

KSB AI Framework

AI 서비스 인프라 프레임워크 소개 및 개발경험담

한국전자통신연구원 | KSB 융합연구단 | 이연희

2018. 10. 18

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018



목차

개발 배경

03

코어개발 시작

10

KSB 프레임워크 둘러보기

25

대표 Use cases

36

KSB 오픈소스 프레임워크 확장하기

41

마치며...

56

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018



개발 배경

SOSCON 2018

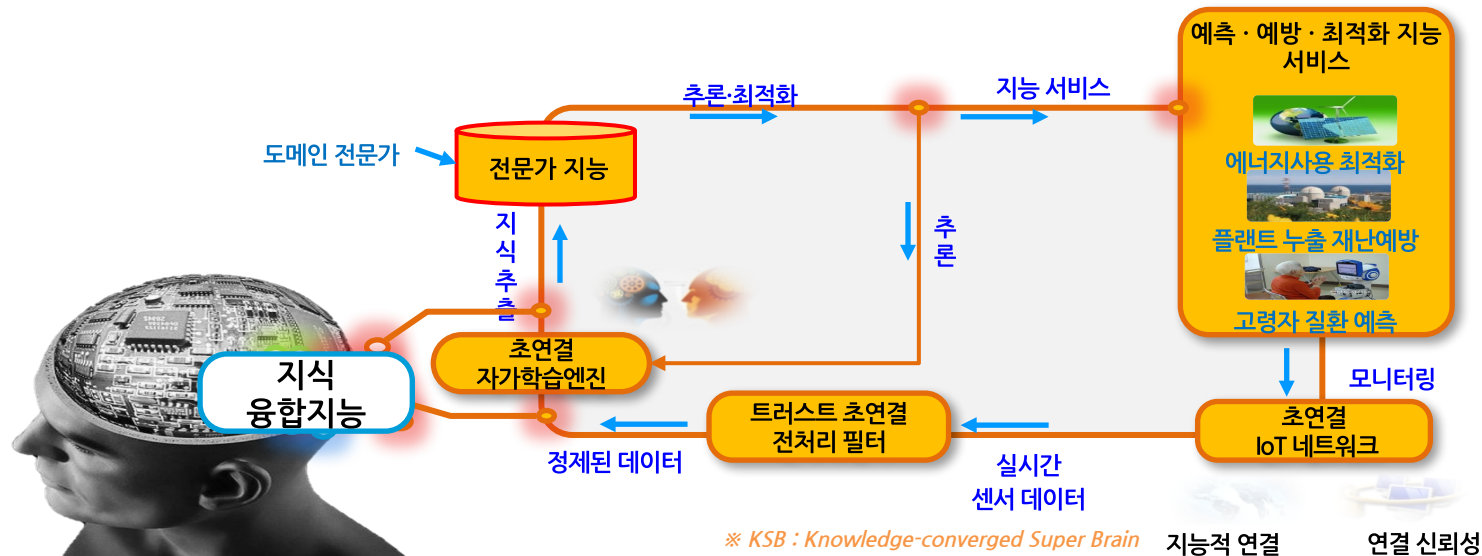
SAMSUNG OPEN SOURCE CONFERENCE 2018



주어진 문제: Knowledge-converged Super Brain

SOSCON 2018

- ① 상태를 모니터링하여 데이터를 수집
- ② 수집된 데이터를 정제
- ③ 기계학습·지식추출하고
- ④ 도메인의 전문가지식과 융합하여 추론·최적화함으로써
- ⑤ 예측·예방·최적화 지능 서비스 제공





주어진 문제: Knowledge-converged Super Brain

SOSCON 2018



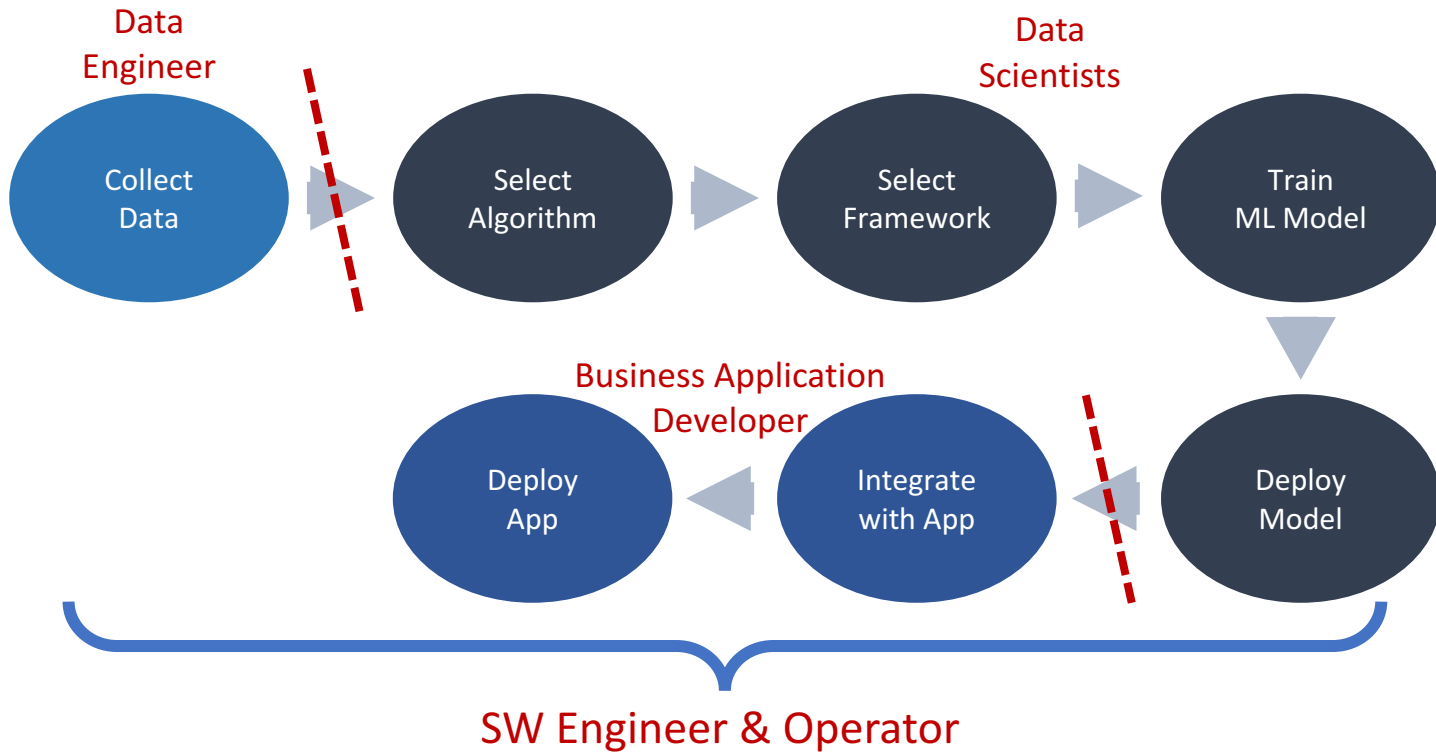
에너지/플랜트/헬스 우선 대상





데이터로부터 AI 기반 응용까지...

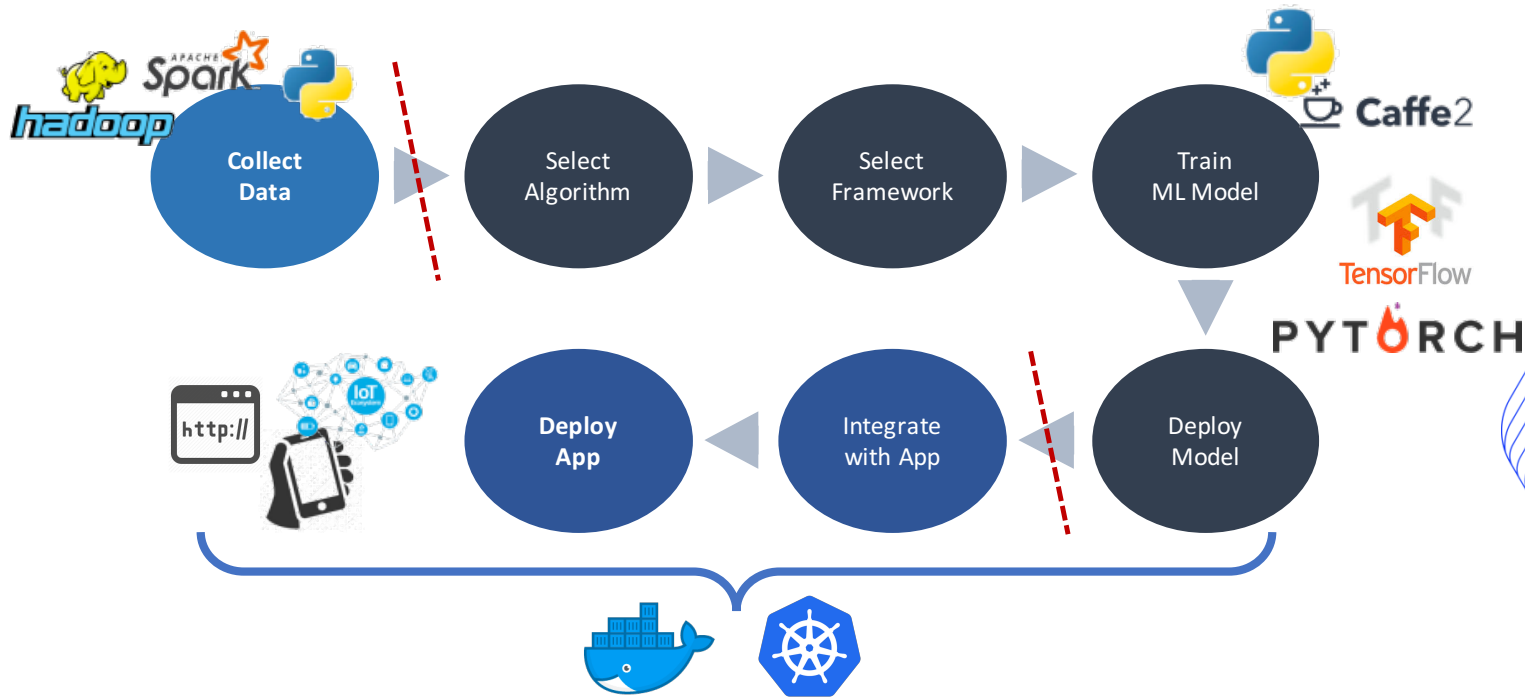
SOSCON 2018





데이터로부터 AI 기반 응용까지...

SOSCON 2018





고민...

SOSCON 2018

- **다양한 역할이 필요하고,**
 - 데이터 엔지니어 & 데이터사이언티스트
 - SW 엔지니어 & 오퍼레이터, 비즈니스 응용 개발자 ...
- **그들은 다양한 기술을 사용함**
 - Tensorflow, caffe, pyTorch
 - Spark, Hadoop, DB, NoSQL, Docker, Kubernetes, Web, App ...
- **무엇보다도 반복적이고 silo된 개발 !!!**

- **하지만 현실은 ...**

- 세상은 끊임없이 변하고 기술은 발전한다 !!
- 우리가 다 구축해 줄 수도 인력도 없고...
- 그렇다면 클라우드가 답 ? vender lock-in 문제, 요금제...





프레임워크 전략으로 전환 !!!

SOSCON 2018

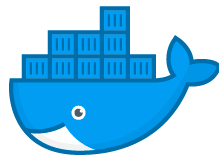


다양한 오픈소스를 프레임워크 개발에 활용

SOSCON 2018

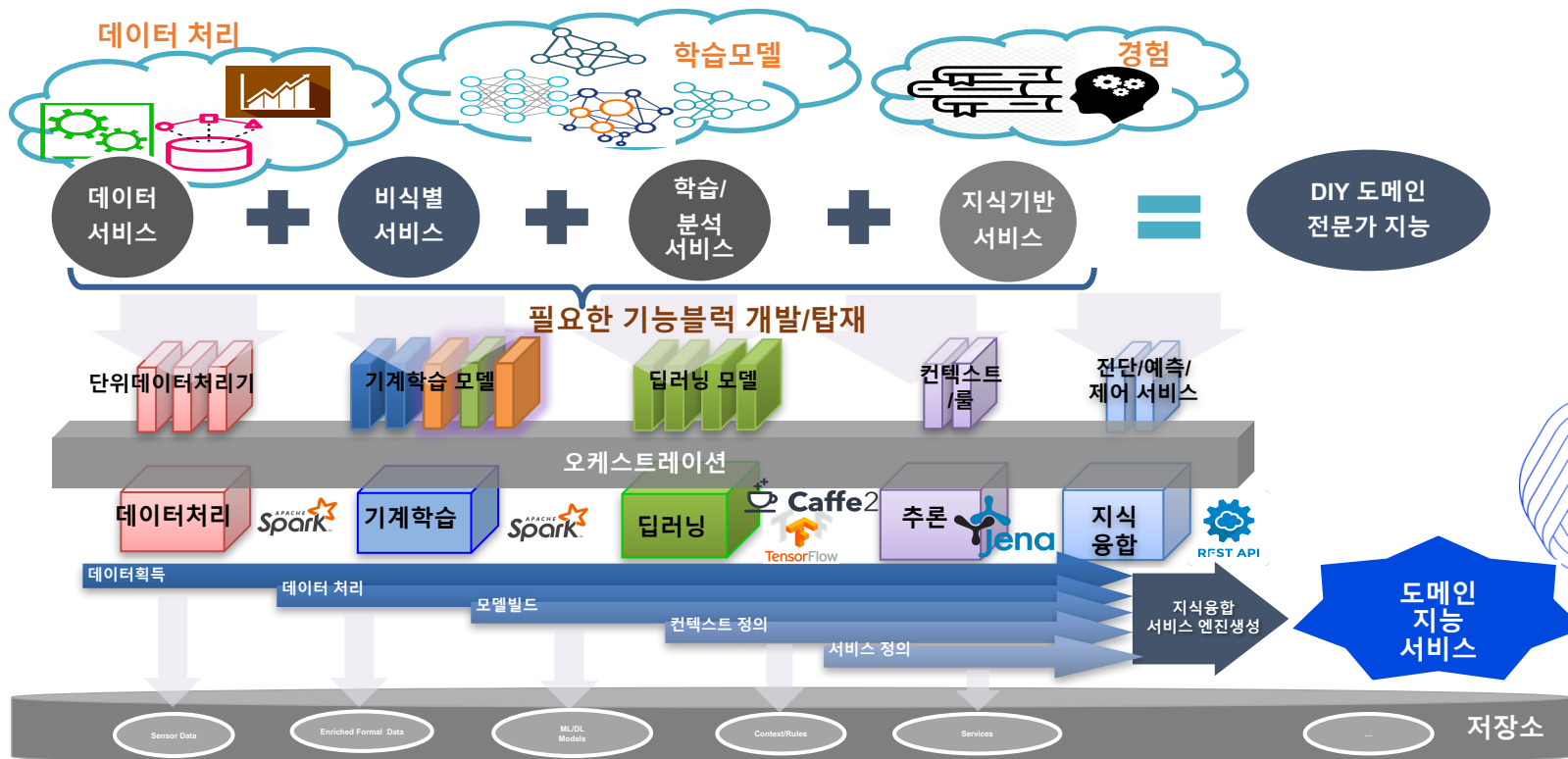


PostgreSQL



프레임워크 개발전략으로 전환

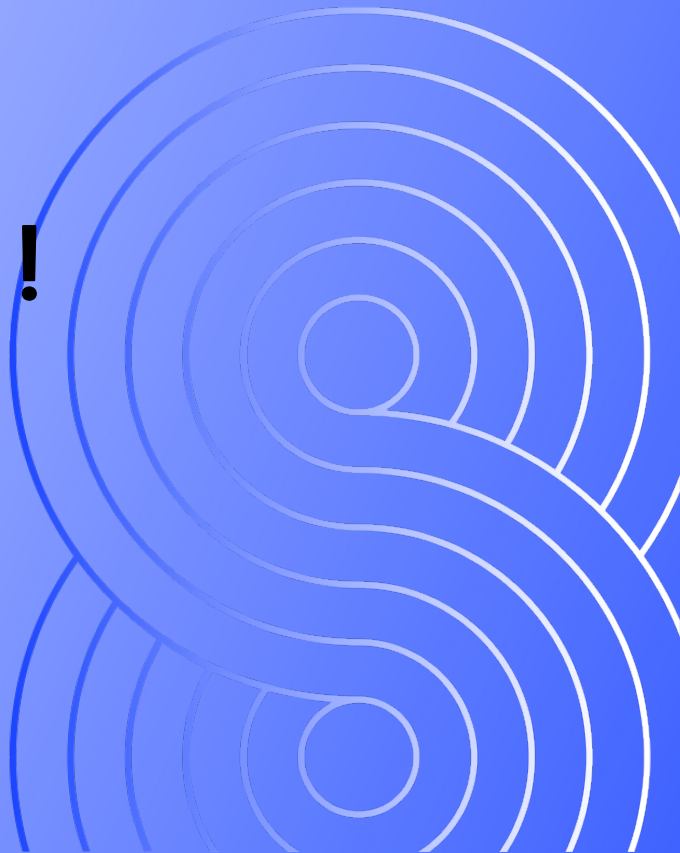
SOSCON 2018



코어 개발 시작 !

