과 동시에 플럼 에이전트가 수집해서 하둡에 전송한다.”

[illegible]

# 솔루션 아키텍처 - 적재 레이어



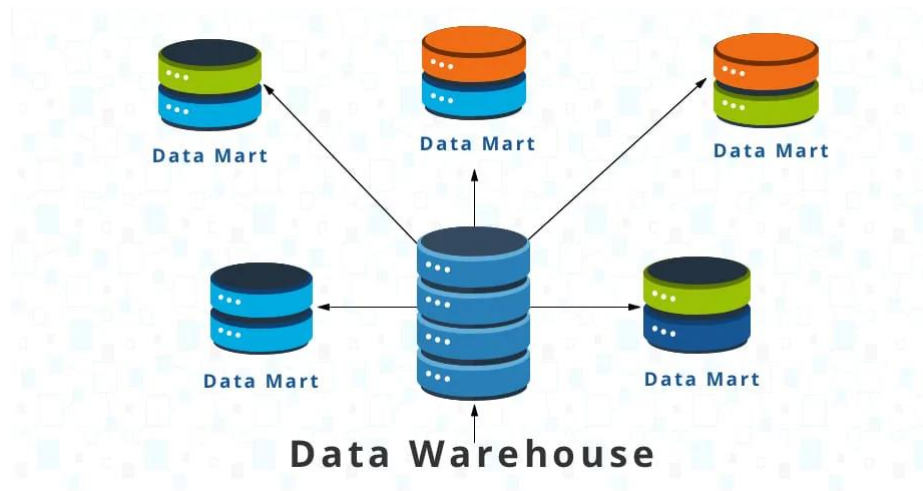
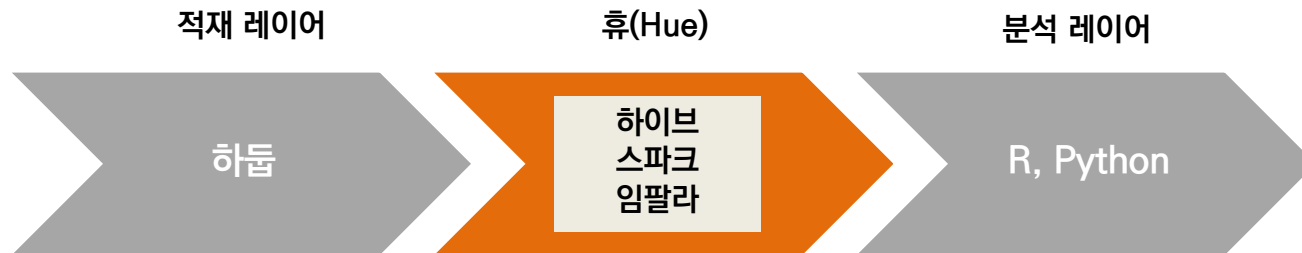
- 스마트 미터에서 발생하는 정보 로그가 비교적 큰 파일이기 때문에 HDFS의 특정 디렉터리에 일자 단위로 파티션해서 적재한다.
- 장기간 적재된 데이터는 최종적으로 일/주/월/년별로 스마트 미터의 다양한 시계열 집계 분석을 할 수 있다.

## ☑ 분산 병렬 처리 시스템

🏠 홈 / workspace / smartmeter / collect / flume\_spool / wrk\_date=20191101

<input type="checkbox"/>	이름	크기	사용자	그룹	권한	날짜
<input type="checkbox"/>	<b>↑</b>		flume	supergroup	drwxr-xr-x	October 31, 2019 07:38 PM
<input type="checkbox"/>	<b>.</b>		flume	supergroup	drwxr-xr-x	October 31, 2019 07:42 PM
<input type="checkbox"/>	<a href="#">flume_spool.1572575918363.log</a>	65.0 MB	flume	supergroup	-rw-r--r--	October 31, 2019 07:39 PM
<input type="checkbox"/>	<a href="#">flume_spool.1572575918364.log</a>	65.0 MB	flume	supergroup	-rw-r--r--	October 31, 2019 07:39 PM
<input type="checkbox"/>	<a href="#">flume_spool.1572575918365.log</a>	65.0 MB	flume	supergroup	-rw-r--r--	October 31, 2019 07:40 PM
<input type="checkbox"/>	<a href="#">flume_spool.1572575918366.log</a>	65.0 MB	flume	supergroup	-rw-r--r--	October 31, 2019 07:40 PM
<input type="checkbox"/>	<a href="#">flume_spool.1572575918367.log</a>	24.8 MB	flume	supergroup	-rw-r--r--	October 31, 2019 07:42 PM

