

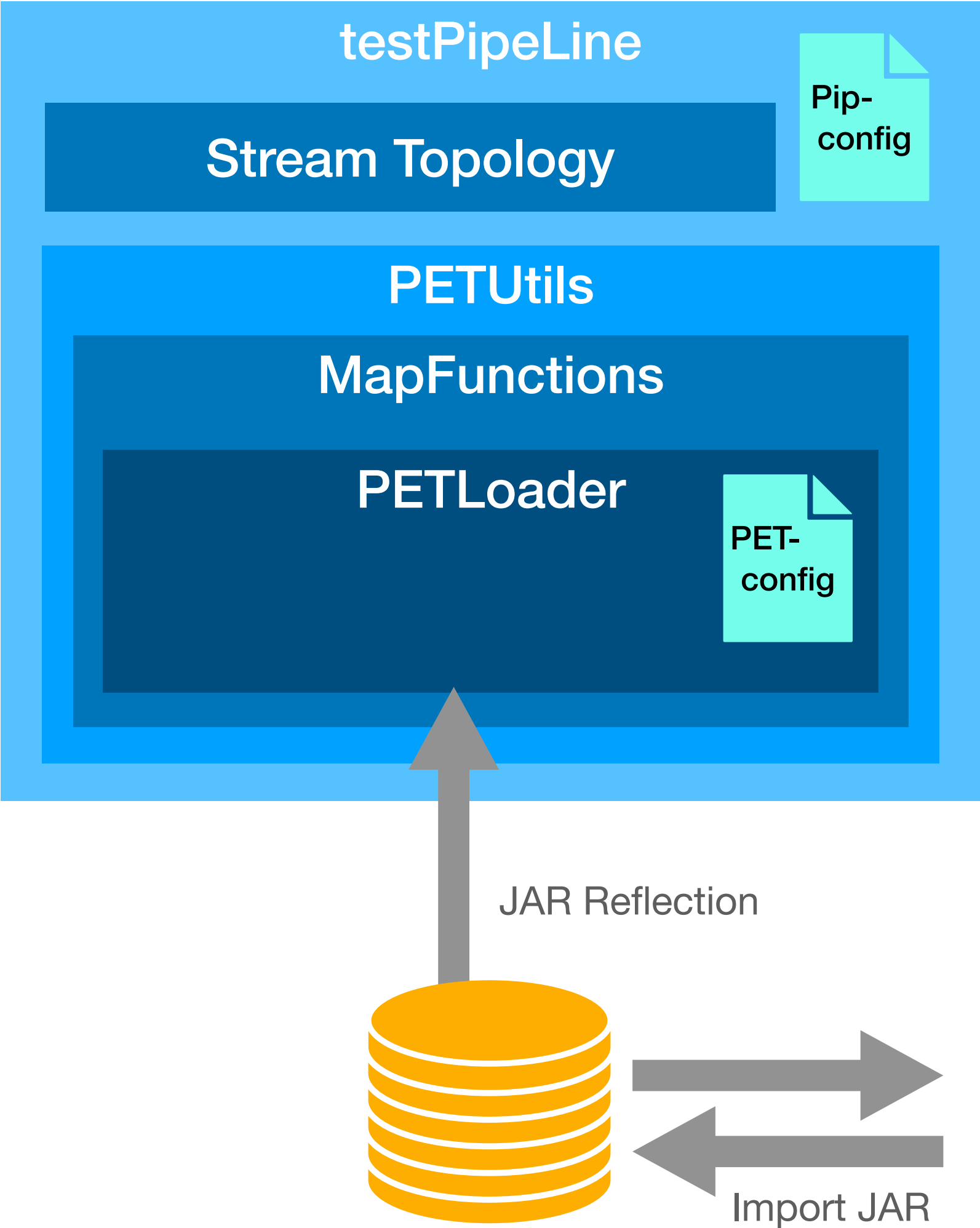