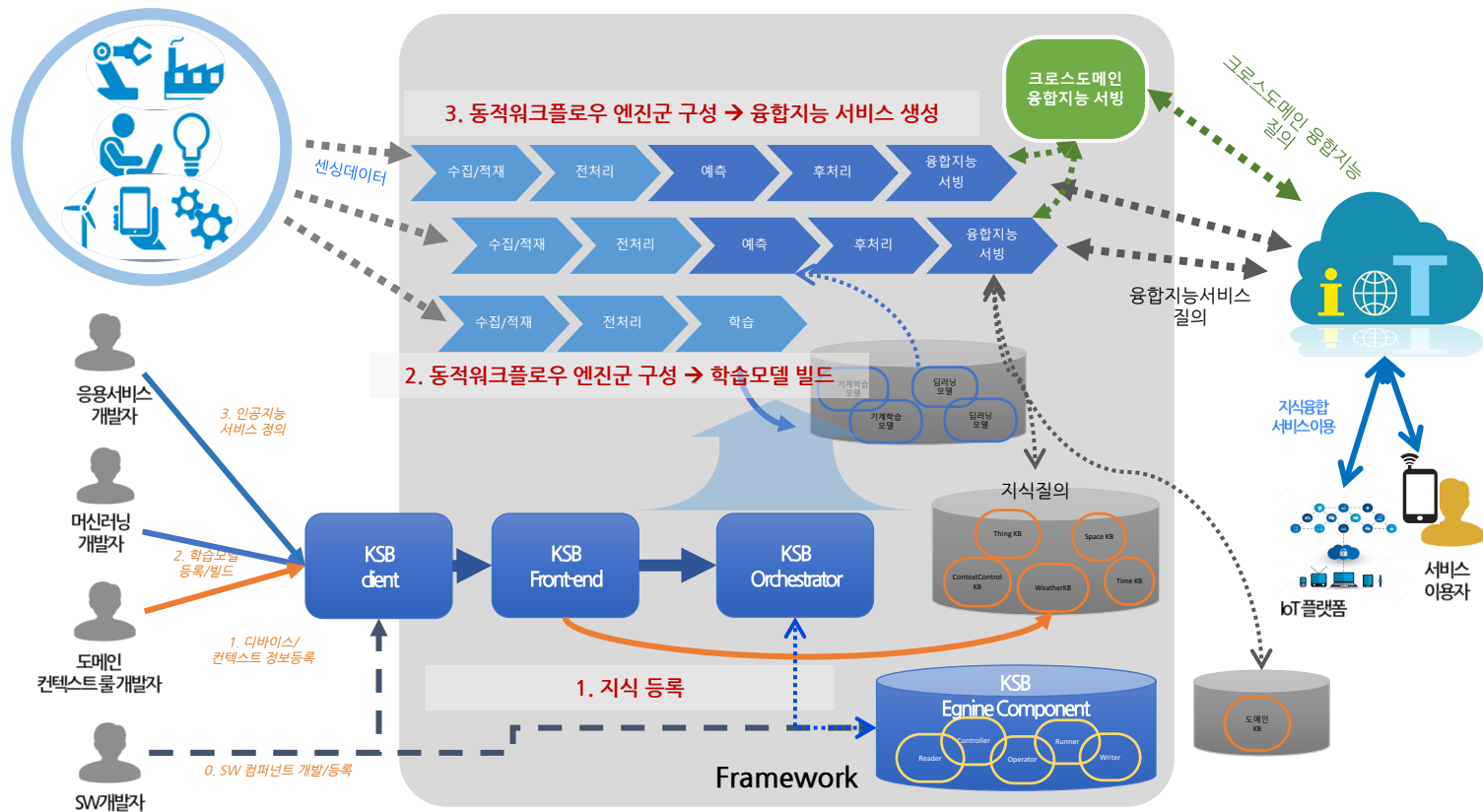
SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018



다양한 도메인의 플랫폼 개발에 활용가능한 공통구조

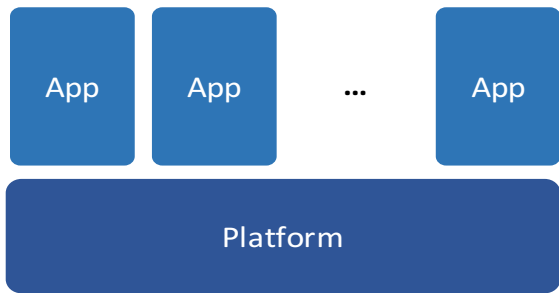
SOSCON 2018



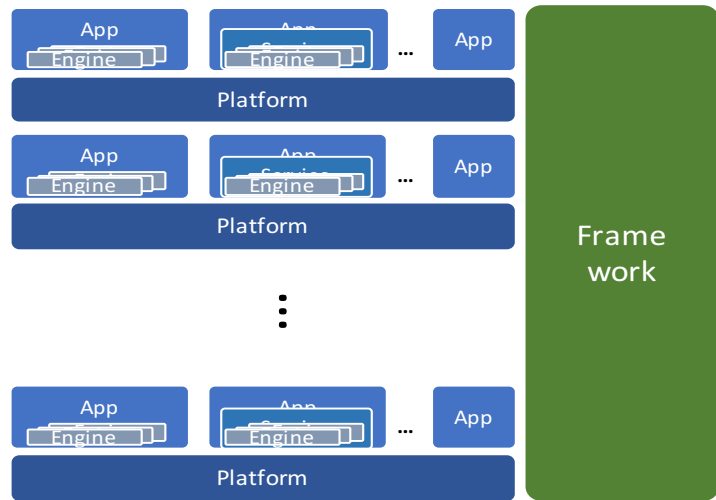


플랫폼 vs 플랫폼 + 프레임워크

SOSCON 2018



플랫폼



프레임워크 + 플랫폼





프레임워크 전략으로 전환 !!!

SOSCON 2018

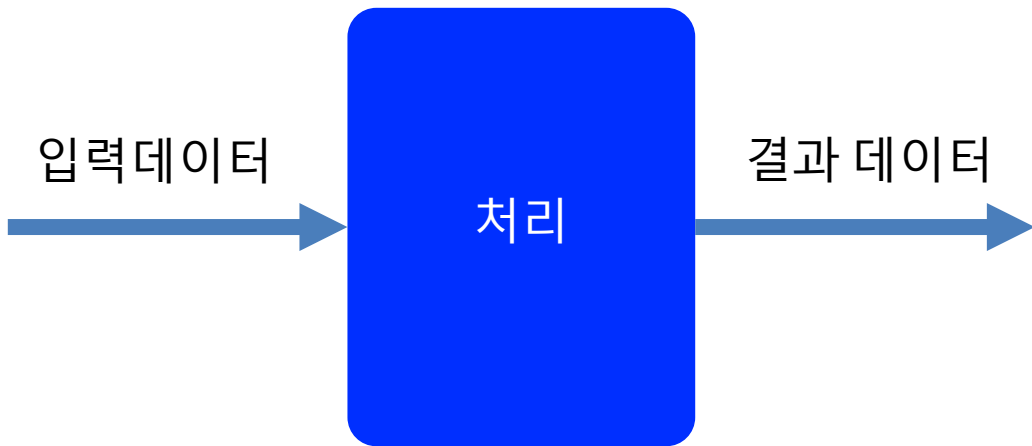




코어개발: 추상화하기

SOSCON 2018

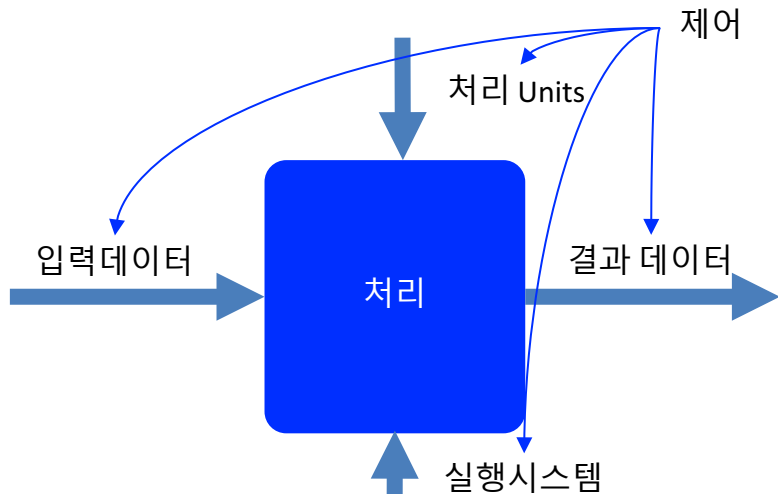
추상화를 위한 공통점을 찾기!





코어개발: 추상화하기

확장점을 부여하자!



- 데이터 IO
- 데이터 처리



- 데이터 IO
- 데이터 처리*
- 연계 실행 플랫폼
- 흐름 정의 및 실행제어

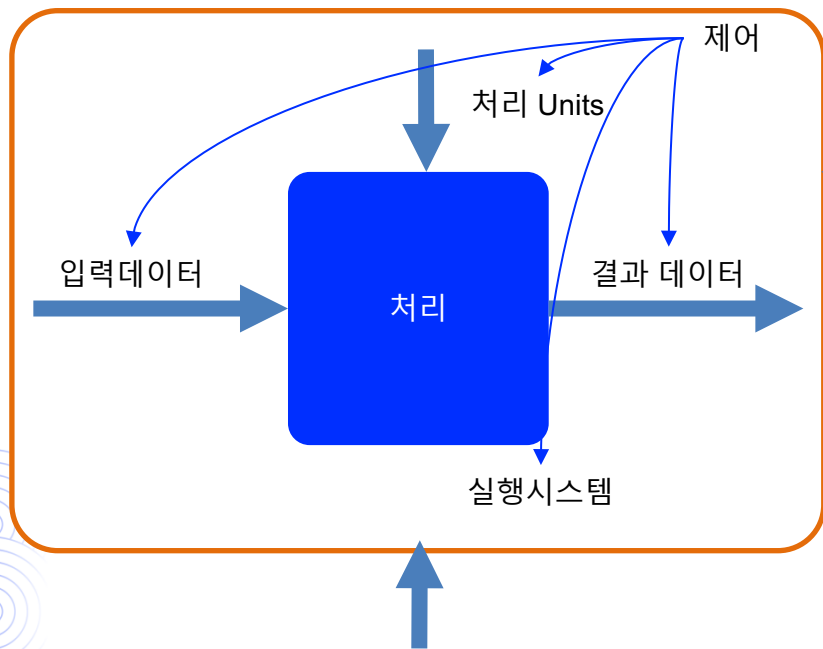
SOSCON 2018



코어개발: 추상화하기

SOSCON 2018

확장점을 부여하자!



엔진

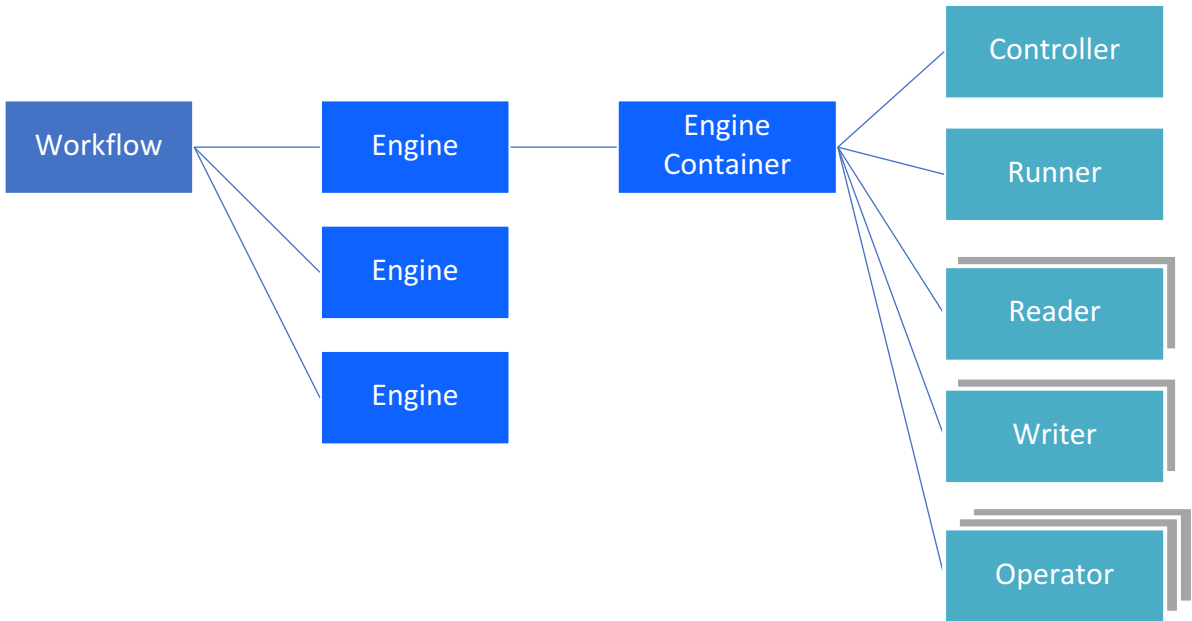
- 엔진 컨테이너 + 엔진 컴퍼넌트
- 모든 컴퍼넌트는 하나의 제어 안에 존재
- 엔진 간의 의존관계 제거 (Data-driven)



코어개발: 추상화하기

SOSCON 2018

컨셉을 추상화하고 하나로 엮자 !

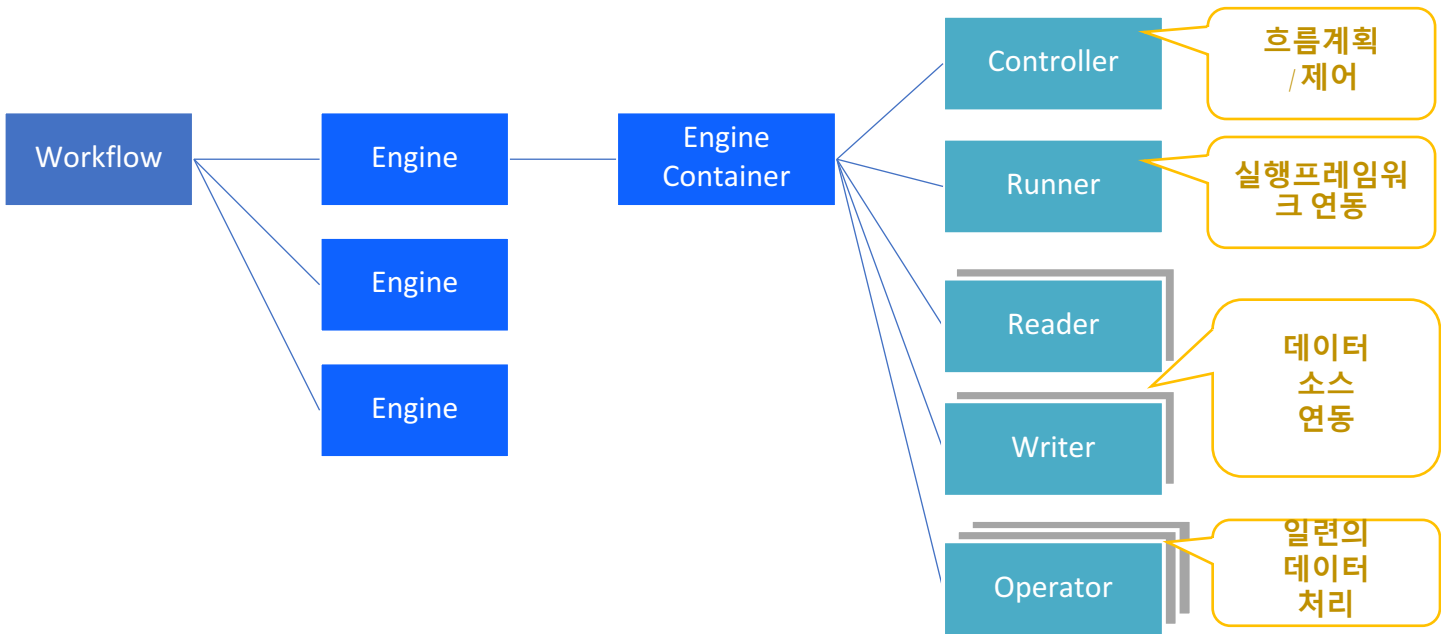




코어개발: 추상화하기

SOSCON 2018

컨셉을 추상화하고 하나로 엮자 !



- **워크플로우**
 - 일련의 작업을 정의하고 실행하기 위한 단위
 - 하나 이상의 엔진에 대한 정의로 구성
- **엔진**
 - 목적을 달성하기 위한 일을 수행하는 하나의 독립된 모듈로서 서브시스템에 해당
 - 엔진컨테이너와 컴퍼넌트들의 조합으로 구성
- **엔진 컴퍼넌트 컨테이너**
 - 필요한 목적에 따른 하나의 실행 서브시스템을 구성하기 위한 틀
 - 물리적 배포의 단위
- **컴퍼넌트**
 - 엔진을 구성하는 구성체로서 엔진컴퍼넌트 컨테이너에 탑재
 - 5 가지 유형의 컴퍼넌트

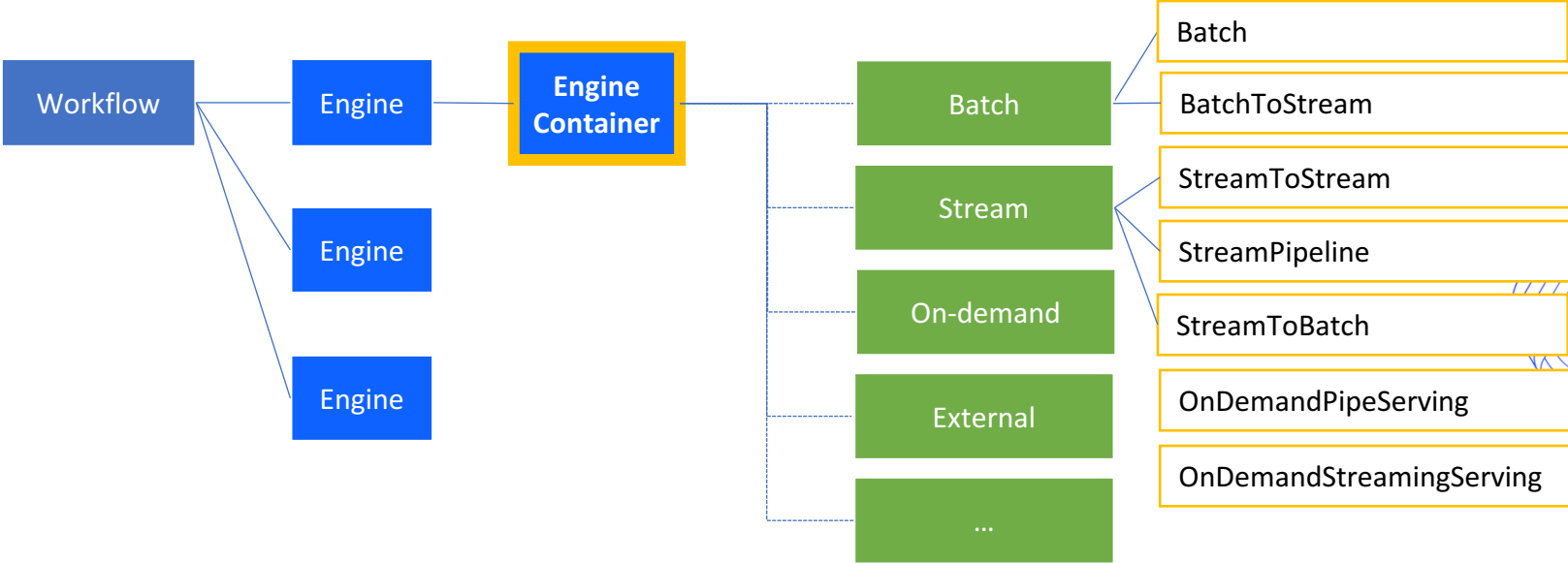




코어개발: 추상화하기

SOSCON 2018

컨셉을 추상화하고 하나로 엮자 !





