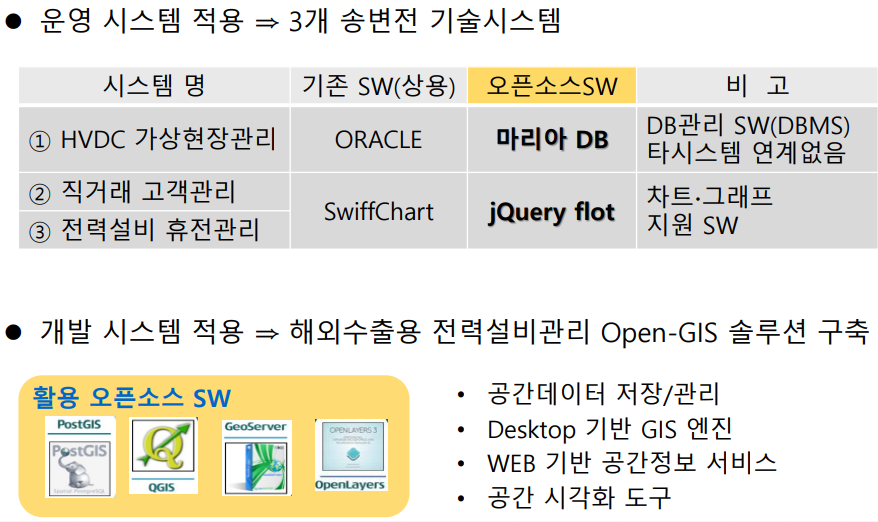
<4차산업혁명을 대비한 전력산업분야 오픈소스 도입 및 활용사례> 한국전력공사 김종명 부장

하드웨어에서 소프트웨어 중심으로 진화 => 플랫폼 중심 체제로 변환

한전 정보시스템은 Oracle 라이선스 구입비용 150억, 년간 20억 유지보수 소비

* 오라클은 법무팀이 2/3, 심심하면 찾아오는 라이선스 규칙 위배 소송, 추가 비용 요구
* 오픈소스를 활용하는 방안을 연구 중  
  
* Open-GIS의 경우 특정 브라우져에서만 구동 / active X 기반  
  🡺 다수 브라우저 지원, 오픈소스를 활용하여 추가 기능 구현  
  🡺 나아가 GIS S/W 플랫폼을 개발 후 상표 등록 ( 지리정보 기반으로 전력 설비 및 계통 운영을 관리하는 솔루션, 북한으로 길이 열렸을 때를 위한 대비라고 자랑 )
* 향후 한전이 생각하는 방향  
  정부 기관에서도 오픈소스 전담 조직이 필요하다.  
  경영진 및 정보시스템 조직내 부정적 인식 극복 필요  
  **오픈소스별 라이선스 정책 모호.다양.복잡(법적분쟁 주의)**

## 한전에서 진행하는 프로젝트 소개 발표 느낌  
정부기관의 특성상 경력직 대부분은 관리자 역할을 수행 => 개발 전담 인력이 부족하기 때문에 전담 조직을 만들더라고 오픈소스 도입에 어려움을 겪을 것이라 예상됨

<오픈소스에 의한 디지털 트랜스포메이션> 한국레드햇 황인찬 부사장

Digital Transformation : 기업의 중요한 화두 / CDO 책임자 필요 / 이미 도입한 기업도 있다.

기존에는?) DT를 단순히 기술적으로만 바라봄

* 하지만 기업의 조직/문화가 먼저 변화하고 이를 받아들일 준비가 되어 있어야 한다.
* OS는 모든 것을 공유하고 기여를 통해 만들어지는 문화. 해당 사상에 의한 조직이 필요하다.

OS는 기존의 틀을 파괴하고 혁신을 주도하는 역할을 수행하고 있다.

* 따로 떨어져 별개로 움직이는 구성원들을 어떻게 연결할 것인가?
* 소수의 전문가 X => 모든 구성원이 함께 참여하고 동등한 입장에서 의견을 내야함
* 기업 내 외부 아니라 고객, 협력사 그리고 경쟁사도 함께 참여해야 한다.