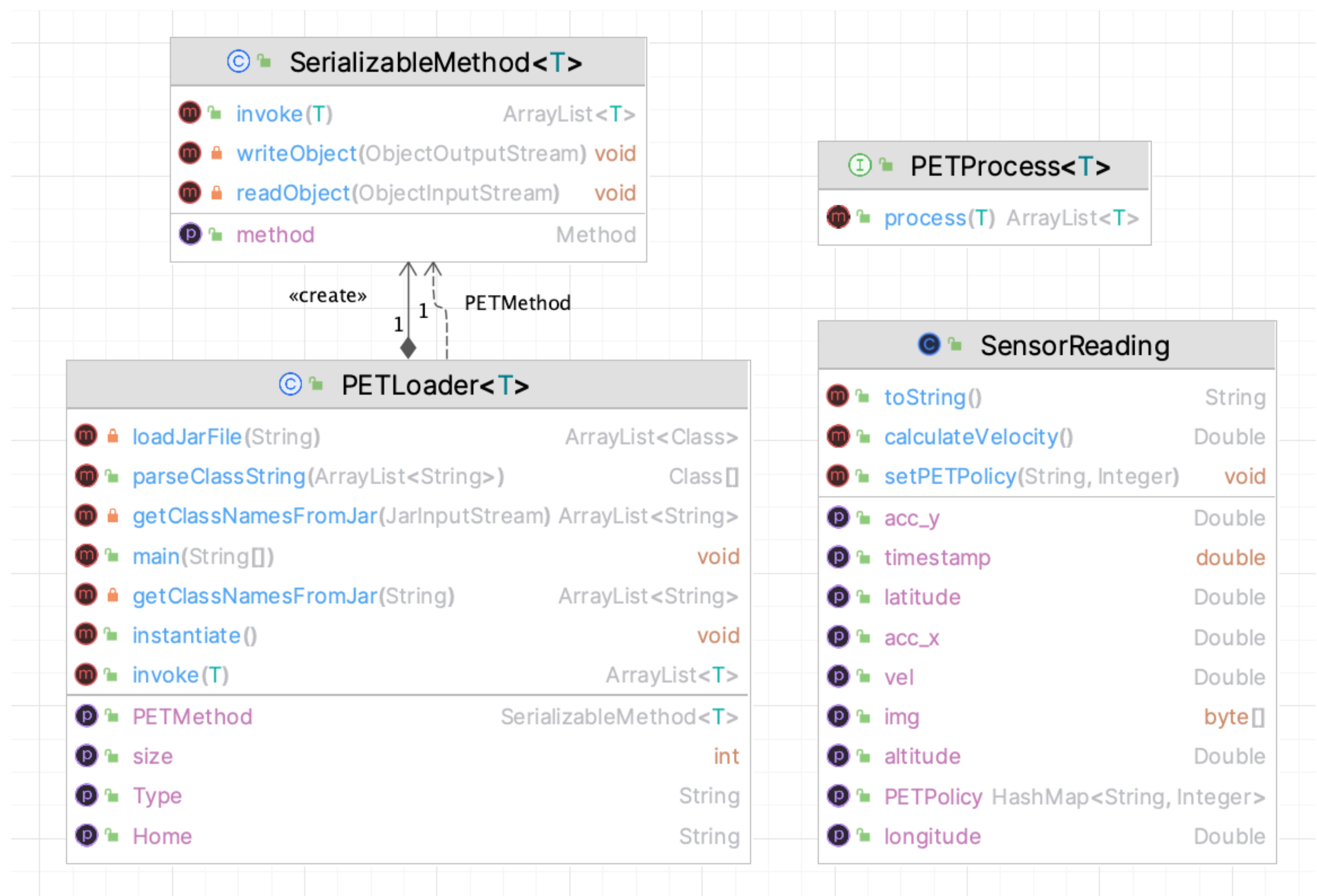
Weekly Report 03