- Stand-alone System vs. Distributed System
- Mutable vs. Immutable
- Streaming process + Process Call



KSB 프레임워크 둘러보기

SOSCON 2018

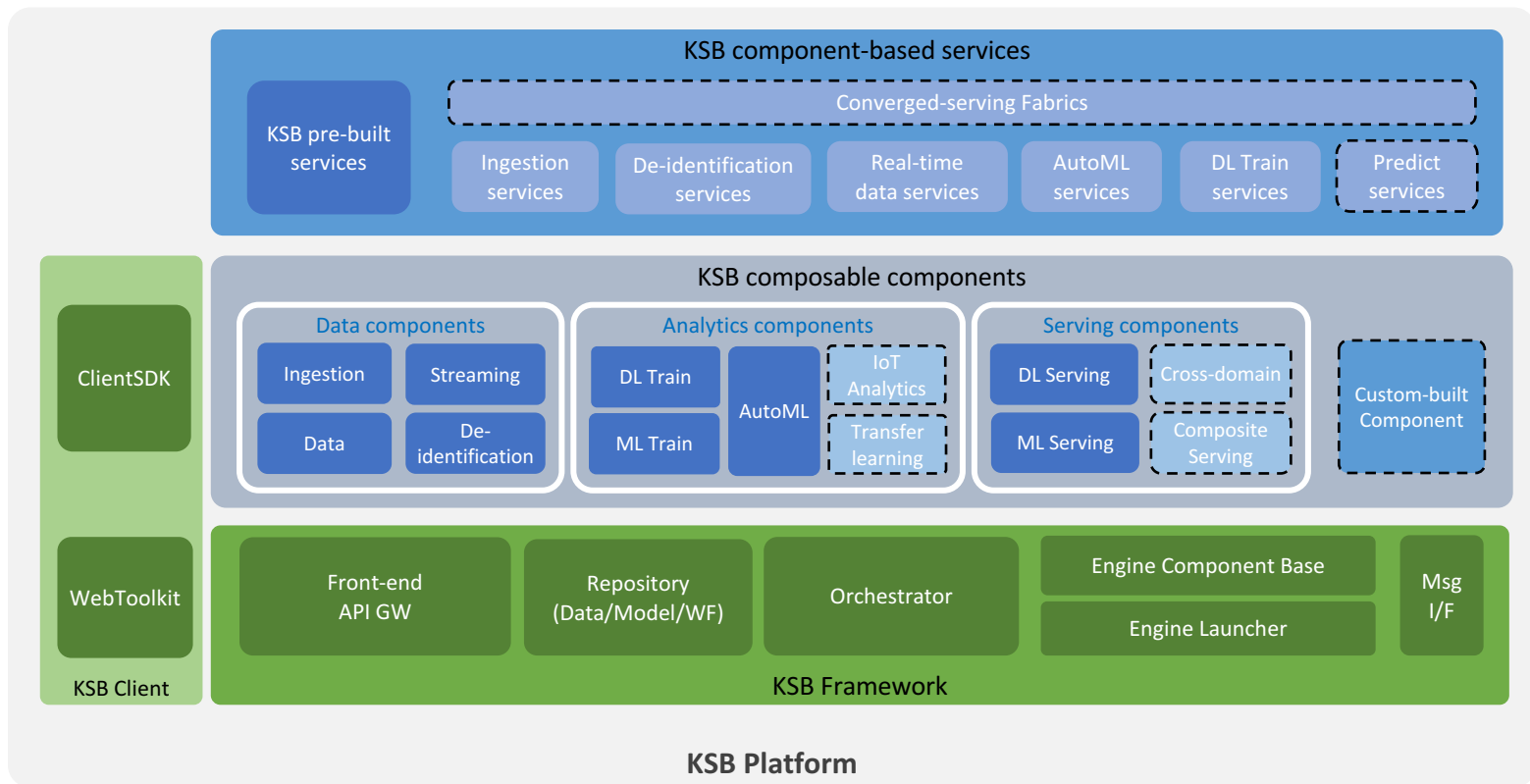
SAMSUNG OPEN SOURCE CONFERENCE 2018

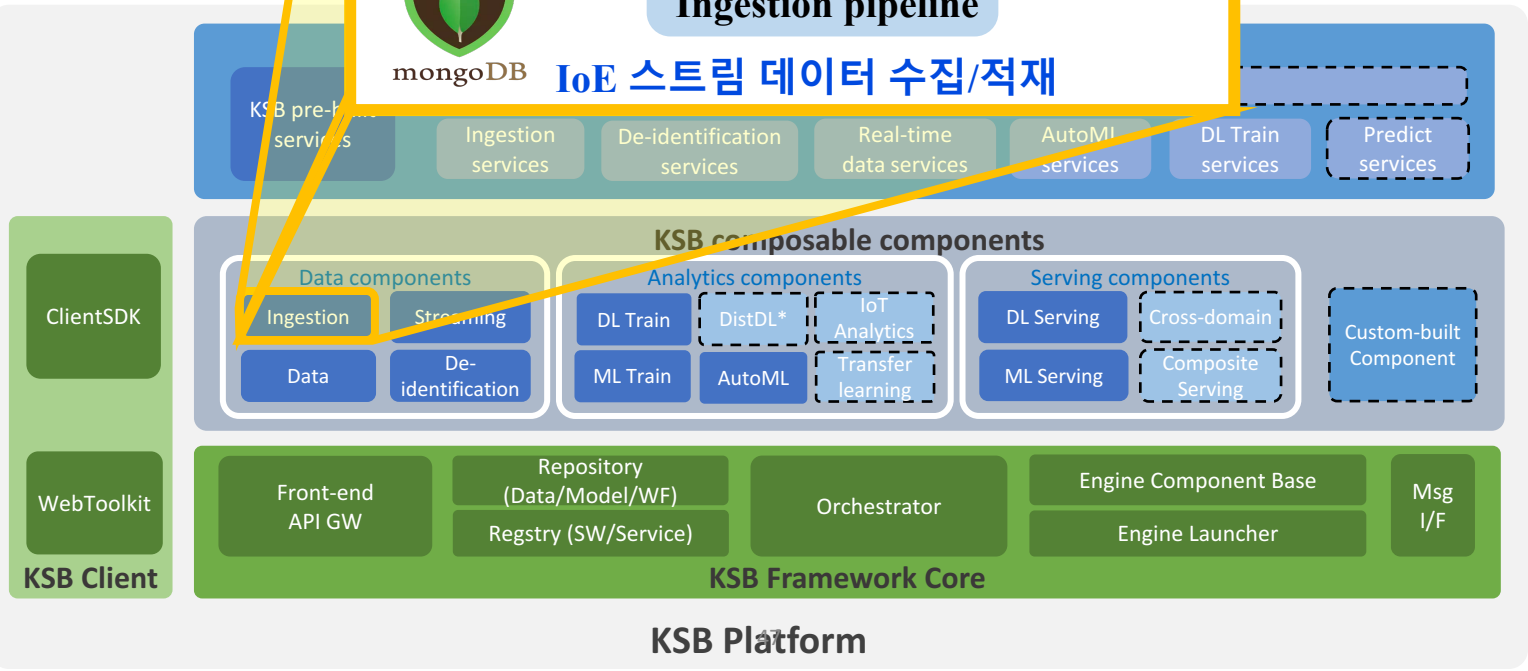


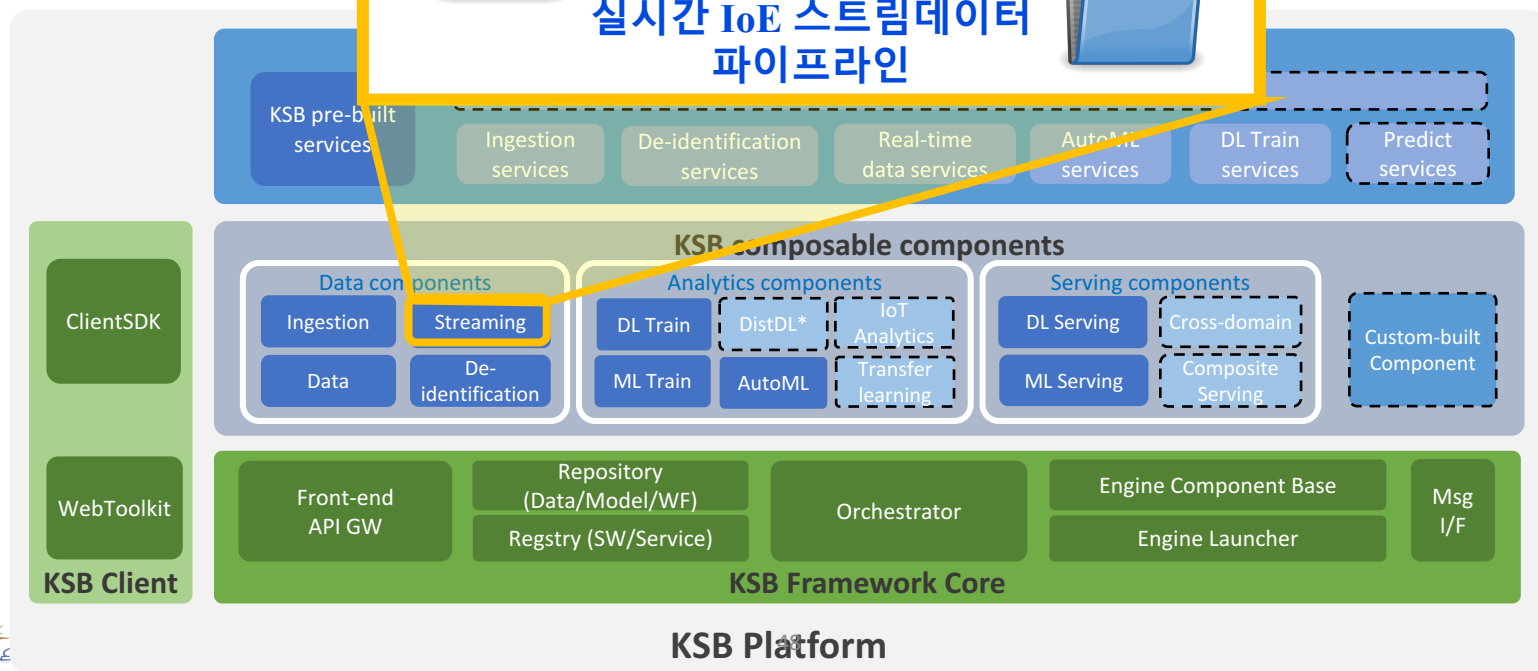


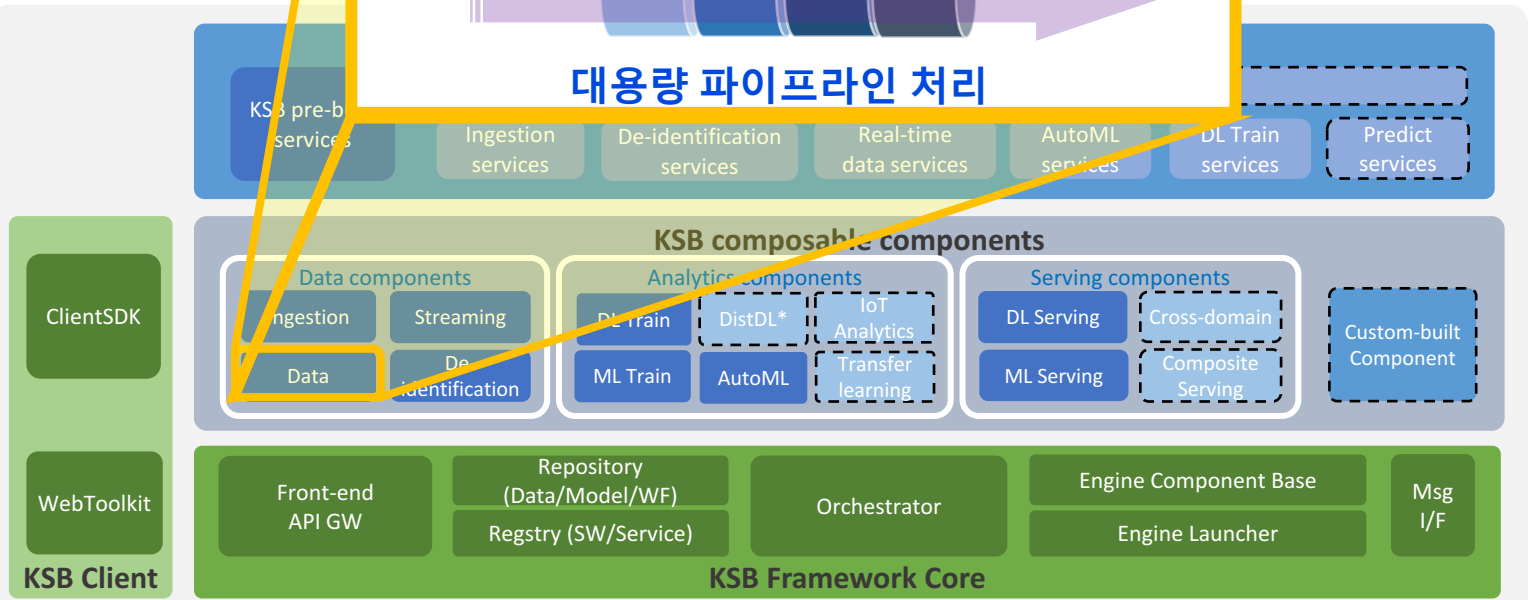
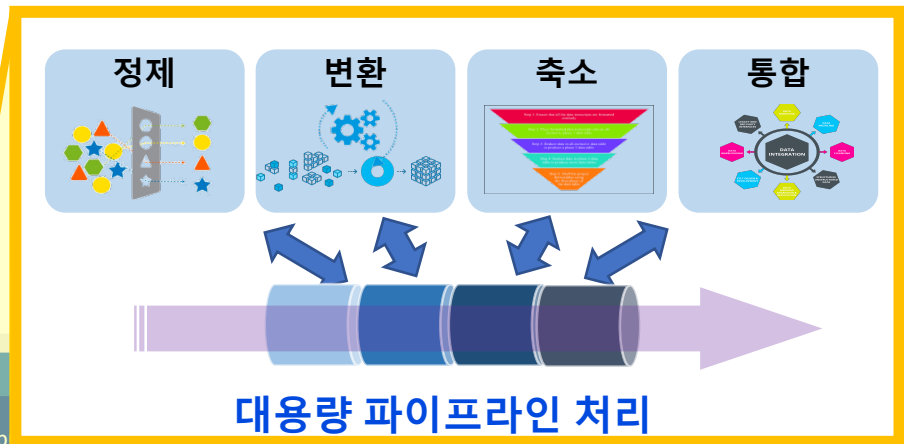
KSB Framework Stack