# 솔루션 아키텍처 - 처리/탐색 레이어



- ☑ 덩치 큰 비정형 데이터를 정교한 필터링, 클린징, 통합, 분리 등으로 정형화하는 작업
- ☑ 업무 도메인에 대한 이해를 바탕으로 탐색적 분석 필요



# 솔루션 아키텍처 - 처리/탐색 레이어

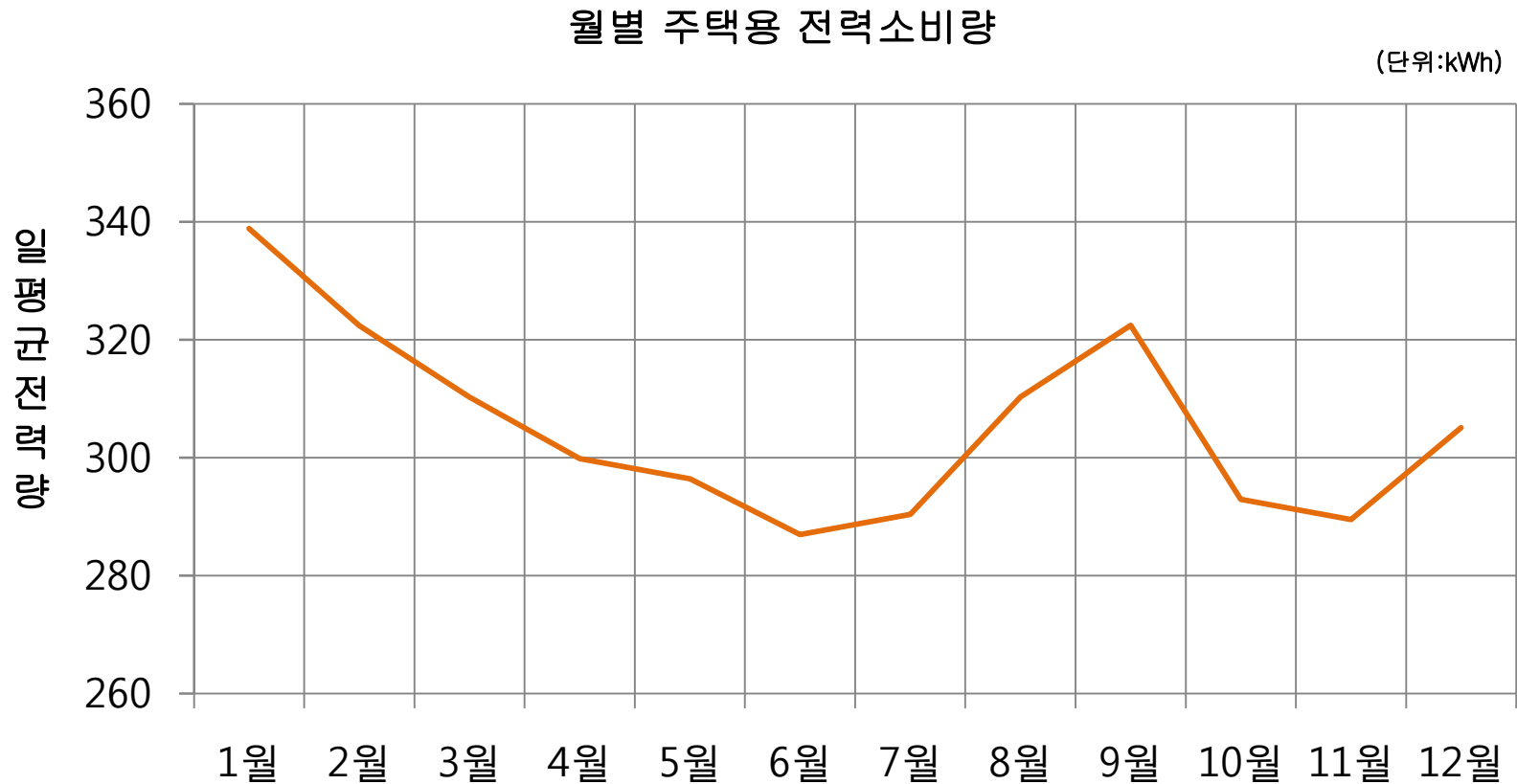
```
1 select user_id , left(date_time,4) as A,  
2 avg(elec) from smartmeter_info  
3 group by user_id,A order by user_id,A ;
```