Practical Information Course

09. Juni. 2023

Implementation

Dynamic loading of JAR Package



Implementation

Configuration

PipeConfig

```
{  
  "PET-CONF": "/Users/lukasye/Projects/pis_project_ss2023_group2/carPET/config/PETconfig.json",  
  "PET-TYPE": ["SPEED"],  
  "USER-CONF": ""  
}
```

```
{  
  "HOMEDIR": "carPET/",  
  "SPEED": {...},  
  "IMAGE": {  
    "0": {  
      "FileName": "ImagePET01-1.0-SNAPSHOT.jar",  
      "Description": "",  
      "FunctionName": "pis.group2.ImageAnonymizer",  
      "ConstructorParameter": ["java.lang.String", "java.lang.Integer"],  
      "Default": ["/foo/bar/src/main/resources/enet/Main.py", 1],  
      "FunctionParameter": ["java.lang.Object"]  
    },  
    "Location": {...}  
  }  
}
```

PETConfig

Implementation

PET

```
***** Speed PET Testing *****  
Totally 2 PETs  
Input: 20.3  
PET 0: [0.0]  
PET 1: [75.0]
```

Input:



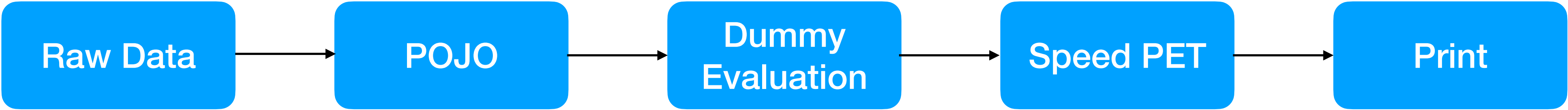
Output:



```
***** Location PET Testing *****  
Totally 2 PETs  
Input: (48.985771846331,8.3941997039792)  
PET 0: [(48.985771846331,8.3941997039792)]  
PET 1: [(48.985871846331,8.3937997039792),  
        (48.985771846331,8.394599703979202),  
        (48.985871846331,8.393999703979201)]
```

Implementation

Result



resources

enet

enet-classes.txt

enet-colors.txt

enet-model.net

Main.py

PIS_data

data_preprocessing.py

gps_info.csv

result

testImage

byteString

field.jpg

test_image_1.jpg

test_image_4.jpg

57

58

59

60

61

62

63

64

65

66

67

68

69

70

71

72

DataStreamSource<String> inputStream = env.readTextFile(filePath);

SingleOutputStreamOperator<SensorReading> sensorStream = inputStream.map(new PETUtils.toSensorReading());

SingleOutputStreamOperator<SensorReading> evaluatedStream = sensorStream.map(new PETUtils.evaluationData());

SingleOutputStreamOperator<SensorReading> speedStream = evaluatedStream.map(
new PETUtils.applyPET(PETconfpath, Type: "SPEED"));

speedStream.print();

}

1 usage ± Lukasye

public void execute() throws Exception {

env.execute();

}

Run

testPipeline

8> SensorReading{PETPolicy={SPEED=0, LOCATION=0, IMAGE=0}, timestamp=1.317382841993512E9, latitude=48.985750517037, longitude=8.3943711077881, altitude=116.34136199951, acc_x=0.98966518209079, acc_y=-0.82304008733744, vel=0.0}

3> SensorReading{PETPolicy={SPEED=0, LOCATION=0, IMAGE=0}, timestamp=1.317382885212577E9, latitude=48.986473134582, longitude=8.3945023103592, altitude=116.6439666748, acc_x=0.18995096756906, acc_y=0.10021523980011, vel=0.0}

6> SensorReading{PETPolicy={SPEED=0, LOCATION=0, IMAGE=0}, timestamp=1.317382928251775E9, latitude=48.985337413869, longitude=8.3933672665269, altitude=116.42433166504, acc_x=-0.71251445644902, acc_y=0.12404831030857, vel=0.0}

PET changed!

PET changed!

3> SensorReading{PETPolicy={SPEED=0, LOCATION=0, IMAGE=0}, timestamp=1.317382885322565E9, latitude=48.986477517916, longitude=8.3944884922673, altitude=116.64752960205, acc_x=0.047842743174243, acc_y=-0.054382616729407, vel=0.0}

PET changed!