SOSCON 2018



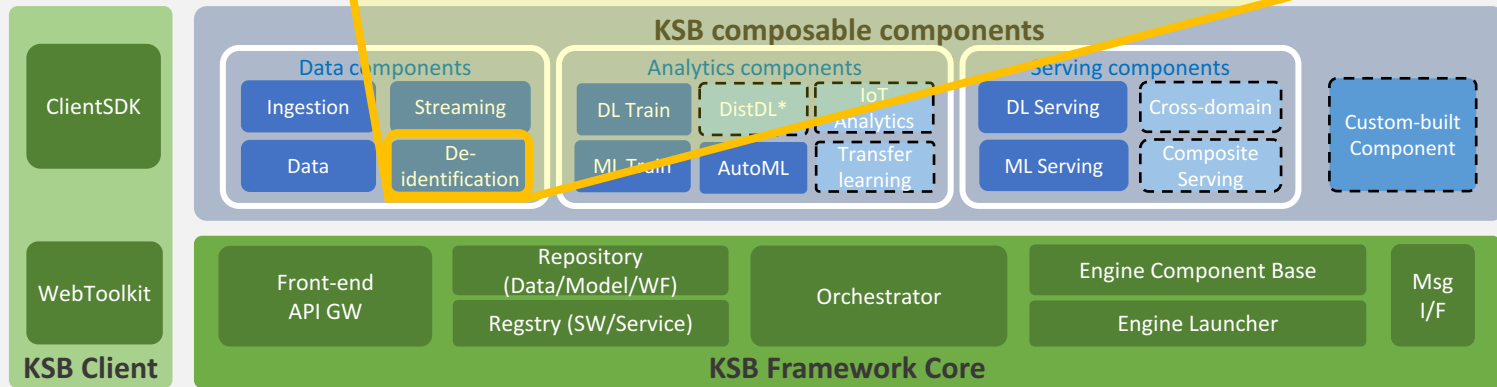


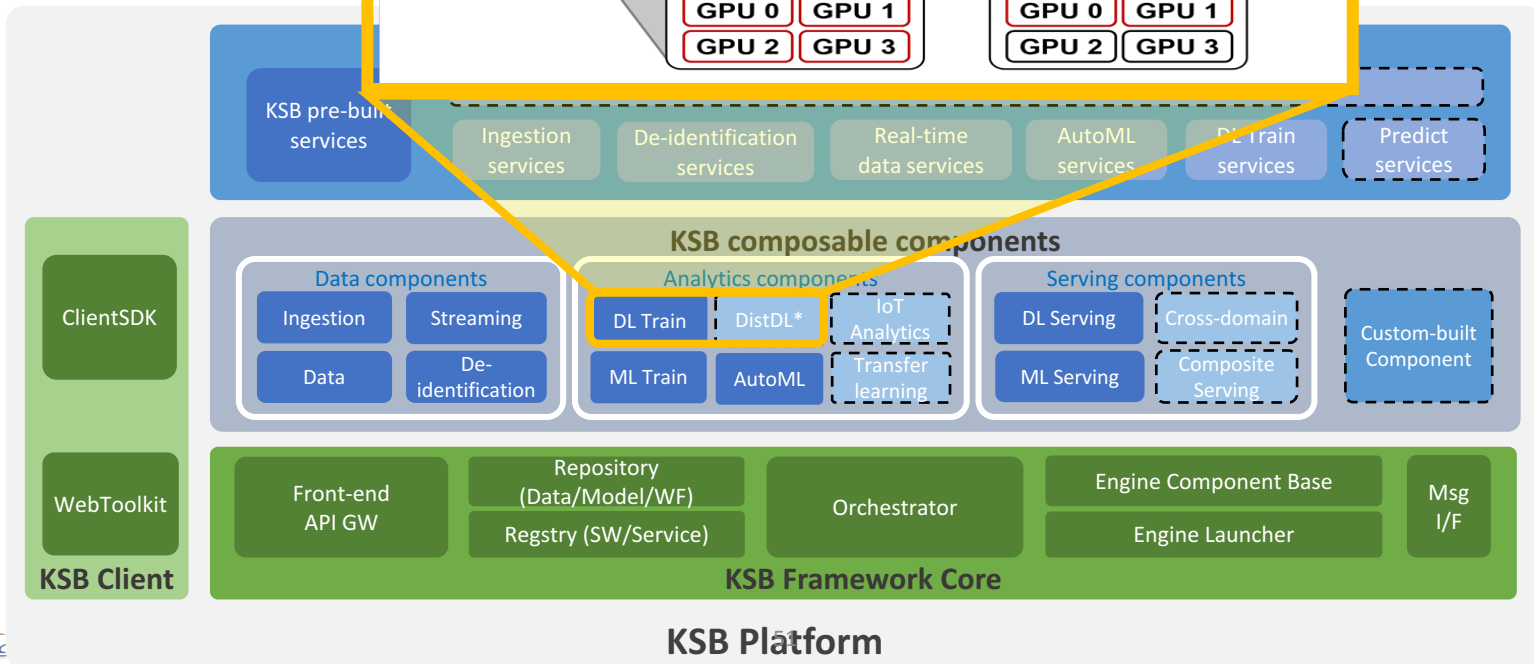
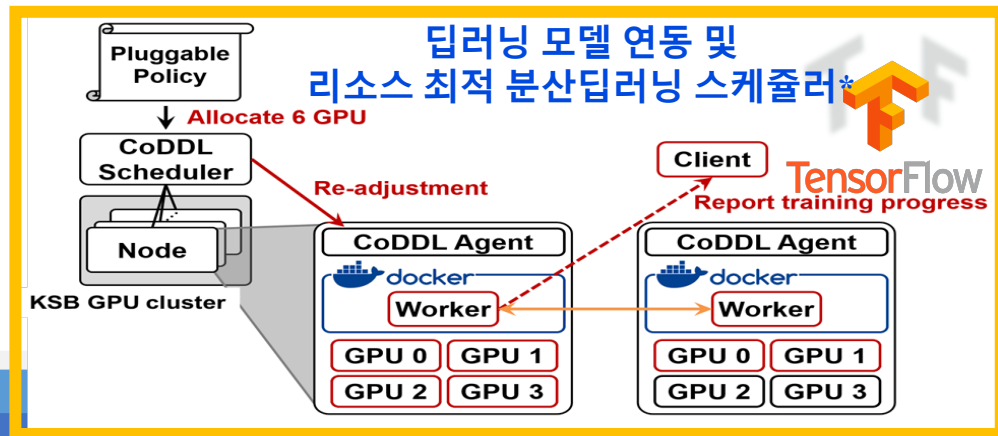




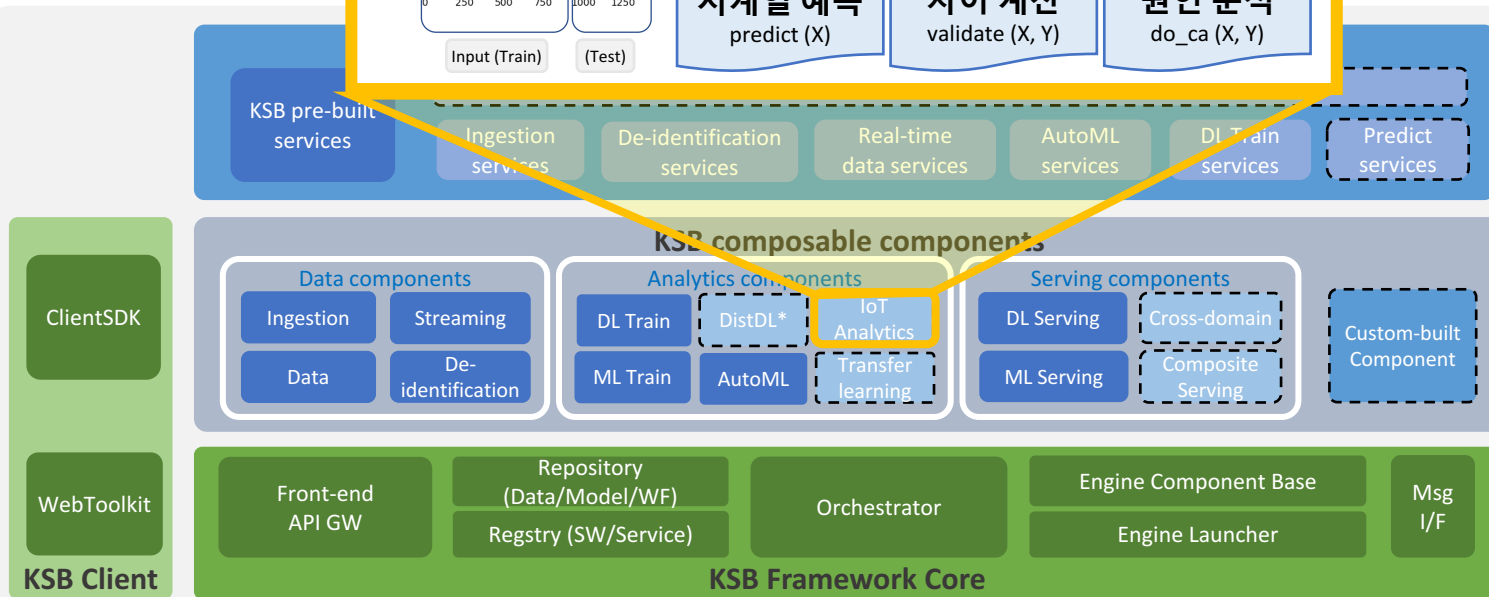
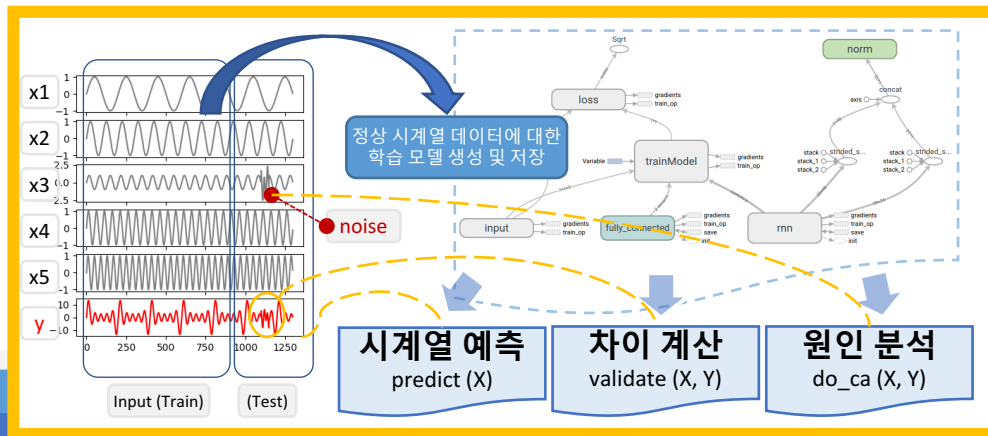


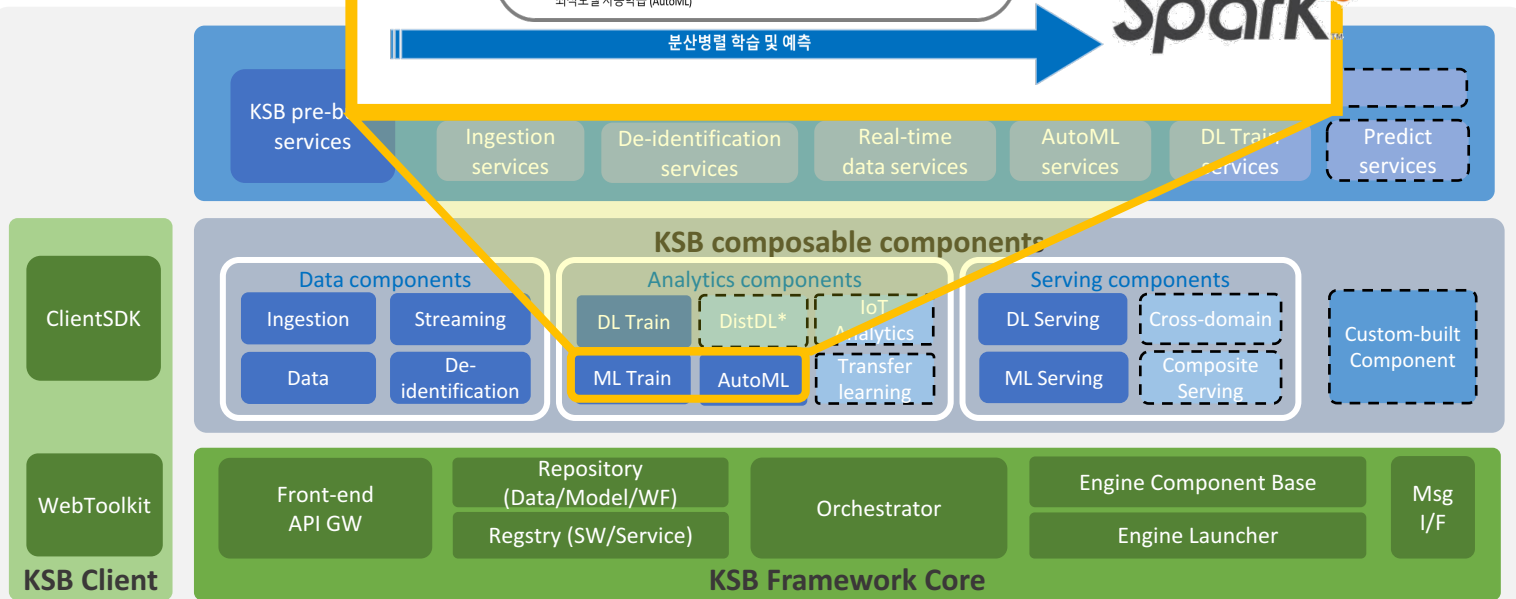
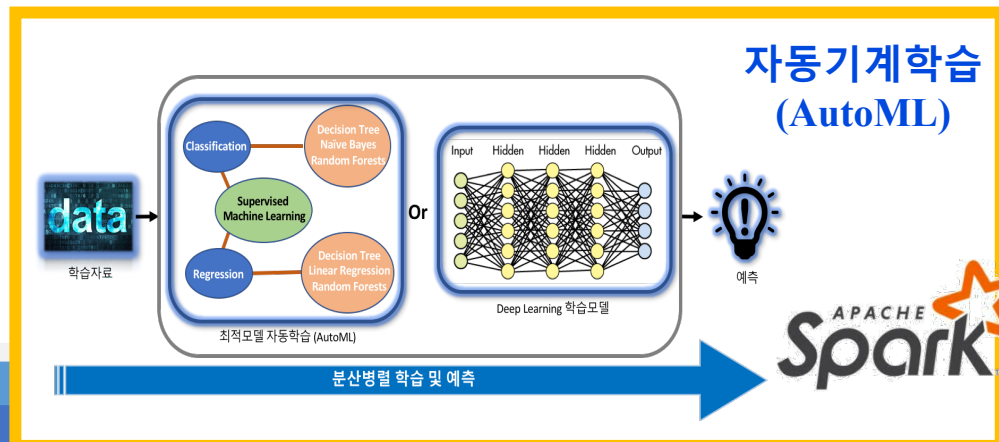
개인정보 및 민감정보 데이터 비식별 처리

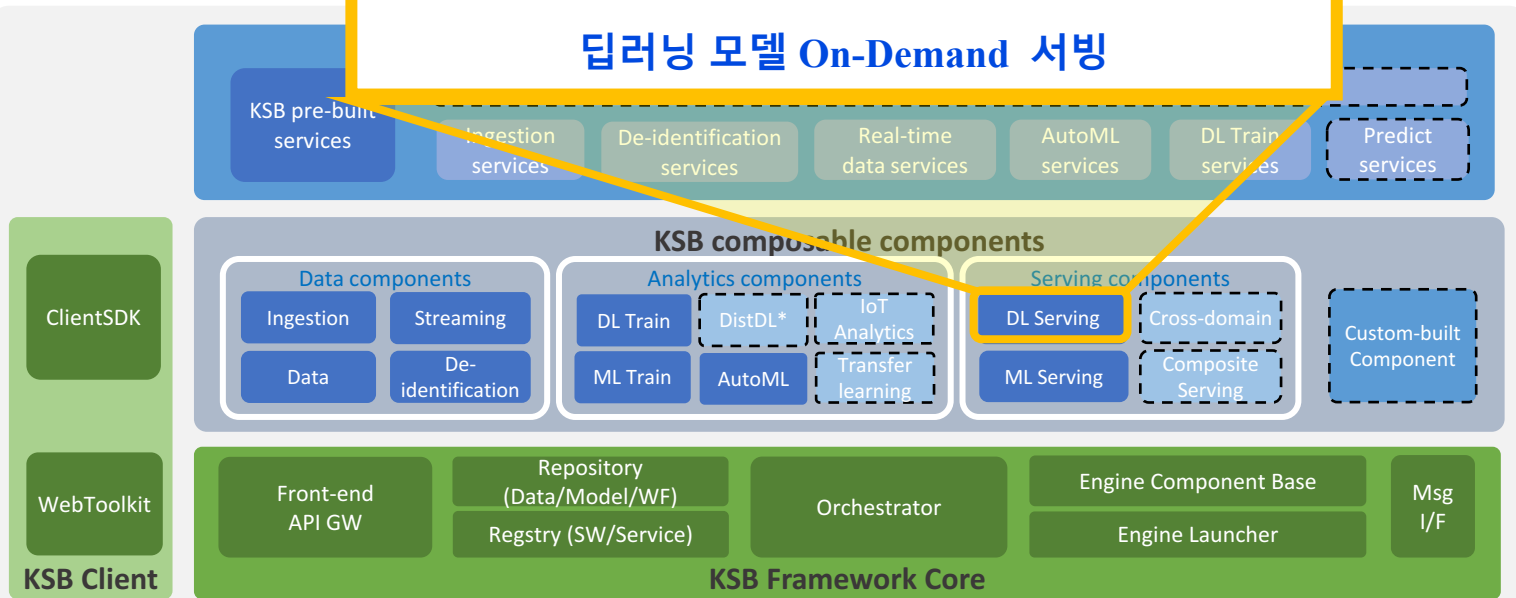
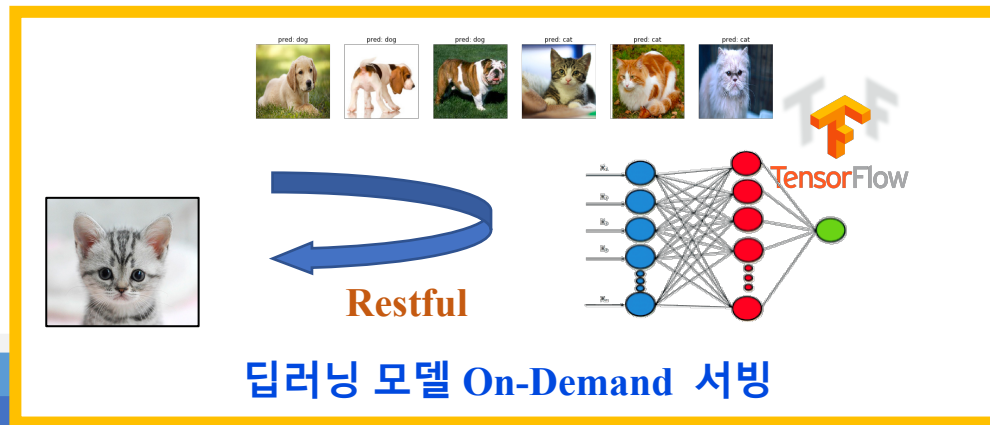