OS를 바탕으로 어떻게 데이터 트랜스포메이션을 할 것 인가?

* 공유, 공개, 참여 / 모든 프로세스와 커뮤니케이션 과정이 공개
* 모든 이슈를 표준화하고 공개 후에 서로 참여를 이끌어야함
* 즉, OS의 문화를 이해하고 이를 조직이 이를 반영해야 한다.

OS를 바라보는 시각

* TOP DOWN( 예산과 경영진의 파워가 필요) => 이는 곧 회사 의사결정 구조라고 할 수 있다.
* OS는 Bottom up 방식을 지향, start small, 일단 시작해서 성공 사례를 공유하고 변화의 바람이 불게하라.
* 레드헷이 이때까지 살아남을 수 있었던 이유는 조직 구성원 모두가 미래를 예측하기 위해 노력함, 어떤 기반의 회사를 인수 / 어떤 커뮤니티와 연계 => 어떻게 같이 성장 할 것인가?
* 의사결정의 병목지점이 없는 환경이 중요하다.

디지털 트렌스포메이션을 위해 기업 내 모든 요소를 디지털화 하는 기술적 접근보다

이를 받아들일 수 있는 문화를 가진 조직의 중요성을 강조한 발표.

<Sustainable business with Open Source> 카카오 공용준 상무

상용 SW에서 이제는 오픈소스가 모든 도메인에 표준 규격이 되고 있다.

* 현재의 상용 SW와 오픈소스간의 기술적 차이는 미비하지만 TCO(total cost of ownership)부분에서 오픈소스가 압도적으로 유리 -> 커뮤니티와 컨트리뷰션의 역할
* 각종 기업들이 합병을 하고 오픈소스로 이루어진 툴을 야금야금 유로로 전환하고 있음  
  ex) 클라우데라&호튼웍스 / 레드햇&IBM은 과연?
* 기업들이 원하는 방향으로 오픈소스를 개발하기 위해 해당 권한이 있는 컨트리뷰터 영입에 혈안.
* 이러한 환경속에서 어떻게 서비스를 지속할 수 있을까?

Automate Test, Deployment, Operation of OpenSource 가 필수적이다.

* 오픈소스를 도입해서 개발하기 시작하면 한명이 담당하는 영역이 너무 넓어짐.
* 많은 버그, 보안 이슈 해결, 시시각각 릴리즈 되는 오픈소스 버전 -> 위에 3가지 요소는 반드시 필요한 부분

현재 DI팀 내부적으로 test 환경이 존재하는가?   
내부적으로만 사용하는 지금 상황에서도 1명이 개발/운영을 담당, Test 부재 => 버전 업데이트, 보안상 크리티컬 한 이슈, 시시각각 발생하는 문제들…

추가QA ) 기존 고객이 상용 SW에서 벗어나지 못하는 이유.

* 초등학교 5학년 동화에 비유. 이미 상용 SW에 길들여져 있다. 벗어나기 힘든 상황
* 하지만 시시각각 변하는 시장/고객의 요구사항을 바탕으로 개발되기에는 OS가 유지
* OS를 바탕으로 한 서비스 개발에 그치지 않고 해당 문화를 전파해야 함

<Serverless로 modern application 만들기> - 빙글 이상현

Serverless와 container는 분명히 다르다. (서로 간의 디스전)

Serverless => 개발자가 Business Logic 만 만들면 되는 미래.

* 운영, 모니터링, 유지보수 모든 것이 없는
* 이미 해외에서는 다른 개발 트렌드에 비해 서버리스가 상당히 화두
* 하지만 한국에서는 새로운(Library, Framework, concept, Case) => 터지면 누가 책임?