	user_id	m	avg(elec)
1	H001	01	174.63143408273177
2	H001	02	166.52056584726077
3	H001	03	159.79380409656562
4	H001	04	154.38403856363166
5	H001	05	152.72014479165082
6	H001	06	147.80297936147534
7	H001	07	149.58178633208558
8	H001	08	160.04281604946991
9	H001	09	166.04517120871549
10	H001	10	149.43139226094664
11	H001	11	130.33318787660355
12	H001	12	137.46386496985224
13	H001	13	131.4838998882554
14	H001	14	130.33318181990322
15	H001	15	148.43138559084994
16	H001	16	160.04281115081198



# 솔루션 아키텍처 - 분석/응용 레이어

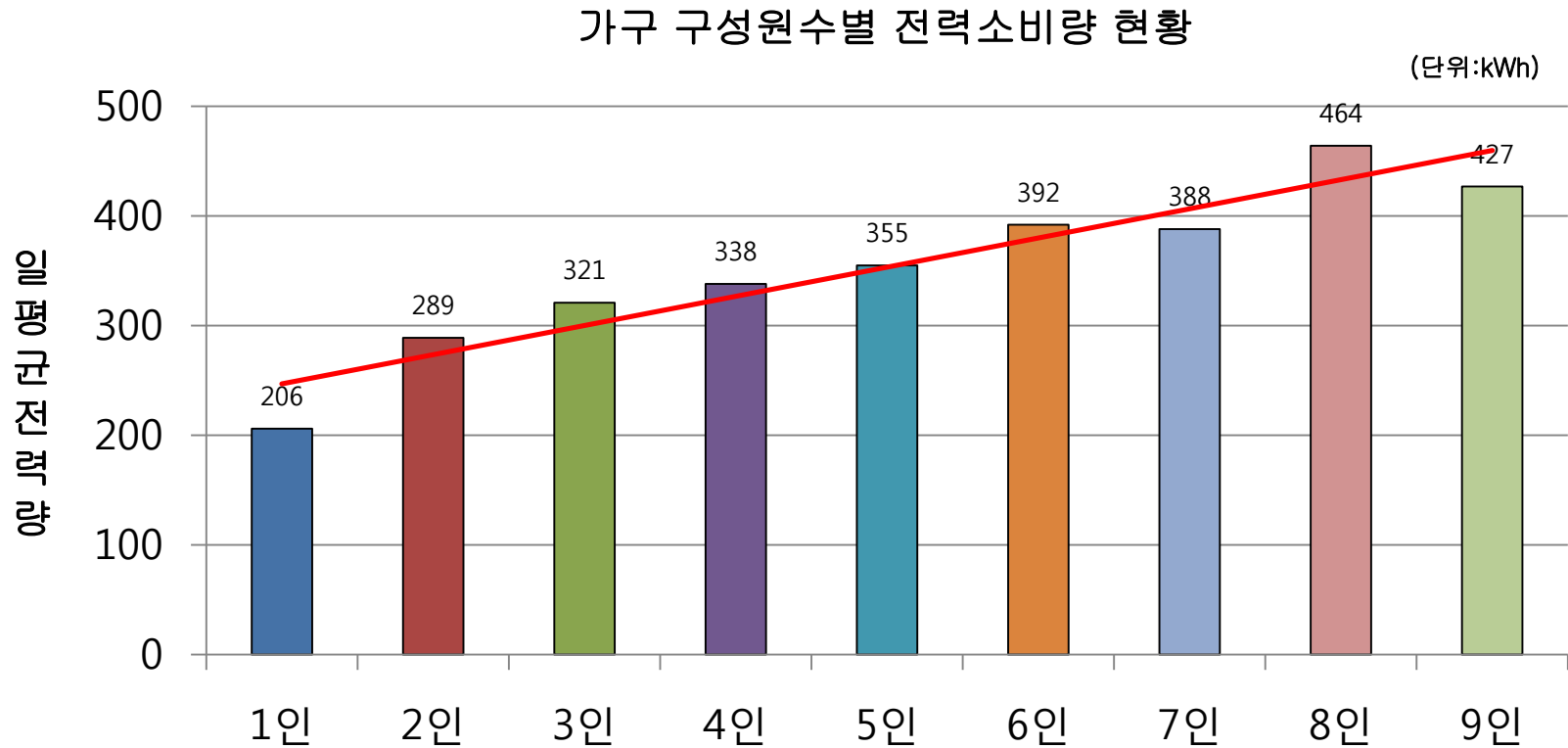
## 주택용 전력소비량 현황



> 동절기, 하절기에 전력소비량 증가

# 솔루션 아키텍처 - 분석/응용 레이어

## 주택용 전력소비량 현황



> 가구 구성원수가 증가할수록 전력소비량 증가

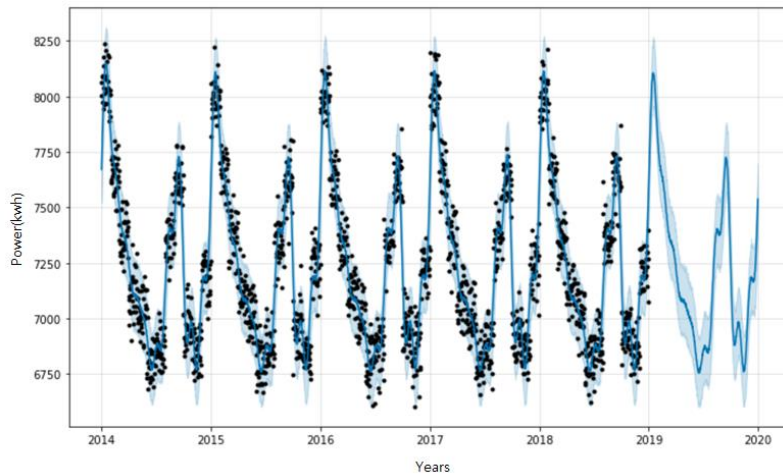
# 솔루션 아키텍처 - 분석/응용 레이어

## 수요예측

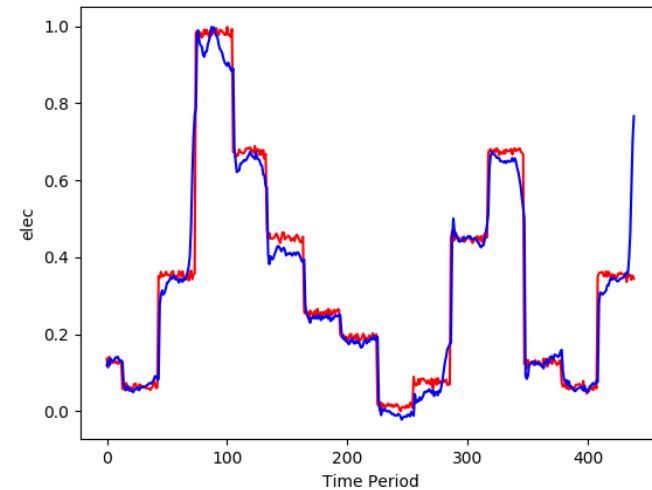