1> SensorReading{PETPolicy={SPEED=1, LOCATION=0, IMAGE=0}, timestamp=1.317382856533203E9, latitude=48.985445395341, longitude=8.3957207394869, altitude=116.33379364014, acc_x=-2.0086495775827, acc_y=0.71037378223876, vel=75.0}

5> SensorReading{PETPolicy={SPEED=1, LOCATION=0, IMAGE=0}, timestamp=1.317382914012063E9, latitude=48.985943402851, longitude=8.3930166878252, altitude=116.13272094727, acc_x=-0.29284027451224, acc_y=0.46677138937355, vel=75.0}

8> SensorReading{PETPolicy={SPEED=1, LOCATION=0, IMAGE=0}, timestamp=1.317382842093627E9, latitude=48.985747770771, longitude=8.3943801375449, altitude=116.32946777344, acc_x=1.099448301002, acc_y=-0.74728543414053, vel=75.0}

2> SensorReading{PETPolicy={SPEED=1, LOCATION=0, IMAGE=0}, timestamp=1.317382870973007E9, latitude=48.986265283849, longitude=8.3956910365068, altitude=116.39036560059, acc_x=-1.3455183159221, acc_y=0.35271154934794, vel=75.0}

3> SensorReading{PETPolicy={SPEED=1, LOCATION=0, IMAGE=0}, timestamp=1.317382885422833E9, latitude=48.986481516988, longitude=8.3944759386502, altitude=116.64939117432, acc_x=-0.16023140567171, acc_y=0.20887586391829, vel=75.0}

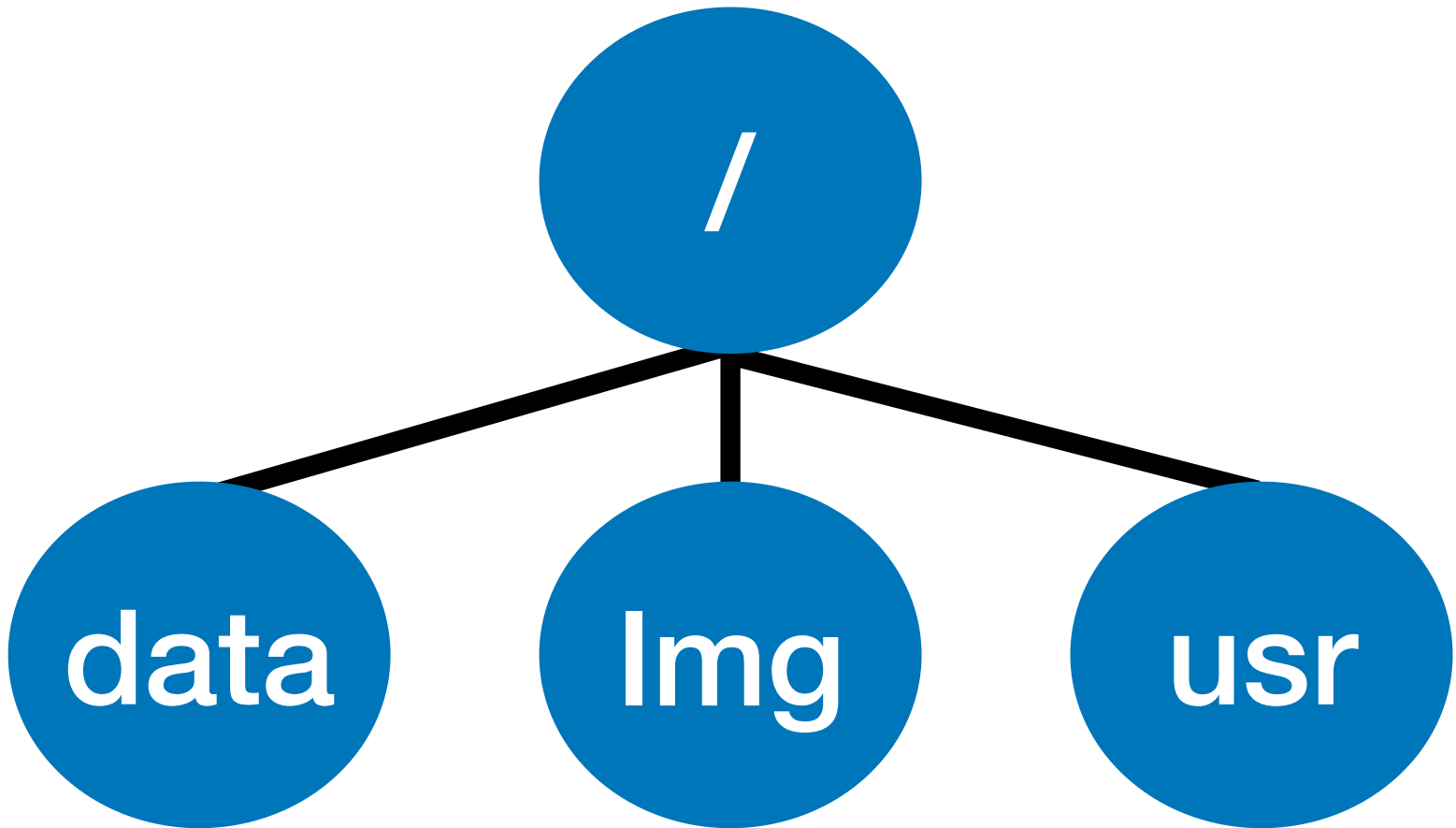
6> SensorReading{PETPolicy={SPEED=1, LOCATION=0, IMAGE=0}, timestamp=1.317382928351761E9, latitude=48.985335118571, longitude=8.3933739458418, altitude=116.42562866211, acc_x=-0.84559385622486, acc_y=0.18975664948666, vel=75.0}