빙글은 모든 것이 서버리스로 구성

* 수동으로 서버 스케일링 하는게 전혀 없음
* 트래픽 나오는 만큼만 돈을 지불
* 중고장터 앱을 만든다고 했을 때 필요한 요소들을 AWS에서 어떻게 활용하면 되는지 소개

<넷플릭스의 플랫폼 엔지니어링> - 피보탈 정윤진

피보탈 : 웹의 모든 오픈소스를 주도적으로 개발하고 있는 회사 (tomcat, spring 등)

넷플릭스

* 마이크로서비스 / 클라우드 네이티브의 발원지
* 한달 40억 시간 스트리밍 / 사용자 1억 4천만 ( 14,000 )

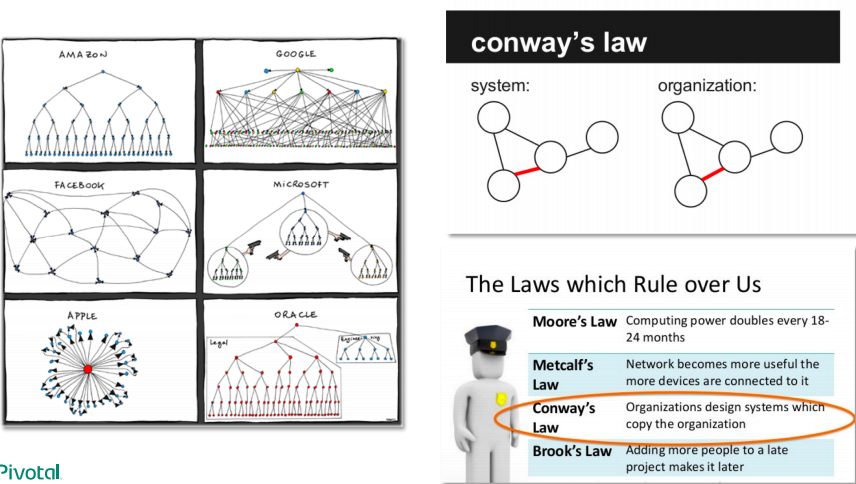
기존에는 Big java <-> Big Oracle으로 구현 / 하지만 디비가 일부 깨지면서 전체 시스템 다운

* 그 후 약 7년이라는 시간을 들여서 현재의 넷플릭스가 탄생
* 모든 구조를 각각 정의하고 4~5명이 한 팀을 이루어서 마이크로서비스를 담당
* 마이크로서비스 각각의 기술들을 youtube에 참고 / 최근 전체 구성을 공개함

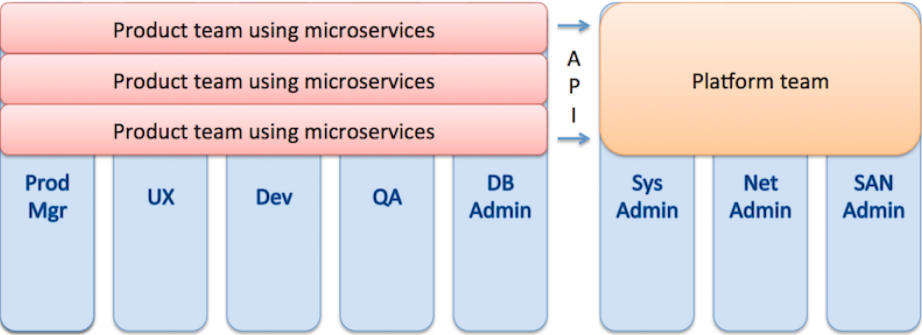
AWS의 경우 약 11초마다 백단에서 업데이트가 일어나고 있음.

* 한국에서 zipcode 수정은? => 기술과 사업경쟁력에서 차이
* [서비스 + DB] / [서비스 + DB] 각각을 별개로 생각하고 구조를 작성해야 한다.  
  마이크로서비스에 적합한 DB를 사용

Conway’s Law. 소프트웨어 구조는 조직의 커뮤니케이션 구조를 닮는다.

