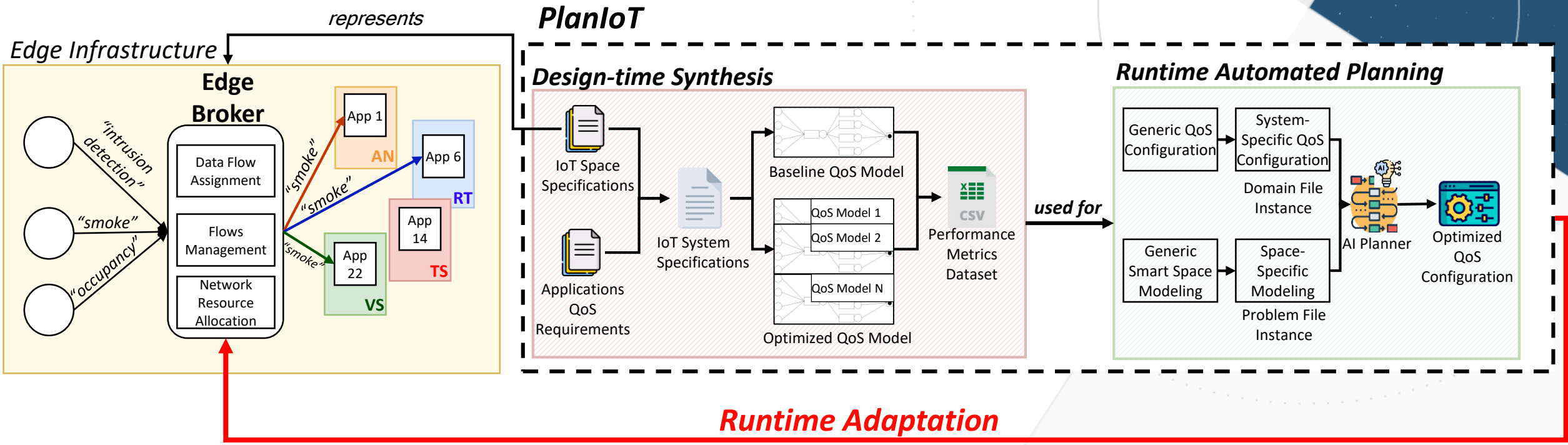
Artifact: Implementation of an Adaptive Flow Management Framework for IoT Spaces

Houssam Hajj Hassan, Georgios Bouloukakis,
Ajay Kattepur, Denis Conan, Djamel Belaïd

Télécom SudParis, IP Paris, France
Ericsson AI Research, India

The PlanIoT Approach

High-level Overview



The PlanIoT Implementation

Plan Generation Process

JSON

IoT system specification

```
 "iotDevices": [
{
     "deviceId": "temp_r324",
     "deviceName": "temperature_sensor",
     "publishFrequency": 5,
     "messageSize": 200,
     "publishesTo": ["temperature_r324"],
     "distribution": "exponential"
},
…
```

IoT devices definition

```
 "applications": [
{
     "applicationId": "app1",
     "applicationName": "dashboard",
     "applicationCategory" : "AN",
     "priority": 0,
     "subscribesTo": ["temperature_r324",
     "smoke_r324"]
},
…
```

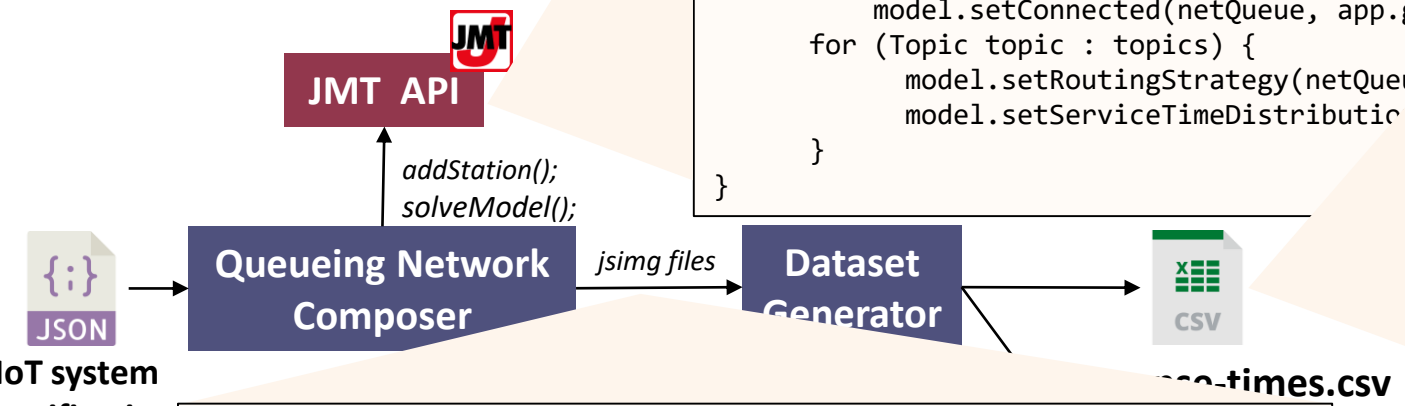
Applications definition

```
"systemBandwidth": 70,
"bandwidthPolicy": "default",
"priorityPolicy": "apps",
"dropRateAN": 0,
"dropRateRT": 0,
"dropRateTS": 0,
"dropRateVS": 0,
"brokerCapacity": 10000
```