### DATASET

date	평균기온(°C)	최저기온(°C)	최고기온(°C)	평균 이슬점온도(°C)	평균 현지기압(hPa)	평균 해면기압(hPa)	합계 일사량(MJ/m2)	평균 지면온도(°C)	elec
20140101	5.1	0.4	9.7	-2.4	1007.4	1015.9	9.79	1.8	847701.4256
20140102	2.6	-2.2	9.1	-6.1	1013.1	1021.7	11.01	1.3	847868.9795
20140103	2.1	-3.4	9.6	-5.5	1009.3	1018	8.42	0.7	848362.2574
20140104	1	-2.7	6.5	-6.2	1011	1019.7	11.58	0.6	847253.1833
20140105	-0.8	-5.9	5.5	-6.3	1015.5	1024.3	10.97	-1.3	848006.2568

FB prophet



LSTM



### 1. 대용량 데이터를 수집, 적재, 탐색(처리), 분석(응용)

15분 단위 전력 생성 데이터 =

$100\text{가구} * 15\text{분주기} * 5\text{년} = 17,280,000\text{건}$  데이터 처리

1초 단위 전력 생성 데이터 =

$100\text{가구} * 1\text{초 주기} * 1\text{주일} = 60,480,000\text{건}$  데이터 처리

### 2. 대용량 실시간 로그를 처리에 특화된 카프카 구현 불가능

RabbitMQ나 ActiveMQ는 효율적이며 구성하기 쉽지만 대용량을 처리하는것에는 Kafka가 더 뛰어남

플럼에서 바로 하둡으로 적재하는 방법도 있지만 하둡에 장애가 발생하면 데이터 유실이라는 치명적 문제 발생

Q&A

**감사합니다.**