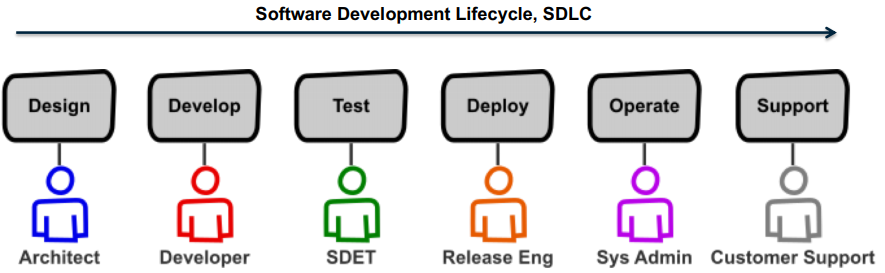


넷플릭스에서는 각 부서별로 나눠져 있던 업무를 마이크로서비스 단위로 쪼개버림.  
각각의 마이크로서비스를 API로 제공해서 플랫폼 팀이 운영만하는 상황  
중요한 화두는 플랫폼으로 전환   
풀스택 개발자에서 Full Cycle Developers라는 새로운 개념이 탄생



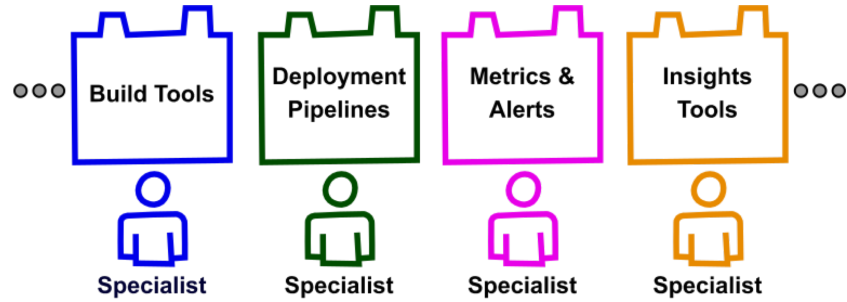
Full Cycle Developers?

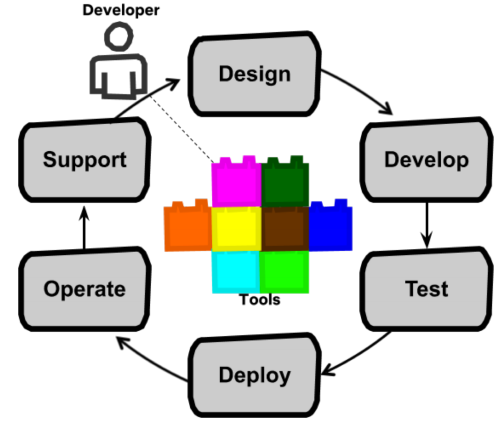
기존에는 업무별 전문가들을 모아둔 팀 단위  
하지만 서비스는 한 개로 공유 => 사일로 현상이 발생  
서비스의 이익 < 조직의 이익



넷플릭스에서는?

마이크로서비스 하나당 4~5명의 전문가로 구성 / 개발, 운영, 배포, 모니터링 모두 담당



  
넷플릭스가 마이크로서비스를 지향하면서 강조한 솔루션들 소개

* Service Auto Healing -> Chaos Engineering  
  재해복구 시스템이 정상적으로 작동할 것 이라고 확인할 수 있는가?  
  방치하다가 터지는 것 보다는 우리 손으로 죽여보는게 더 좋다.   
  랜덤으로 마이크로서비스를 죽이고 장애 대응, 하지만 모든 운영에는 지장이 없음

DI팀에서 활용하고 있는 서버들이 만약에… 혹시라도… 진짜로… 죽어버리면 복구가 가능할까?  
백업 주기 선정 필요 / 한번쯤은 재해복구 상황을 가정하고 시뮬레이션도 필요할 듯

<대용량 서비스 설계 방법> - 강대명

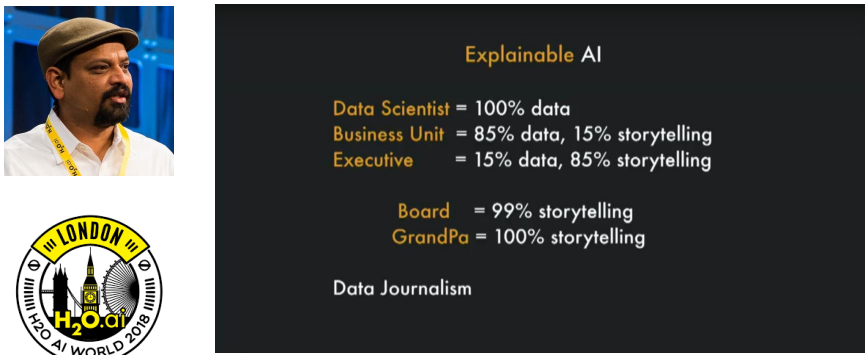
다양한 상황을 가정하고 거기에 따른 해결방안 소개

해결 -> 문제 발생 -> 해결 ….