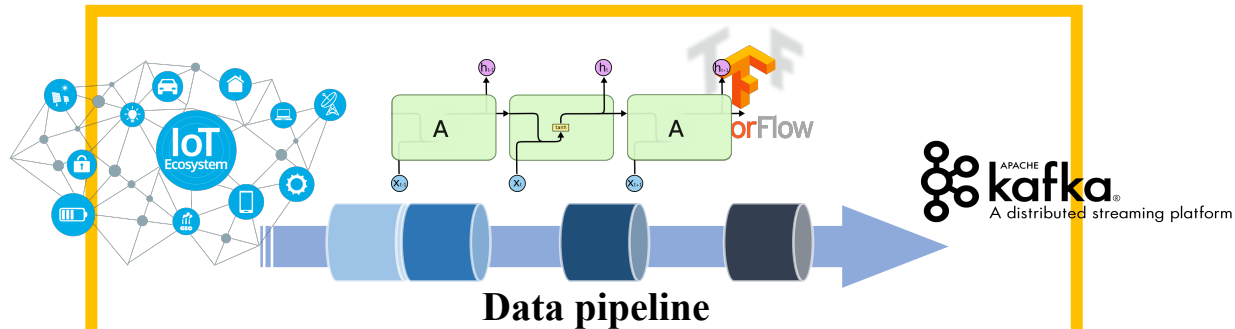


이상감지

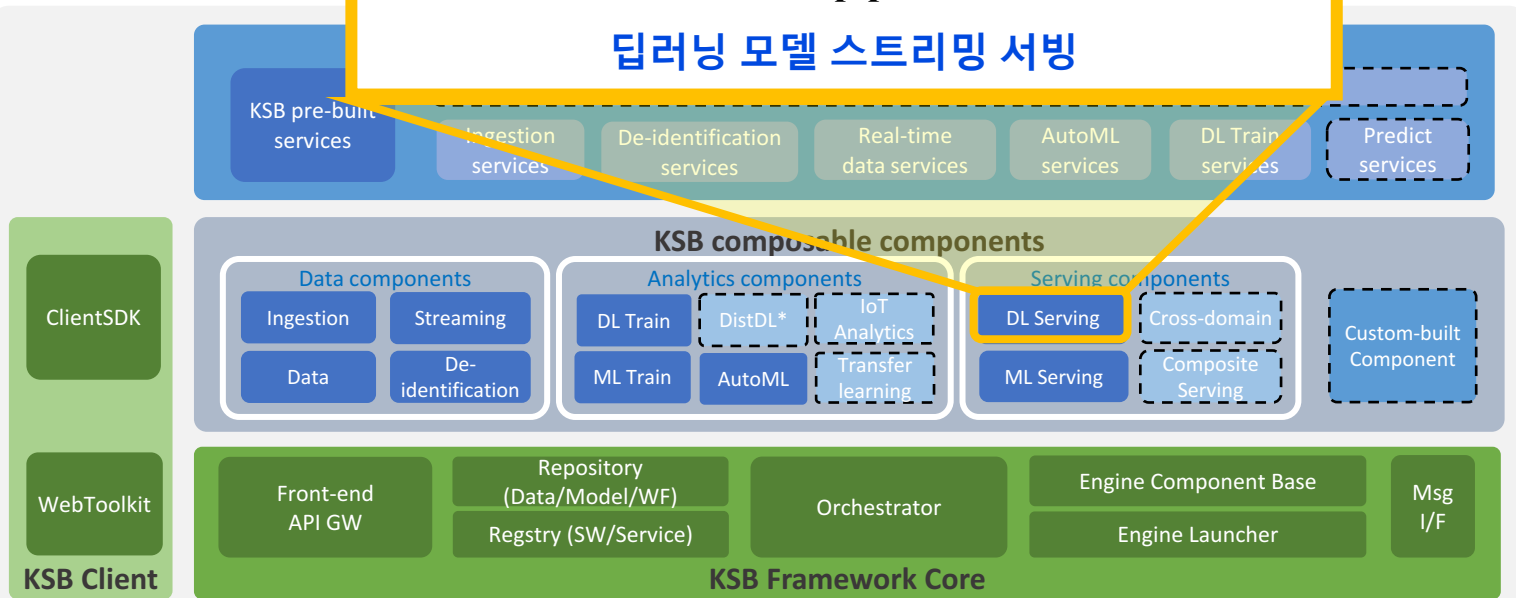


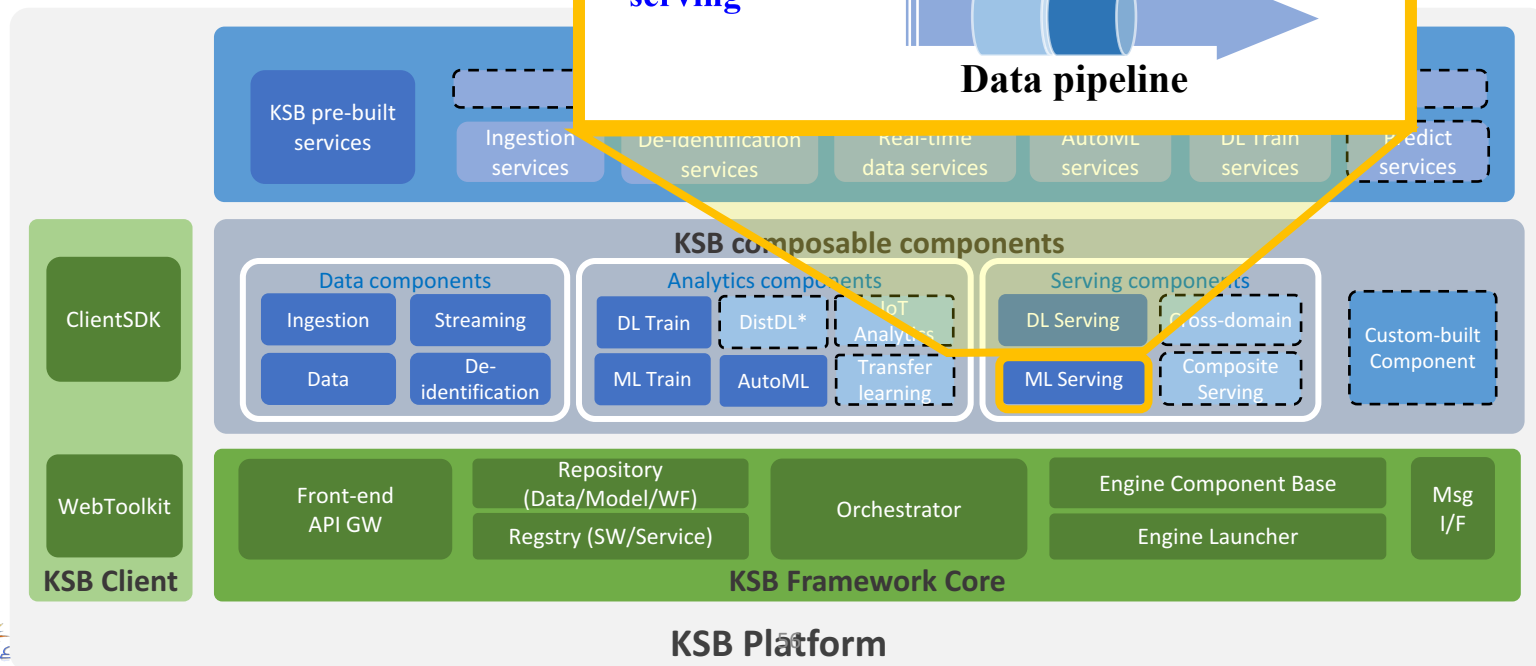
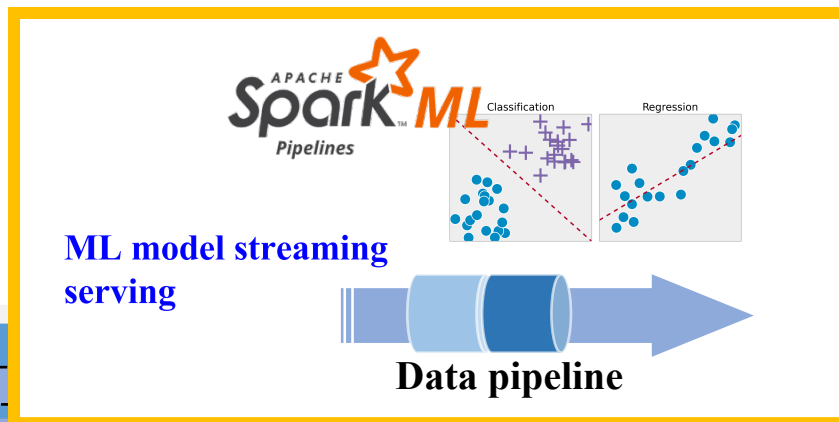


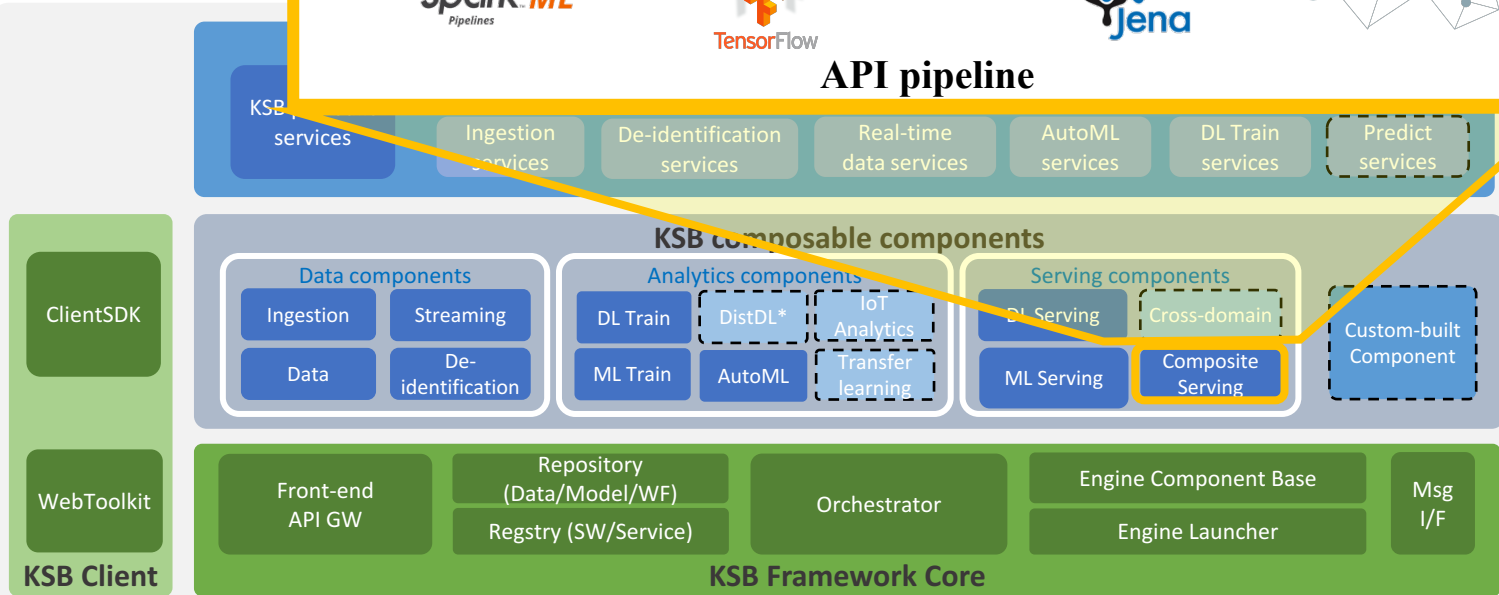
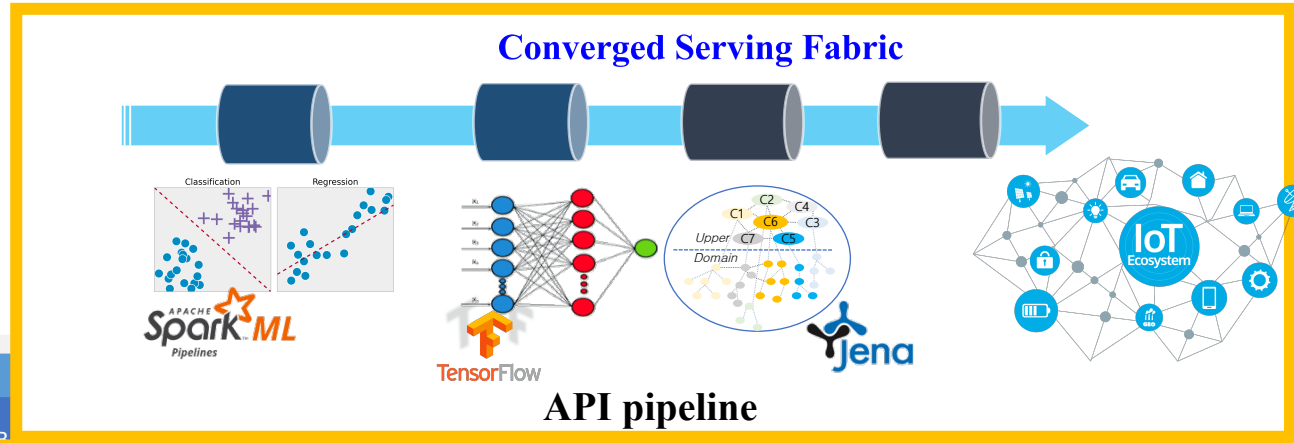


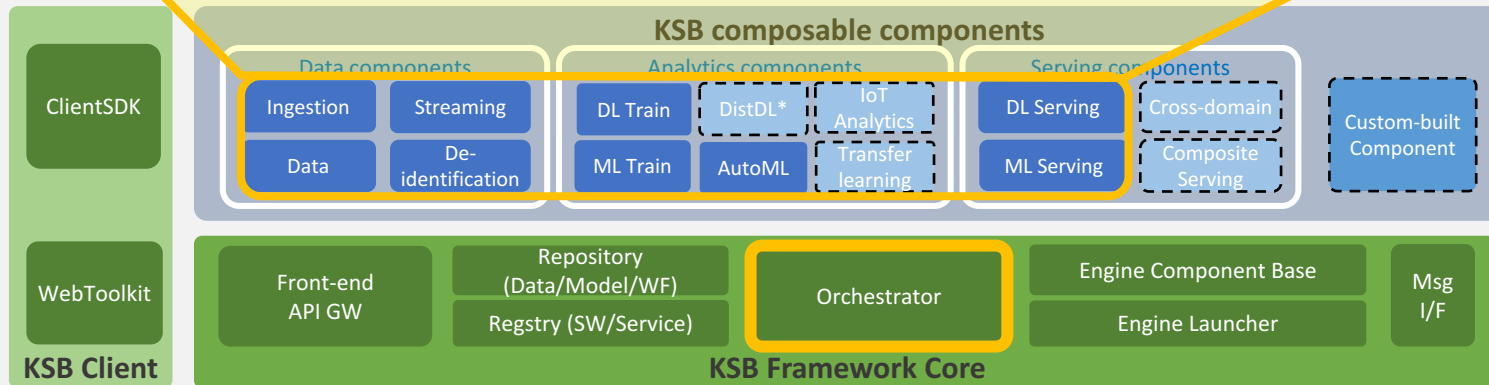
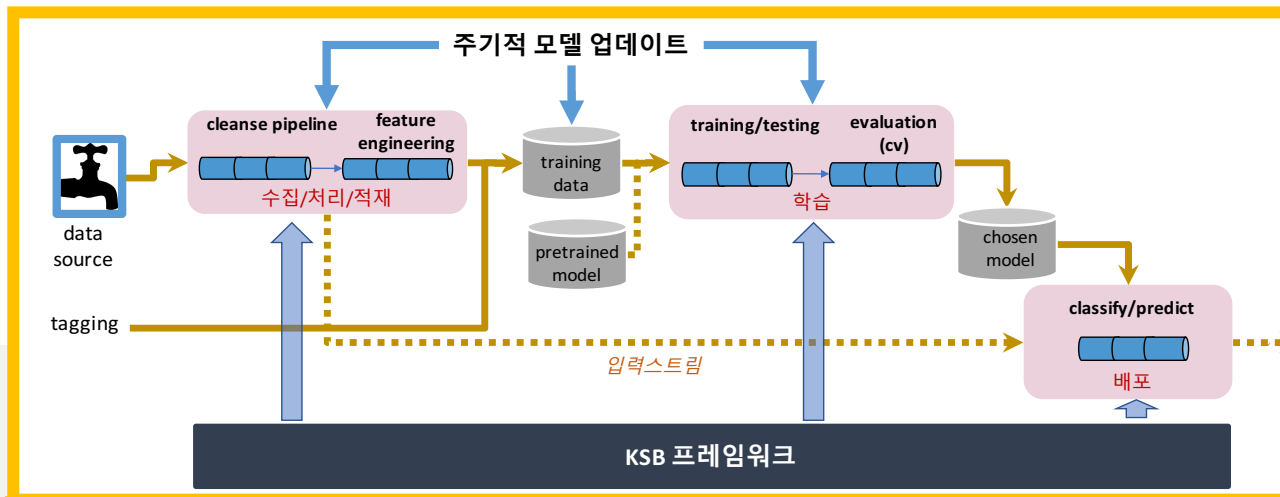