carPET > src > main > java > pis > group2 > algorithm > testPipeline > buildPipeline

Implementation

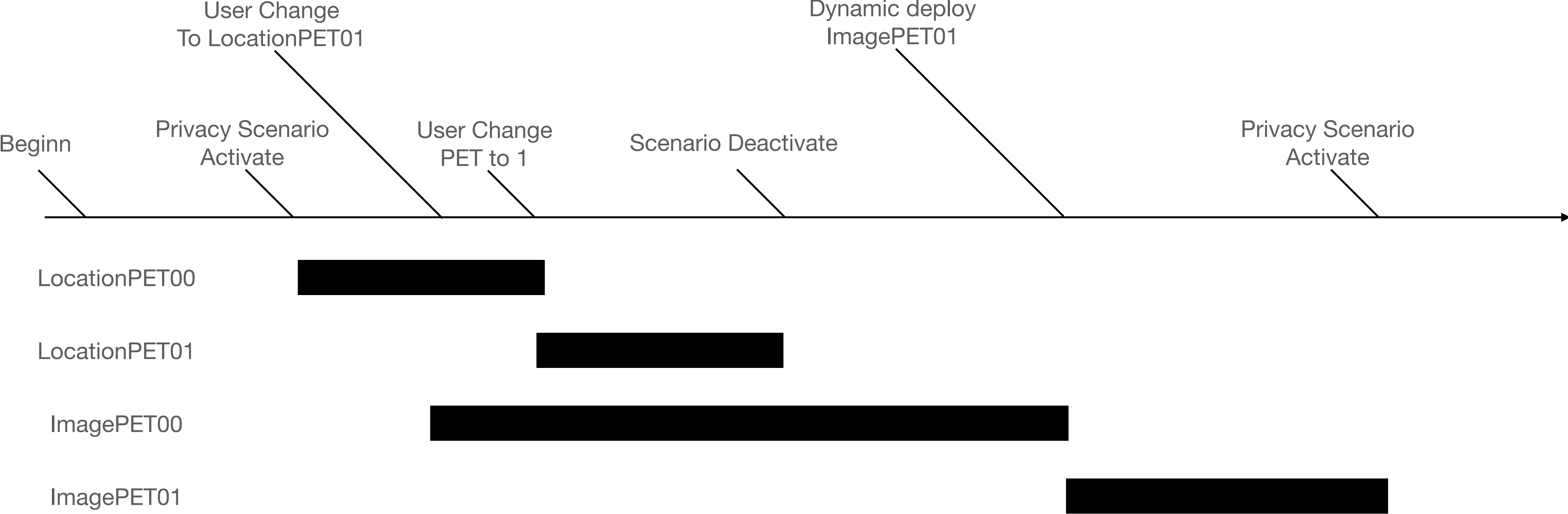
Data source generator

Id	Command	Parameter	Describe
1	Run	—num X	Feed X lines of data(image) to Kafka
2	Write	— topic X —msg Y	Write Customer message to topic Y
3	RunDelay	—num X —delay Y	Feed X lines to Kafka with delay Y
4	Foo	—env X	Manually activate scenario



Senario

Demo



Target for next phase

- Dynamically deployment of package in field
- Mechanism for benchmark
- Documentation
- Generalise data source generator
- Design GUI for demonstration