DI팀에서 생각해봤을 때 데이터 처리

* 지금은 하나의 데이터베이스로 유지가 가능하다. (필요할 때 마다 생성, 별도의 정의 없음)
* 만약 서비스를 위한 데이터를 모은다고 했을 때? (광고 성과 분석 / Search base 등)   
  언젠가는 하나의 DB로는 보관이 불가능하며 데이터를 분산 해야하는 상황이 올 수도 있다.  
  기술적으로 고민을 하기전에 **데이터를 어떻게 나눌 것인가?**🡺 **데이터 샤딩**에 대한 정의가 필요할 것 / 데이터를 어떻게 나누고 찾을 것인가?

<기업의 빅데이터 분석에서 서비스까지 성공 실패 이야기> - 김영균

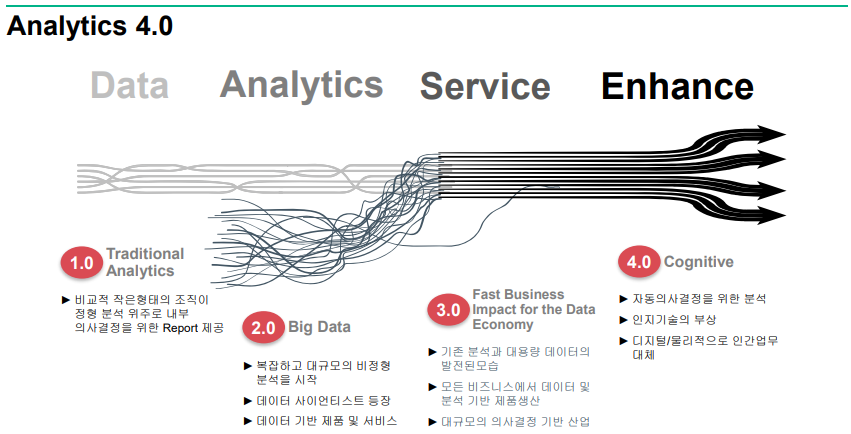


빅데이터를 운영 / 서비스를 하기 위해서 데이터를 모으고 AI를 적용해보고자 하는 방향성들

하지만 적용했을 때 활용은 어떻게 할 것인가에 대한 체계가 미흡한게 국내 상황 (2.5 단계)

현재 아티언스의 현황은 1.0? 다음 단계로 넘어가기 위해서 준비해야할 것들은 무엇이 있을까

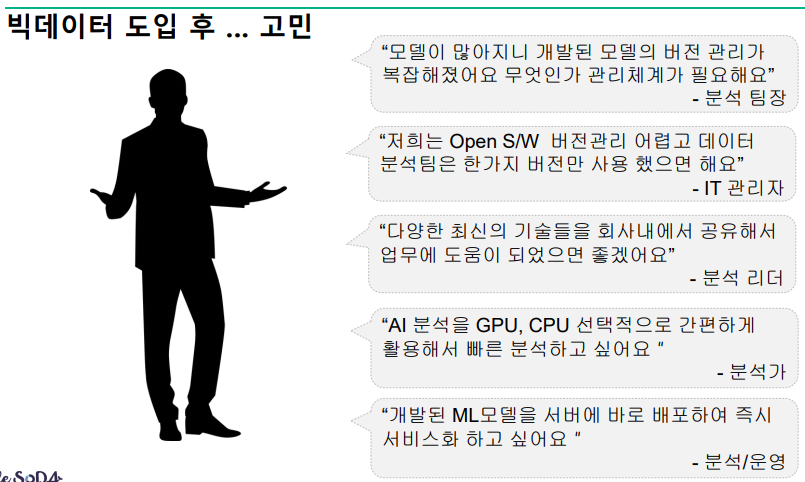
아티언스가 잘 할 수 있는 것에 대한 고민 (SEM / SEO 분석?)