딥러닝 모델 스트리밍 서버





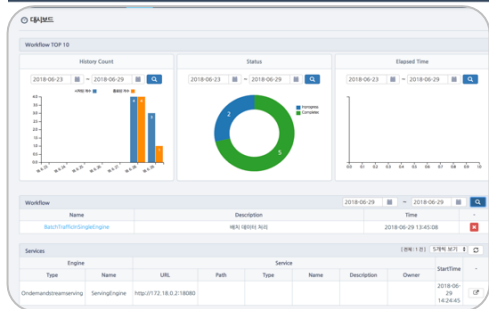
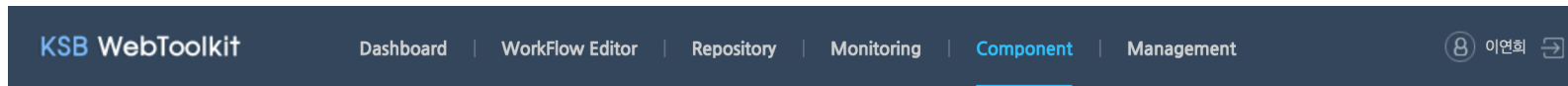




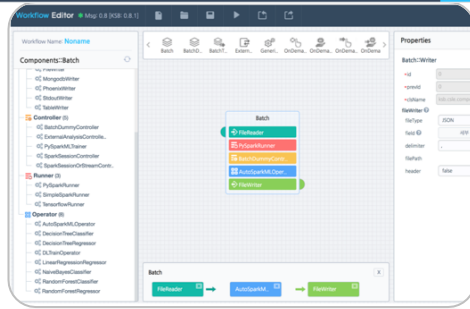


KSB 웹툴킷 둘러보기

SOSCON 2018



대쉬보드



워크플로우 편집

Repository

Name	Version	Last Modified
Batch	1.0.0	2018-09-29 13:23:44
BatchWriter	1.0.0	2018-09-29 13:23:44
BatchReader	1.0.0	2018-09-29 13:23:44

저장소/시각화

Monitoring

Name	Status	Last Modified
Batch	Running	2018-09-29 13:23:44
BatchWriter	Running	2018-09-29 13:23:44
BatchReader	Running	2018-09-29 13:23:44

모니터링

Component 관리

Name	Version	Last Modified
Batch	1.0.0	2018-09-29 13:23:44
BatchWriter	1.0.0	2018-09-29 13:23:44
BatchReader	1.0.0	2018-09-29 13:23:44

컴퍼넌트 관리

사용자 설정 관리

Name	Version	Last Modified
Batch	1.0.0	2018-09-29 13:23:44
BatchWriter	1.0.0	2018-09-29 13:23:44
BatchReader	1.0.0	2018-09-29 13:23:44

관리자



Framework APIs

SOSCON 2018

- Workflow APIs

POST	/submitJob	{ workflow spec }
POST	/startJob	{ engineId* }
POST	/stopJob	{ engineId* }

- Monitoring APIs

POST	/log
------	------

- Repository APIs

PUT	/repo/create
POST	/repo/append
GET	/repo/get
...	



- User / UserGroup / Dashboard / ...

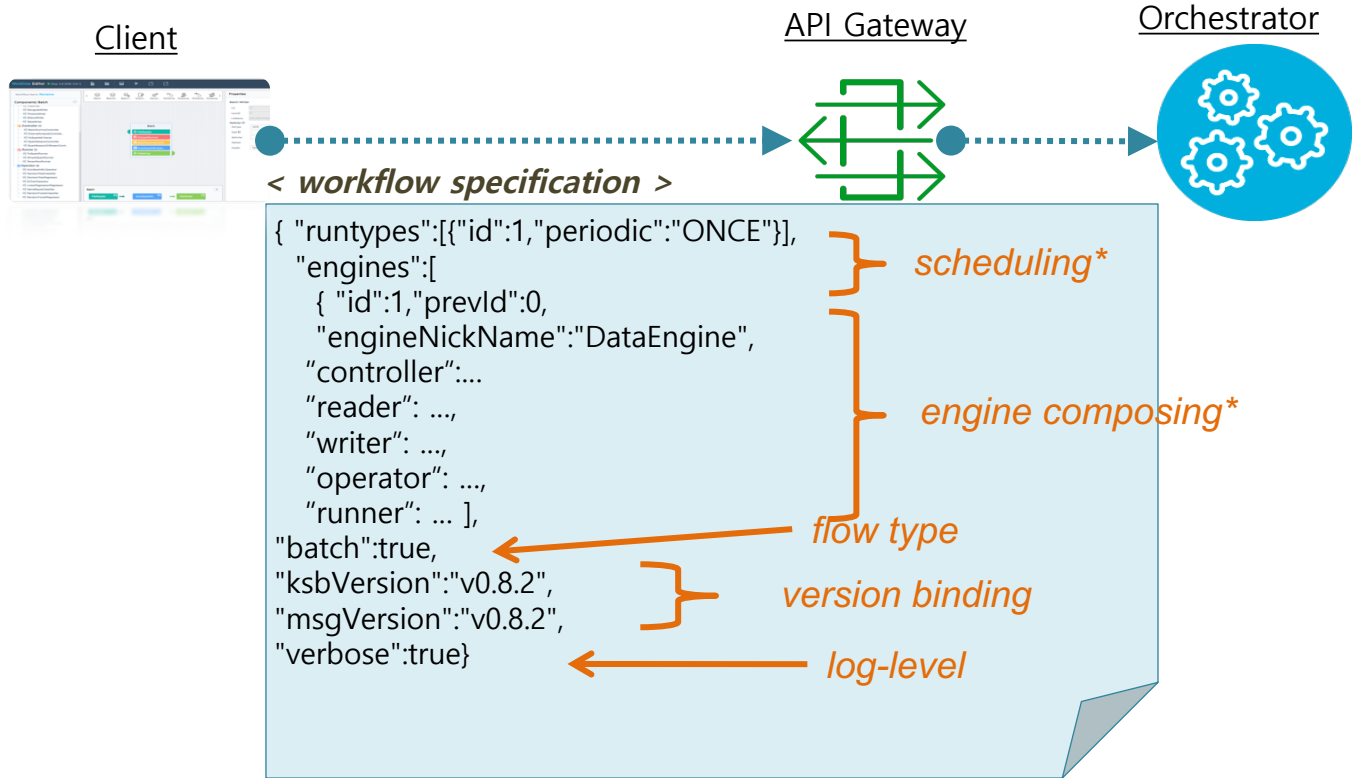
- Component management (webtoolkit)



Workflow submit API

SOSCON 2018

- Workflow spec contains:

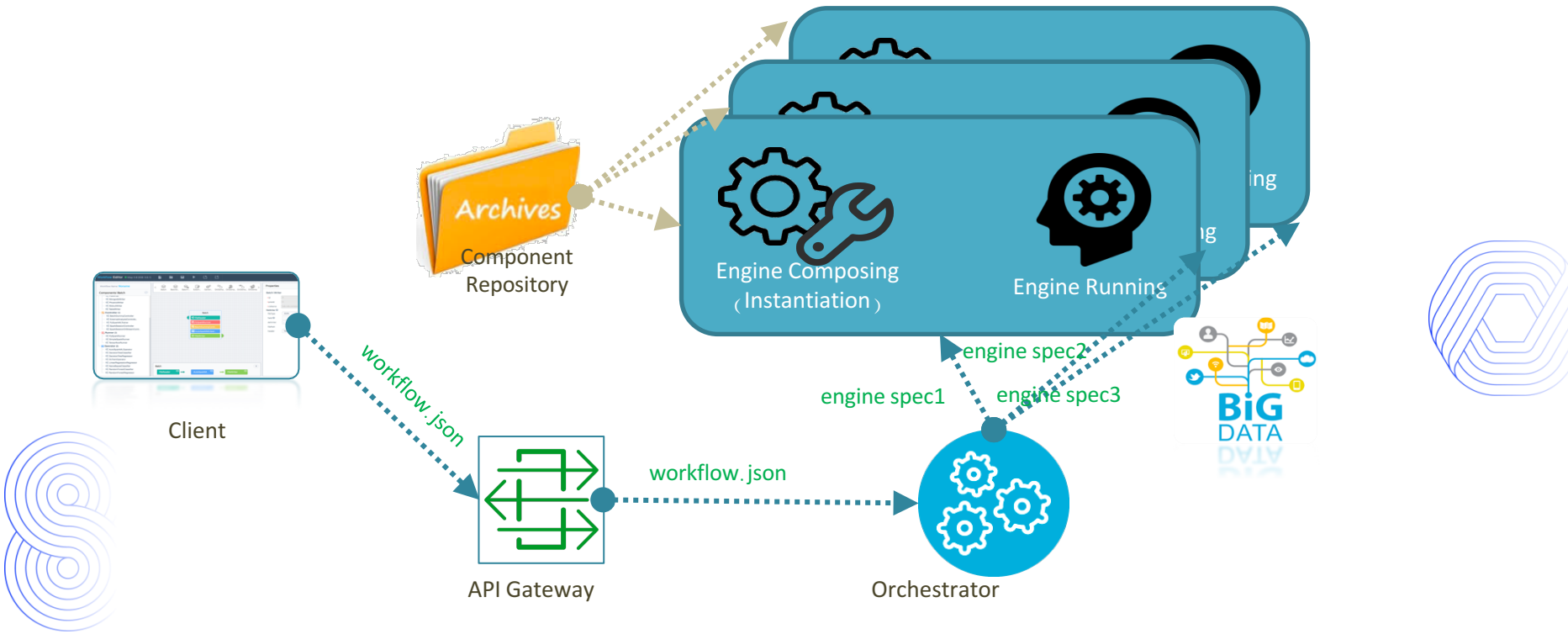




How Data Engine works

SOSCON 2018

- Batch engine
- Streaming engine

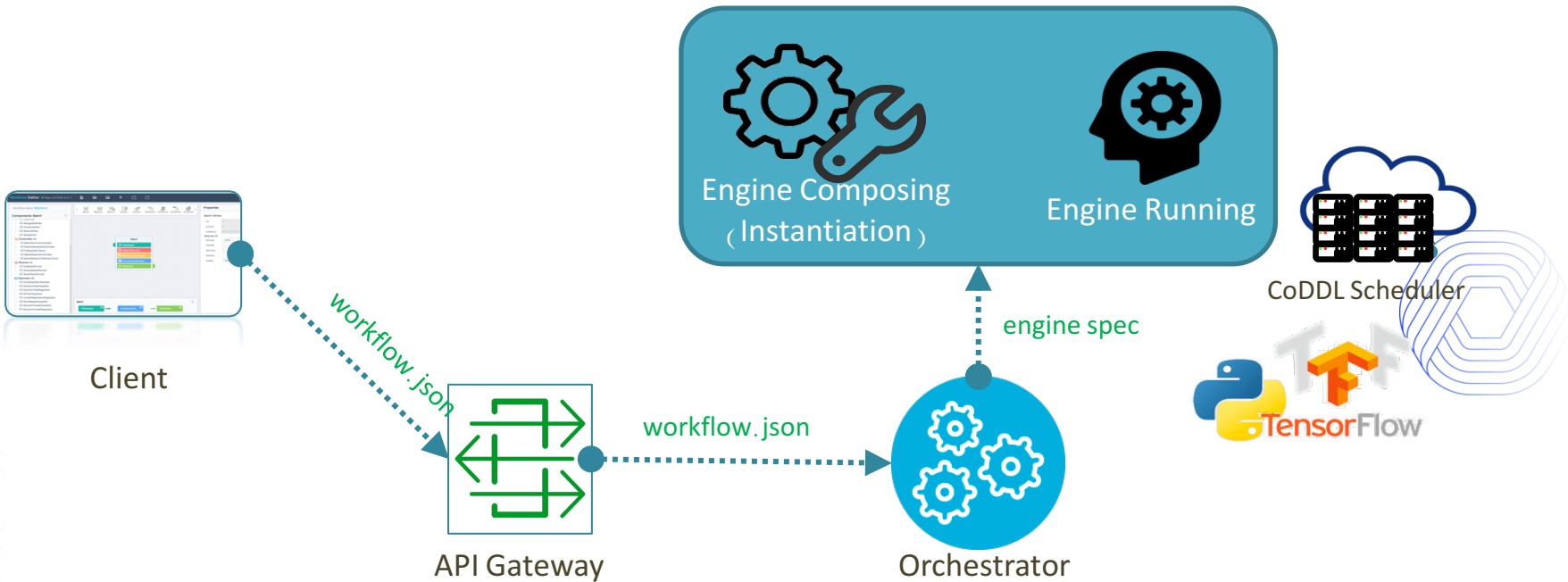




How Analysis Engine works

SOSCON 2018

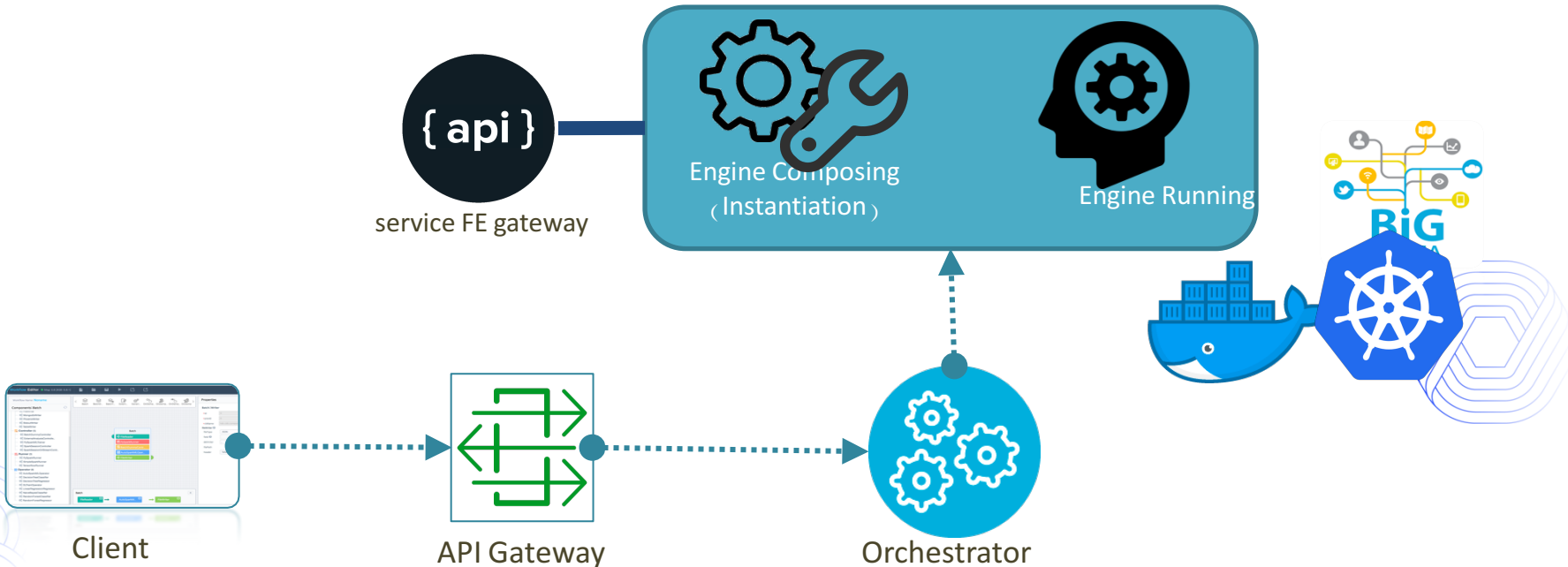
- Streaming Engine
- External Engine





How Serving Engine works

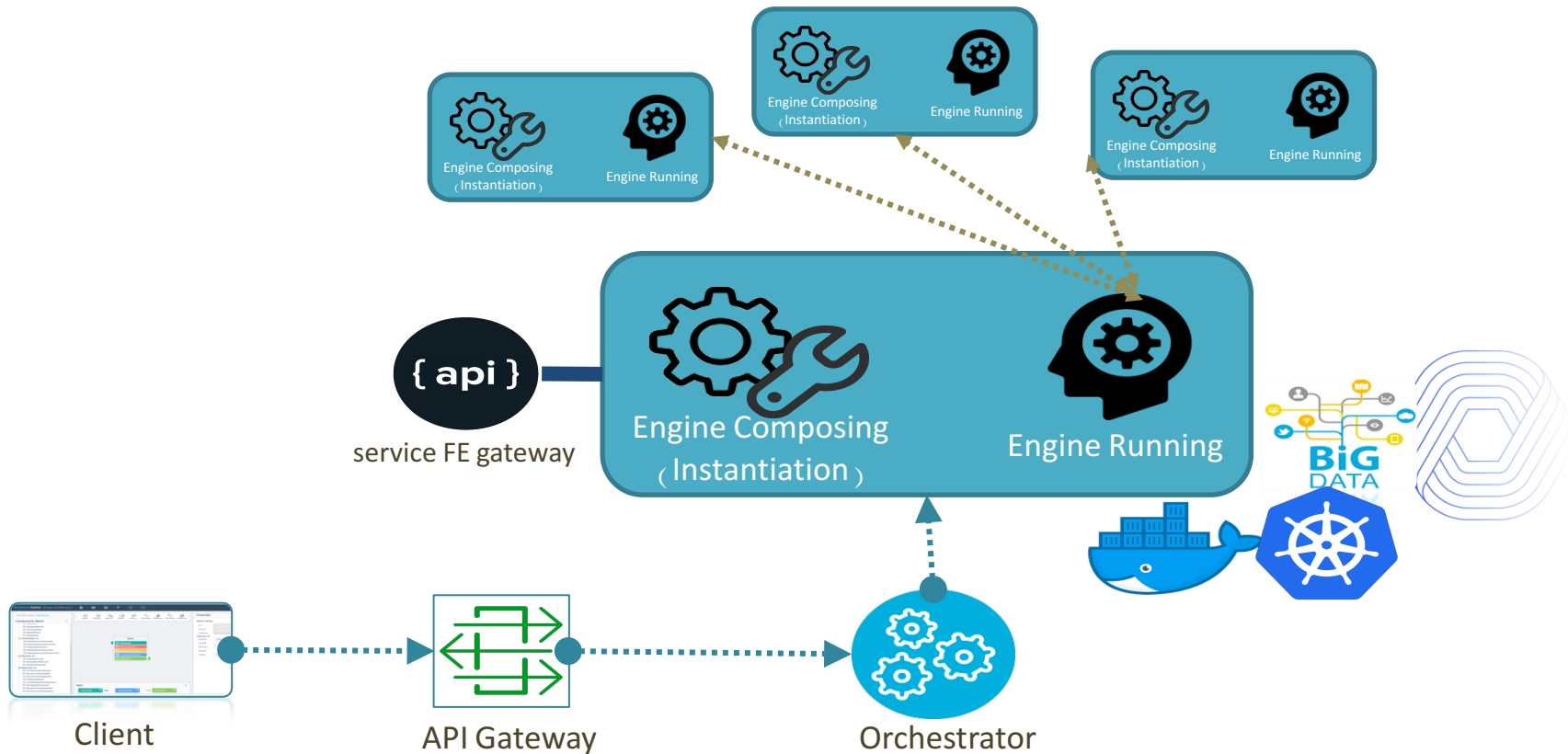
SOSCON 2018





How Serving Engine works

SOSCON 2018



KSB 오픈소스 프레임워크 확장하기

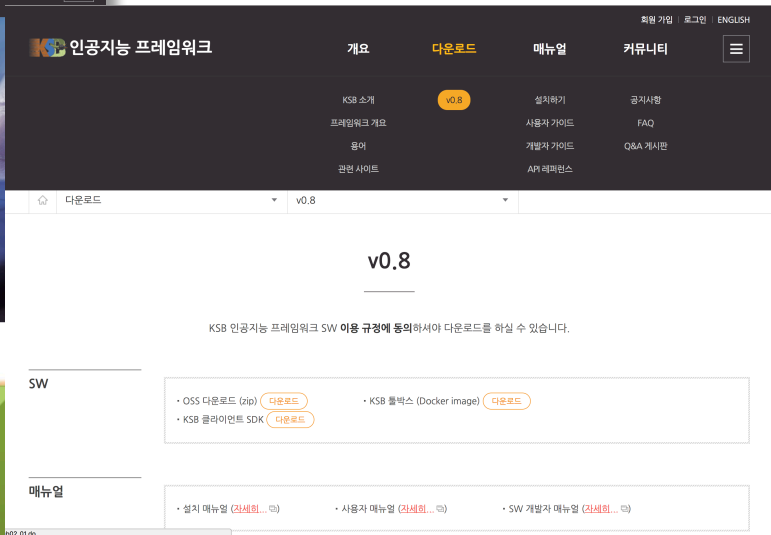
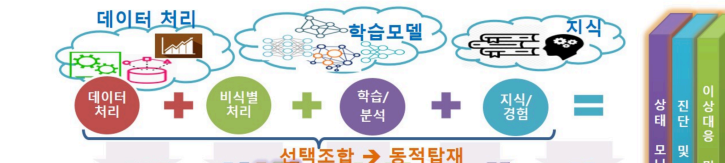
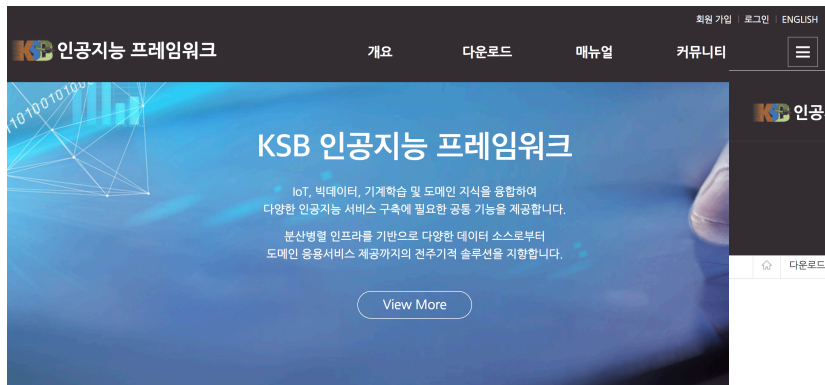
SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018

사용해 보기

SOSCON 2018

KSB 인공지능 프레임워크 홈페이지: <https://csleoss.etri.re.kr>





사용해 보기

SOSCON 2018

툴박스 vs 바이너리 설치 다운로드

KSB 인공지능 프레임워크를 빠르게 시작해 볼 수 있는 docker image 형태의 툴박스와 고객의 환경에 맞게 구성하여 설치할 수 있는 바이너리 설치 방식을 제공합니다.

KSB 툴박스 (docker image)	KSB 툴박스는 KSB의 새로운 기능을 배우고 개발하고 테스트하고 사용해 볼 수 있는 빠르고 간편한 개인 데스크톱 환경입니다. docker image 형태로 제공하는 KSB 툴박스 최신판을 이용하면 데이터 처리, Analytics, 모델 서빙 등을 위한 환경을 쉽고 편리하게 구성하여 빠르게 시작할 수 있습니다.	바로가기 ▶
KSB 인공지능 프레임워크 (소스코드 및 바이너리)	KSB 인공지능 프레임워크를 고객의 환경(클러스터 환경 등)에 맞게 설치하여 사용할 수 있습니다.	바로가기 ▶
KSB 웹툴킷	KSB 웹툴킷을 설치하여 KSB 인공지능 프레임워크를 웹 환경에서 사용할 수 있습니다.	바로가기 ▶

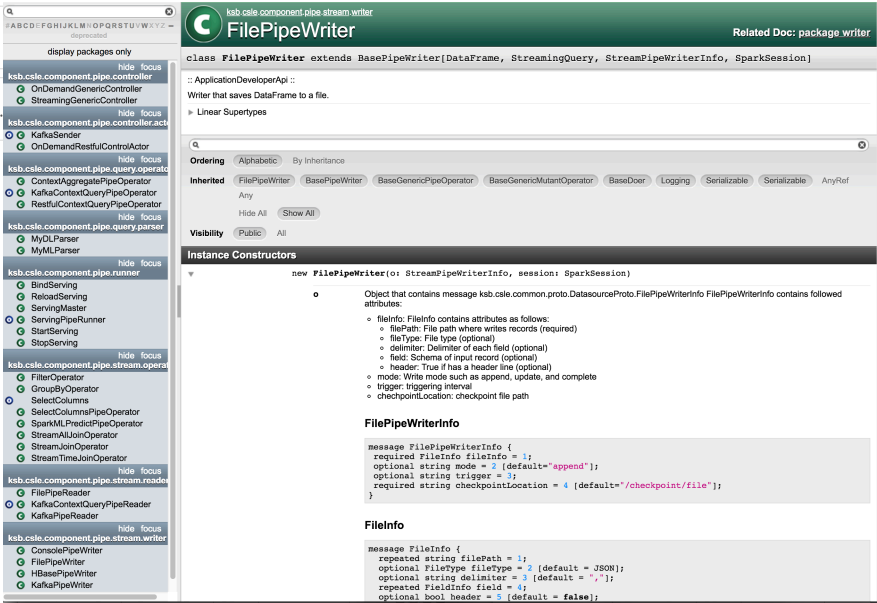




프레임워크 확장하기

SOSCON 2018

개발환경 구성	KSB 인공지능 프레임워크의 엔진 컴퍼넌트를 개발하고, scala client api를 이용한 워크플로우를 작성하거나 실행하기 위한 KSB Client 개발환경을 구성하는 절차를 설명합니다.	바로가기 ▶
컴퍼넌트 개발	KSB 인공지능 프레임워크가 제공하지 않는 신규 컴퍼넌트(Reader, Writer, Operator 등)를 개발하는 방법을 설명합니다.	바로가기 ▶
워크플로우 예제 개발	Eclipse 환경에서 워크플로우를 작성하여 실행하는 예제를 소개합니다.	
KSB 웹툴킷 컴퍼넌트 등록	개발한 컴퍼넌트를 KSB웹툴킷에서 이용할 수 있도록 등록하는 방법을 설명합니다.	



개발자 가이드



API 레퍼런스 문서



프레임워크 확장하기

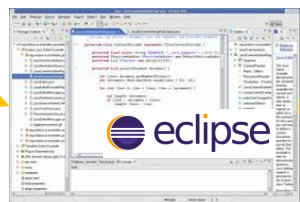
SOSCON 2018



SW engineer

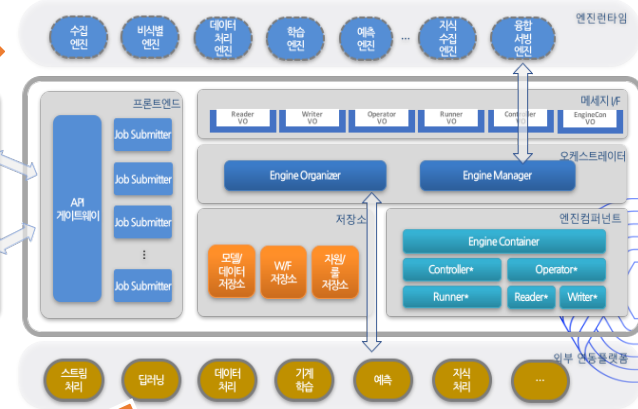
2. register UI component

3. version control of UI components



1. develop component (Msg & Operator)

commit



workflow spec with Version

5. dynamic binding according to version

4. version control of runtime components

build&deploy





프레임워크 확장하기

SOSCON 2018

KSB WebToolkit

Dashboard | Workflow Editor | Repository | Monitoring | Component | Management

Component 관리

Enum Component | Message Component | Engine Component | UI Component | Revision

Version wrap

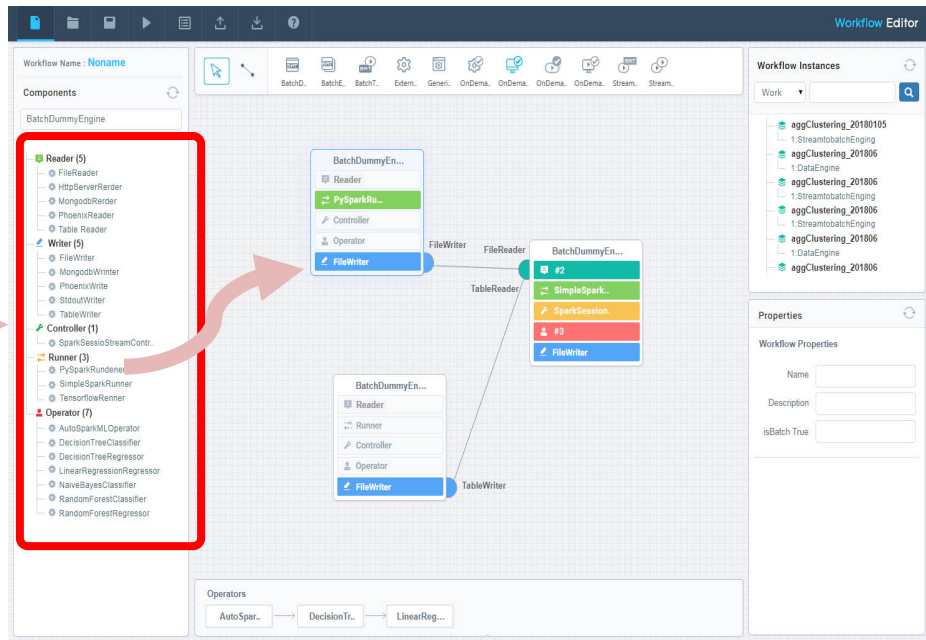
C	W	Component Version	KSB Verison	Description
	☒	0.8.2	0.8.2	
	☐	0.8	0.8.1	branch 0.8.1

Current Version

Component Version	KSB Verison	Description
0.8.2	0.8.2	branch 0.8.2

Working Version

Component Version	KSB Verison	Description
0.8.2	0.8.2	branch 0.8.2





KSB 프레임워크 SW 개발환경 구성

KSB 프레임워크 실행(테스트) 환경구
성

Scala IDE (개발도구) 설치

Ubuntu 예제:

Mac OSX 예제:

Java8 sdk 설치

Maven 빌드도구 설치

Ubuntu 예제:

Mac OSX 예제:

KSB 인공지능 프레임워크 오픈소스
설치

Eclipse 프로젝트 Import

KSB 프레임워크 SW 개발환경 구성

KSB 프레임워크를 이용한 신규 컴퍼넌트의 개발과 scala clientSDK를 이용한 워크플로우 실행 시나리오 작
성 및 실행을 요청하기 위한 개발환경 구성방법을 설명합니다.

KSB 프레임워크 SW 개발을 위한 환경을 구성하기 위한 절차는 다음과 같습니다.

- KSB 프레임워크 실행 및 테스트 환경구성 ([툴박스 설치](#) 참조)
- Scala IDE 설치
- Java8 sdk 설치
- Maven 빌드도구 설치
- ksb-oss_dist(KSB ClientSDK 포함) 설치
- Eclipse 프로젝트 Import
- Maven을 이용한 ksb-oss_dist(KSB Client SDK 포함) 프로젝트 빌드

KSB 프레임워크 실행(테스트) 환경구성

[KSB 툴박스 설치](#) 에서 제시하는 절차에 따라 KSB 프레임워크 실행(테스트) 환경을 구성합니다.

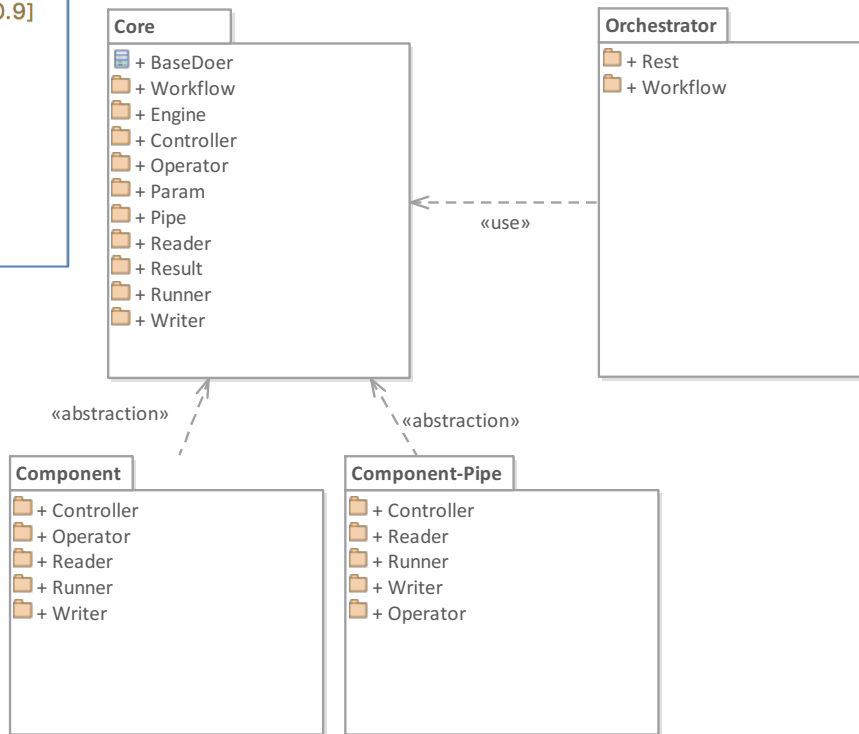
Scala IDE (개발도구) 설치

운영체제 (Ubuntu 또는 Mac osx)에 따라서 아래의 방법에 따라 Scala IDE를 설치합니다. 운영체제가 아래의



ksb-csle 프로젝트 구조

```
▼ > ksb-parent_2.11 [ksb-csle feature/oss-branch-0.9]
  ► > ksb-component-custom_2.11 [ksb-csle feature/oss-branch-0.9]
  ► > ksb-component-pipe_2.11 [ksb-csle feature/oss-branch-0.9]
  ► > ksb-components_2.11 [ksb-csle feature/oss-branch-0.9]
  ► > ksb-core-digest [ksb-csle feature/oss-branch-0.9]
  ► > ksb-didentification_2.11 [ksb-csle feature/oss-branch-0.9]
  ► > ksb-examples_2.11 [ksb-csle feature/oss-branch-0.9]
  ► > ksb-parent_2.11 [ksb-csle feature/oss-branch-0.9]
```





KSB 프레임워크 확장하기: 1) Protobuf Message 작성

SOSCON 2018

신규 데이터 처리 컴퍼넌트 (오퍼레이터) 만들기

컴퍼넌트용 Protocol Buffer 메시지 작성

코어 새로 빌드하기

신규 컴퍼넌트 스칼라 코드 작성

컴퍼넌트 빌드 및 프레임워크 배포

신규 데이터 처리 컴퍼넌트 (오퍼레이터)

데이터 처리 컴퍼넌트를 만들기 위한 작성 절차는 다음과 같습니다.

- 컴퍼넌트용 Protocol Buffer 메시지 작성
- 코어 새로 빌드
- 신규 컴퍼넌트 스칼라 코드 작성
- 컴퍼넌트 빌드 및 프레임워크 배포

아래에는 신규 컴퍼넌트를 개발하고 이를 프레임워크에 배포하여 실제 시스템 상에 배포하는 과정을 설명합니다. 자세한 설명을 위해서 *column select* (*ColumnSelectWithFile*) 개발을 대상으로 설명을 하도록 하겠습니다.

컴퍼넌트용 Protocol Buffer 메시지 작성

컴퍼넌트용 Protocol Buffer 메시지 작성

우선, 개발할 *ColumnSelectWithFile* 오퍼레이터에 전달될 파라미터를 Google의 Protocol buffer 메시지 방식에서 다운로드한 KSB 프레임워크 오픈소스를 *ScalaIDE*를 이용하여 프로젝트를 열어 **ksb-core-digest_2.11** (참고로, **ksb-core-digest_2.11** 모듈은 엔진을 생성하고 구동하기 위한 컴퍼넌트들의 파라미터를 정의한 Proto 모듈입니다.)