컨설팅 사례 소개 ( Anomaly Detection)

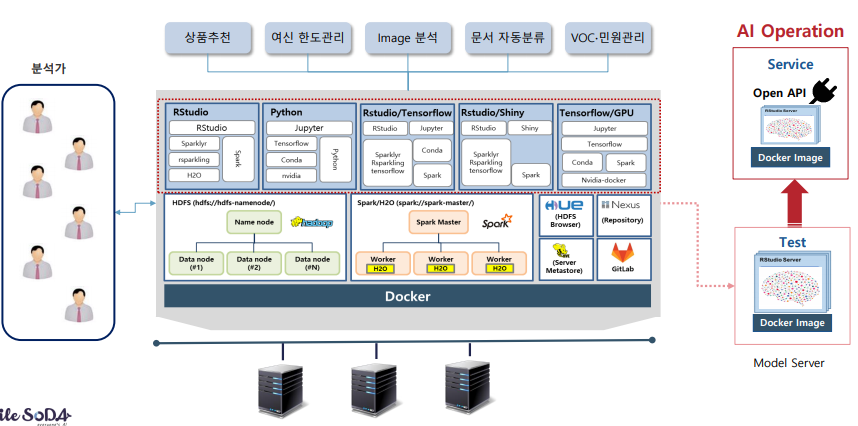
시나리오 vs 모델

* 모델이 정확도는 더 높은데 근거는? (설명 불가능)
* 프레임워크화가 되어있지 않기에 컨설팅에서 끝남. 도움이 없다면 추가 분석이 불가능하다.  
  기업들의 자생 능력이 부족 (이를 활용한 운영 컨설팅이 중요 포인트 ex. Uplus 운영 보고서)
* 빅데이터를 활용한 모델을 도입하더라도 추가적인 문제들이 발생



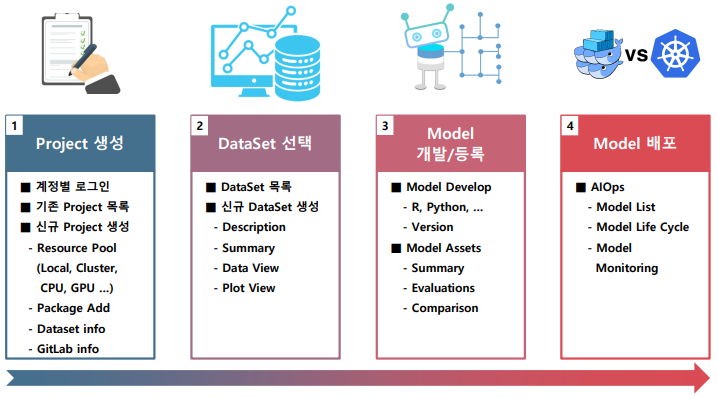
데브옵스의 출현

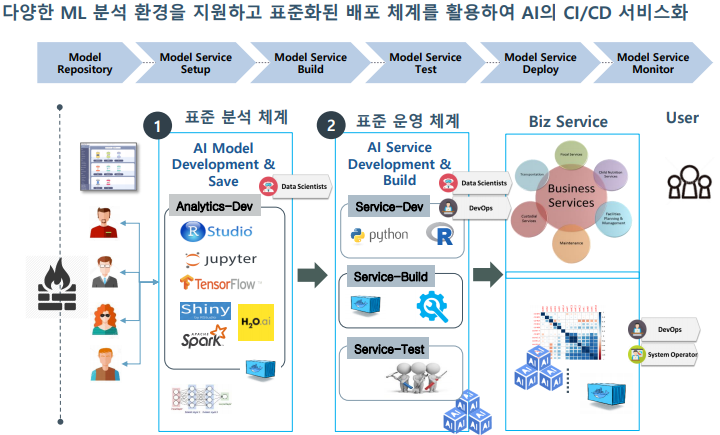
* 소프트웨어 제품과 서비스를 빠른 시간에 개발 및 배포하는 것을 목적
* 애자일소다는 AI DevOps를 추구하는 회사 (SparklingSoDA Architecture)