예제로 작성할 *ColumnSelectWithFileOperator* 오퍼레이터는 *table* 형태의 *dataframe*의 필드 중 특정 필드 고차 하는 필드명을 파라미터로 입력받는 대신 미리 작성해 둔 필드명을 정의한 파일을 읽어 해당 필드만 선택하여 *ksb.csle.common.proto* 패키지 내에 있는 *stream_operator_proto.proto* 파일을 열어 아래와 같이 *SelectColumnsWithFileInfo* 작성합니다.

```
syntax = "proto2";

package ksb.csle.common.proto;

message SelectColumnsWithFileInfo {
    required string columnIdPath = 1;
}
```

작성한 Protocol Buffer 메시지가 스트림 처리용으로 사용될 예정이므로 *StreamOperatorInfo* 내에 *oneof_operator* *SelectColumnsWithFileInfo* 유형의 *selectColumnsWithFile* 필드로 등록합니다.

```
message StreamOperatorInfo {
    required int32 id = 1;
    required int32 prevId = 2;
    required string clsName = 4;

    oneof oneof_operators {
        FilterInfo filter = 10;
        MissingValueImputationInfo missingValueImputation = 11;
        GroupbyInfo groupby = 12;
        ReshapeWithConcatInfo reshapeWithConcat = 13;
        PivotInfo pivot = 14;
        AddTimeIndexColumnInfo addTimeIndexColumn = 15;

        ...

        // for ColumnSelectWithFileOperator.
        SelectColumnsWithFileInfo selectColumnsWithFile = 800;
    }
}
```



1) Proto Message 작성

SOSCON 2018

```
message MySelectColumnsWithFileInfo {  
    required string columnIdPath = 1;  
}
```

stream_operator_proto.proto

```
message StreamOperatorInfo {  
    required int32 id = 1;  
    required int32 prevId = 2;  
    required string clsName = 4;
```

```
    oneof oneof_operators {  
        FilterInfo filter = 10;  
        MissingValueImputationInfo missingValueImputation = 11;  
        GroupbyInfo groupby = 12;  
        ReshapeWithConcatInfo reshapeWithConcat = 13;  
        PivotInfo pivot = 14;  
        AddTimeIndexColumnInfo addTimeIndexColumn = 15;
```

...

// for MyColumnSelectWithFileOperator.

```
MySelectColumnsWithFileInfo mySelectColumnsWithFile = 1000; }
```

stream_congtrol_proto.proto



KSB 프레임워크 확장하기: 2) 새로운 프레임워크 코어 빌드 생성

CON 2018

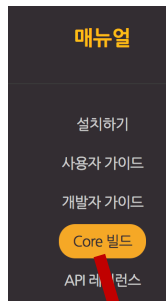
신규 데이터 처리 컴퍼넌트 (오퍼레이터) 만들기

컴퍼넌트용 Protocol Buffer 메시지 작성

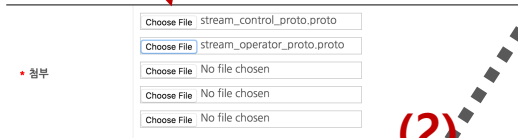
코어 새로 빌드하기

신규 컴퍼넌트 스칼라 코드 작성

컴퍼넌트 빌드 및 프레임워크 배포



(1)



(2)

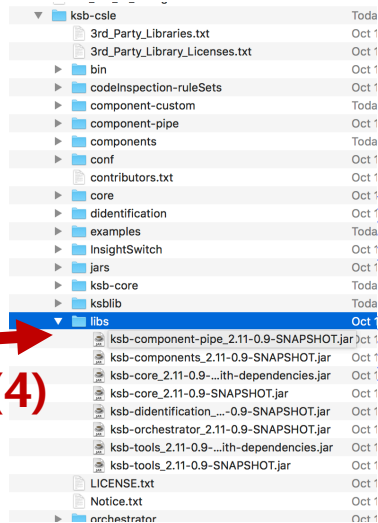
Core 빌드 요청



(3)

my@email.com

(4)





보낸사람csle.manager@gmail.com받은 날짜20

받는사람이연희

제목[KSB 인공지능 프레임워크] 요청하신 커스텀 jar 파일입니다

일반 첨부파일 - 1개(4MB)모두저장

ksb-core_2.11-0.8.1-SNAPSHOT (4MB)

안녕하세요,

요청하신 커스텀 jar 파일을 첨부하였습니다.
첨부파일 다운 후, 확장자를 반드시 .jar 로 변경해주세요.

커스텀 jar 생성을 위해 보내주신 proto 메시지는 아래와 같습니다.

```
### stream_control_proto.proto ###
-----
syntax = "proto2";

package ksb.csle.common.proto;

import "ksb/csle/common/proto/stream_operator_proto.proto";
import "ksb/csle/common/proto/stream_dident_proto.proto";
import "ksb/csle/common/proto/ondemand_operator_proto.proto";

message StreamOperatorInfo {
  required int32 id = 1;
  required int32 prevId = 2;
  required string clsName = 4;

  oneof operators {
```

(4) jar 확장자 붙인 후 복사



18



ksb-csle	Toda
3rd_Party_Libraries.txt	Oct 1
3rd_Party_Library_Licenses.txt	Oct 1
bin	Oct 1
codeInspection-ruleSets	Oct 1
component-custom	Toda
component-pipe	Oct 1
components	Toda
conf	Oct 1
contributors.txt	Oct 1
core	Oct 1
didentification	Oct 1
examples	Toda
InsightSwitch	Oct 1
jars	Oct 1
ksb-core	Toda
ksbllib	Toda
libs	Oct 1
ksb-component-pipe_2.11-0.9-SNAPSHOT.jar	Oct 1
ksb-components_2.11-0.9-SNAPSHOT.jar	Oct 1
ksb-core_2.11-0.9-...ith-dependencies.jar	Oct 1
ksb-core_2.11-0.9-SNAPSHOT.jar	Oct 1
ksb-didentification_...-0.9-SNAPSHOT.jar	Oct 1
ksb-orchestrator_2.11-0.9-SNAPSHOT.jar	Oct 1
ksb-tools_2.11-0.9-...ith-dependencies.jar	Oct 1
ksb-tools_2.11-0.9-SNAPSHOT.jar	Oct 1
LICENSE.txt	Oct 1
Notice.txt	Oct 1
orchestrator	Oct 1

프레임워크 확장하기: 3) 컴퍼넌트 개발

SOSCON 2018

신규 데이터 처리 컴퍼넌트 (오퍼레이터) 만들기

컴퍼넌트용 Protocol Buffer 메시지 작성

코어 새로 빌드하기

신규 컴퍼넌트 스칼라 코드 작성

컴퍼넌트 빌드 및 프레임워크 배포

(2)

(3)

(4)

신규 컴퍼넌트 스칼라 코드 작성

```
class ColumnSelectWithFileOperator(
  o: StreamOperatorInfo
) extends BaseDataOperator[StreamOperatorInfo, DataFrame](o) {

  private val p: SelectColumnsWithFileInfo = o.getSelectColumnsWithFileInfo
  private val path = p.getColumnIdPath
  private var columnNames: Array[String] = null

  private def selectColumns(df: DataFrame): DataFrame = {
    logger.debug(s"OpId ${o.getId} : SelectColumns")

    if (columnNames == null) {
      columnNames = df.sparkSession.sparkContext.textFile(path)
        .map(_.split(",")).collect.flatten
    }
    val result = df.select(columnNames.head, columnNames.tail:_* )

    logger.debug(result.show.toString)
    logger.debug(result.printSchema.toString)

    result
  }

  /**
   * Operates selectColumns.
   *
   * @param runner BaseRunner to run
   * @param df      Input dataframe
   * @return DataFrame Output dataframe
   */
  override def operate(df: DataFrame): DataFrame = selectColumns(df)
}
```

신규 데이터 처리 컴퍼넌트 (오퍼레이터) 만들기

컴퍼넌트용 Protocol Buffer 메시지 작성

코어 새로 빌드하기

신규 컴퍼넌트 스칼라 코드 작성

컴퍼넌트 빌드 및 프레임워크 배포

(4)

컴퍼넌트 빌드 및 프레임워크 배포

새로 작성한 메세제와 오퍼레이터가 포함된 바이너리를 KSB 프레임워크 실행바이너리가 있는 위치에 배포합니다. (레파지토리에 배포가 완료됨과 동시에 별도의 설치과정 없이 클라이언트의 요청 시 KSB 프레임워크에 의해 동적으로 클러스터 상의 노드들에 배포되어 엔진을 생성하고 구동하는데 적용이 됩니다.)

새로 배포해야할 KSB 프레임워크의 패키지는 아래와 같이 두 개의 jar에 해당합니다.

- **ksb-core_2.11-{version}.jar**

-- 신규메세지 포함된 core 바이너리로서 core 빌드 요청결과로 받은 파일

- **ksb-component-custom_2.11-{version}-jar-with-dependencies.jar**

-- 새로 작성한 컴퍼넌트 포함. 콘솔창에서 아래와 같이 maven 명령을 실행함으로써 위의 ksb-component-custom_2.11 빌드파일을 생성할 수 있습니다.

```
$ cd custom-components
$ mvn package install -DskipTests -Drat.skip=true
```

상기의 빌드된 패키지를 KSB 프레임워크 상에서 지정한 아카이브 디렉토리에 복사해서 넣어줌으로써 배포가 완료가 완료됩니다.

참고로, 아카이브 위치는 별도의 설정이 없다면 내부적으로 `${KSB_HOME}/jars/${ksb-vversion}` 디렉토리를 기본값으로 사용합니다.

프레임워크 확장하기: 4) 컴퍼넌트 사용하기

SOSCON 2018

워크플로우 작성 및 제출하기

```
object ColumnSelectionWithFileExample extends Logging {
  val appId: String = "Data-TrafficExample"

  def main(args: Array[String]) {
    val configArgs = ConfigArgs(appId = Option(appId))
    val workflowJson = ProtoUtils.msgToJson(workflow.asInstanceOf[Message])

    logger.info("configArgs:" + configArgs)
    logger.info("workflow: " + workflowJson)

    val client = new SimpleCsleClient("localhost", 19999)
    client.submit(workflowJson)
    client.close()
  }
}
```

```
// ClientSDK를 이용한 워크플로우 사양 작성
private def workflow = {
  val infileInfo = FileInfo.newBuilder()
    .addFilePath(
      s"file://${System.getProperty(
        "user.dir")}/input/input_columnSelection.csv"
    .replaceAll("\\\\", "/"))
    .setFileType(FileInfo.FileType.CSV)
    .setDelimiter(",")
    .setHeader(true)
    .build
  val reader = BatchReaderInfo.newBuilder()
    .setId(1)
    .setPrevId(0)
    .setClsName("ksb.csle.component.reader.FileReader")
    .setFileReader(infileInfo)
}
```

```
// 새로 개발한 ColumnSelectWithFileOperator를 이용하기 위하여 proto 메시지 인스턴스를 생성하고 이를 ColumnSelectWithFileOp
val selectedColumnsWithFileInfo = SelectColumnsWithFileInfo.newBuilder()
  .setColumnIdPath(
    s"file://${System.getProperty("user.dir")}/input/input_columnIds_columnSelection.csv")
  .build
val operator = StreamOperatorInfo.newBuilder()
  .setId(1)
  .setPrevId(0)
  .setClsName("ksb.csle.component.operator.reduction.ColumnSelectWithFileOperator")
  .setSelectColumnsWithFile(selectedColumnsWithFileInfo)
  .build
```

```
val outfileInfo = FileInfo.newBuilder()
  .addFilePath(
    s"file://${System.getProperty("user.dir")}/output/output_columnSelectWithFile.csv"
```

```
// 새로 개발한 ColumnSelectWithFileOperator를 이용하기 위하여 proto 메시지 인스턴스를 생성하고 이를 ColumnSe
val selectedColumnsWithFileInfo = SelectColumnsWithFileInfo.newBuilder()
  .setColumnIdPath(
    s"file://${System.getProperty("user.dir")}/input/input_columnIds_columnSelection.csv")
  .build
val operator = StreamOperatorInfo.newBuilder()
  .setId(1)
  .setPrevId(0)
  .setClsName("ksb.csle.component.operator.reduction.ColumnSelectWithFileOperator")
  .setSelectColumnsWithFile(selectedColumnsWithFileInfo)
  .build
```



프레임워크 확장하기: 웹툰킷 컴퍼넌트로 등록

SOSCON 2018

웹툰킷 컴퍼넌트 버전관리/리비전

Component 관리

Enum ComponentMessage ComponentEngine ComponentUI ComponentRevision

Version Map

C	W	Message Version	KSB Verison	Description	
	☐	0.8	0.8.1	branch 0.8	
	☐	0.7	0.8	branch 0.8	
	☒	0.6	0.8	test	

Current Version

Message Version	KSB Verison	Description
0.8	0.8.1	branch 0.8

Working Version

Message Version	KSB Verison	Description
0.6	0.8	test

Current version 선택

Working version 선택

복제 버튼





프레임워크 확장하기: 웹툰킷 컴퍼넌트로 등록

SOSCON 2018

Component 관리

Enum Component Message Component Engine Component UI Component Revision

MsgComponent (msg: 0.6, ksb: 0.8) 7

Component [New MsgComponent] 8 9 Name 6 Enter a name of MsgComponent! 1

Required	FieldType	FieldComponent	FieldName	Order	DefaultValue
				0	

Help 2

Required	FieldType	FieldComponent	FieldName	Order	DefaultValue

MsgComponentField 정보가 존재하지 않습니다. 3

OneofGroups

Name 4 Enter a name of OneofGroup!

Name MsgComponent 5 FieldName Order

OneofGroup 정보가 존재하지 않습니다.

EnumComponent

Name 5 Enter a name of EnumComponent!

Name FieldName Order

EnumComponent 정보가 존재하지 않습니다.

MsgComponent Help

Short Help

Help

Message 컴퍼넌트 등록

엔진 컨테이너에 매핑하기

Enum Component Message Component Engine Component UI Component Revision

UIComponent (msg: 0.6, ksb: 0.8) DB +

UIComponent [New 'UIComponent'] DB + MsgComponent DB + DB +

Type Reader DB + Class Name ksb.csle.component.pipe.stream.reader.FilePi

Nickname FilePipeReader Description 파일 스트림 입력기

UIComponent Mapping DB +

UIComponent [Select 'UIComponent'] DB + Engine [Select 'UIEngineComponent'] DB +

Mapping Entries DB +

MsgVersion	Engine	Type	Nickname
추가된 매핑 정보가 존재하지 않습니다.			

Mapping Relations

MsgVersion	UIComponent	MsgComponent	Engine	Type	Field
매핑 정보가 존재하지 않습니다.					





프레임워크 확장하기: 웹툴킷 컴퍼넌트로 등록

SOSCON 2018

엔진에 매핑하기

Enum Component Message Component Engine Component **UI Component** Revision

UIComponent (msg: 0.6, ksb: 0.8)

UIComponent	[Reader] FilePipeReader	MsgComponent	FilePipeReaderInfo [Q]
Type	Reader	Class Name	ksb.csle.component.pipe.stream.reader.FilePipeReader
Nickname	FilePipeReader	Description	파일 스트림 입력기

UIComponent Mapping

1 선택

3

4

Field Name	MsgComponent	MsgVersion
filePipeReader	FilePipeReaderInfo [Q]	
kafkaPipeReader	KafkaPipeReaderInfo [Q]	0.6
filePipeReader	FilePipeReaderInfo [Q]	

2 Oneof 입력

Mapping Entries

MsgVersion	Engine
0.6	StreamJoinEngine [Q]

Workflow Editor * Msg: 0.8 [KSB: 0.8]

Workflow Name: Noname

Components :: StreamJoin

- Reader (2)
 - FilePipeReader
 - 파일 스트림 입력기
- Writer (2)
 - ConsolePipeWriter
 - FilePipeWriter
- Controller (1)
 - StreamingGenericControlle..
- Runner (1)
 - SimpleSparkRunner
- Operator (9)
 - ColumnOperationOperator
 - ColumnsRenamer
 - FilterOperator
 - GroupByOperator
 - SelectColumnsPipeOperator
 - SparkMLPredictor
 - StreamAllJoinOperator
 - StreamJoinOperator
 - TimeAggregator

Nickname

Description

StreamJoin

FilePipeRe..

Workflow instances

BatchAutoML

EnergyMDLJoinSplitExample

Properties

StreamJoin::Reader

1 FilePipeReader

*id 0

*prevId 0

*cbName ksb.csle.orchestrator.pipe.stream.reader.FileP

FilePipeReader

*path

*fileType JSON

maxFilesPerTrigger 0

*addTimestamp false

timeColName timestamp

watermark 1 seconds

field 세부 설정 내용...

delimiter ,

header false



마무리

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018





프레임워크 주요 특징

SOSCON 2018

- **(활용성) 일원화**
 - Unified Data and AI manipulation
 - real-time streaming to streaming predictive model serving with continuous model updates
- **(사용성) 사용 편의성**
 - DIY Workflow Editor
 - Develop and deploy in the same place
- **(확장성) 모듈러식 구성**
 - Accumulate SW component → IT technology accumulation
 - Build Predictive API using ML Model → ML/DL-based Predictive API accumulation
 - Converged API Fabric → Various domain Intelligent service





마무리

SOSCON 2018

- 릴리즈 계획
 - 2018년 7월 3일 v0.8 release
 - 2018년 10월 v1.0 release
 - <https://csleoss.etri.re.kr> 통해 배포예정
- 향후계획
 - 다양한 환경(edge/cloud) 배포
 - 이들간의 orchestration, 보안 이슈 등 해결



THANK YOU

SOSCON 2018

SAMSUNG OPEN SOURCE CONFERENCE 2018