* 데이터를 저장하는 방식 (Hadoop / Spark)
* 분석하는 환경(R, Python, Tensorflow …)
* 운영/배포/테스트/서비스환경

모든 것을 사용자가 편의로 선택가능 (데이터셋 / 모델링 환경 / 운영배포 환경)





**추구하는 서비스에 대한 Life Cycle 정의가 필요 /**

Ex) 빅데이터 구축 -> 모델링 -> 배포 즉 모든 개발은 담당자가 존재

실제 운영하는 사람들은 데이터만 바라보고 있다. 이 데이터가 어떻게 발생하며 내 눈에 보이는지 전혀 모르는 상황.

담당자가 퇴사하면 ? 기술은 어떻게 전파할 것? 기존에 세팅 로그는?   
서비스에 대한 체계가 준비되어 있는가에 대한 고민이 필요할 듯

<기술의 발전, 그리고 기술 커뮤니티의 발전 부제: Apache Spark과 스사모(스파크 사용자 모임)>

현재로써 대용량 데이터 처리 시, Spark을 대체할 만한 대체제는 별로 없음

머신러닝, 딥러닝 연구 및 적용시에도 대용량 데이터 처리는 단연 Apache Spark

<분산 스트리밍 플랫폼 - Apache Kafka 소개 부제 : Kafka & Kafka KRU>

스트리밍 데이터?

* 수천개의 데이터 소스에서 실시간으로 생성되는 데이터 / 작은 크기(KB)로 동시에 전송
* 여행 좌표 데이터 / 은행 거래 / 전자 상거래 구매 / 게임 플레이어 활동 / 통신 등등
* 이를 위한 최고의 플랫폼
* 대부분 빅데이터 서비스는 카프카 + 스파크 조합으로 구현되어 있음

<Machine Learning Algorithm and pipeline with kaggle competitions>

다양한 머신러닝 / 딥러닝 방법론 소개

* CTR 분석하는 캐글 대회도 있다던데…?

**DI팀 내부 데이터 처리 업무로의 도입 검토**

아티언스에서 데이터를 활용하는 현황 : Analytics 1.0 수준  
데이터를 활용한 서비스를 도입하기 위해 준비해야 할 것들은 무엇이 있을까?  
 \* 백업 주기 / 시뮬레이션  
 \* 서비스 도입 이전에 test에 대한 부분도 고민할 필요성이 존재   
 \* 향후 데이터 보관을 위한 데이터 샤딩에 대한 정의도 언젠가는 필요(분산 처리가 필요할 때)  
 \* 오픈소스를 활용한 서비스 진행 시 라이선스에 대한 검토가 반드시 필요(NIPA를 통한 상담)

**DI팀 태깅/분석 서비스의 현황/문제점 및 개선 아이디어 검토**

Uplus GA 통합 프로젝트  
 \* 컨설팅에서 끝나지 않고 운영 보고서를 제안한 상황  
 \* 한 단계 성숙한 분석 서비스를 위한 준비의 필요성 (GA 툴 활용 / 데이터를 바탕으로 한 분석)

GTM 태깅 서비스의 현황  
 \* 아티언스 <-> 이노션 <-> 매체  
 \* 실제 운영하는 사람들은 데이터만 보고 있음  
 \* 매체에서 데이터가 어떻게 발생하는지를 모름   
매체에서 문의 -> 이노션에서 문의를 전달 -> 아티언스에서 설명   
불필요한 의사소통들이 존재 / **데이터 확인 권한을 받아올 수는 없을까?**

\* 과연 데이터를 모아서 뭘 하려고 하는걸까? 에 대한 궁금증이 존재  
[남들이 하니깐 우리도 해보자. 이를 위해 데이터를 모으자!] 수준이면 태깅 서비스를 한 단계 업그레이드 시킬 수 있지 않을까?  
기아의 경우 GA 개선 프로젝트를 통한 리타게팅 모수 수집이 목표  
현대 측에도 비슷한 제안이 가능하지 않을까?