IoT system parameters

The PlanIoT Implementation

Plan Generation Process




```
public static void composeQueueingNetwork() {
      CommonModel jmtModel = new CommonModel();
      inQueue = model.addStation("inQueue",CommonConstants.STATION_TYPE_SERVER);
      netQueue = model.addStation("netQueue",CommonConstants.STATION_TYPE_SERVER);
      model.setStationQueueStrategy(netQueue,model.STATION_QUEUE_STRATEGY_NON_PREEMPTIVE_PRIORITY);
      for (Application app : applications)
          model.setConnected(netQueue, app.getQueue(), true);
      for (Topic topic : topics) {
            model.setRoutingStrategy(netQueue,topic.getClass(),probRouting);
            model.setServiceTimeDistributic
      }
}
```

topic	app	default	prioRT	prioVS	prioRTVSTSAN	dropVS1	dropVS2AN2	...
topic1	app1	0.899	1.224	1.136	0.036	0.886	0.927	...
topic10	app11	0.692	0.773	0.854	0.0162	0.684	0.661	...
topic10	app9	0.699	0.756	0.855	0.015	0.671	0.658	...
topic11	app12	0.798	0.977	1.084	0.024	0.789	0.776	...
topic12	app9	0.800	0.882	0.980	0.017	0.802	0.798	...
topic13	app10	0.792	0.890	0.978	0.0197	0.789	0.795	...

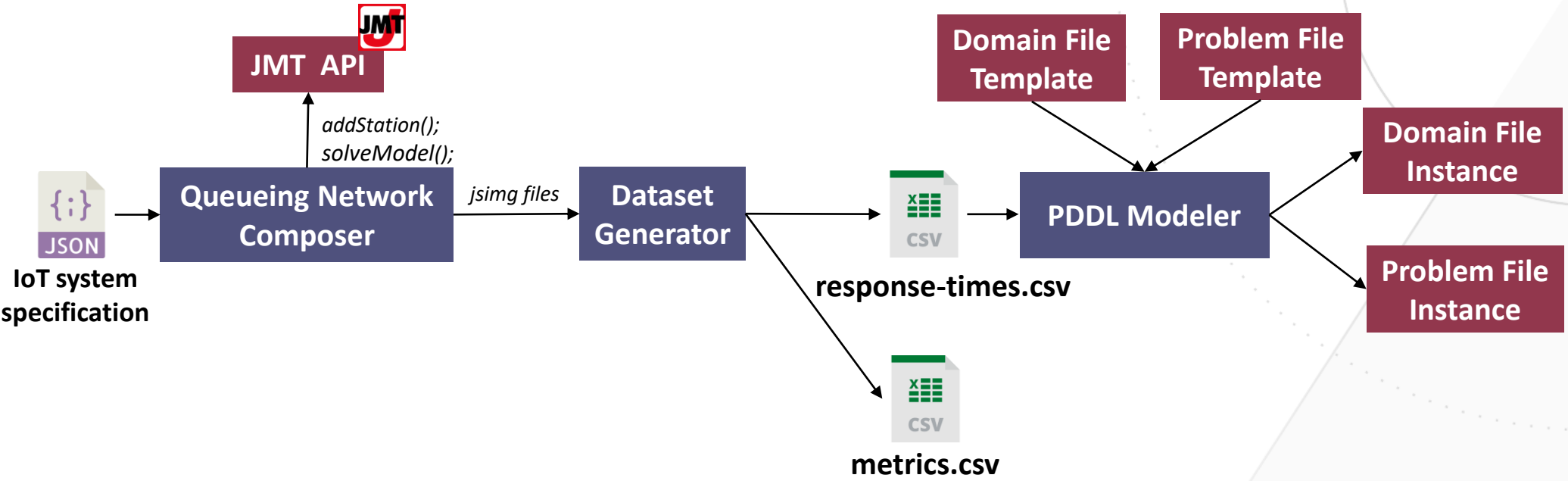
-times.csv

```
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
logDecimalSeparator="." logDelimiter=","
logPath="your_local_path\JMT\" modelDefinitionPath="."
modelName="default.jsimg" solutionMethod="simulation"
xsi:noNamespaceSchemaLocation="SIMmodeloutput.xsd">    <measure
alfa="0.05" analyzedSamples="8194" class="" discardedSamples="65"
logFile="your_local_path/JMT/networkQueue_Utilization.csv"
lowerLimit="0.40003963524075425" maxSamples="1000000"
meanValue="0.42805882812360384" measureType="Utilization"
nodeType="station" precision="0.05" station="outputQueue"
successful="true" upperLimit="0.4560780210064534"/>
```

topic	app	response_time	throughput	drop_rate
topic10	app11	0.6927146474811501	0.003779867027741613	0.0
topic10	app9	0.6996978145171252	0.003779867027741613	0.0
topic11	app12	0.7982395522950397	0.1172509965121246	0.0
topic11	app15	0.799657539567181	0.1172509965121246	0.0
topic11	app8	0.7977678454742368	0.1172509965121246	0.0
topic12	app10	0.7950121622847551	0.003110639907232652	0.0

The PlanIoT Implementation

Plan Generation Process



The PlanIoT Implementation

PDDL Files Generation

Planning problems are expressed using the Planning Domain Definition Language (PDDL), an action centered language that provides a standard syntax to describe actions by their parameters, preconditions, and effects. A plan consists of two descriptions: a *domain* file and a *problem* file.

response-times.csv

```
(:action prioritize_RT
   :parameters (?t  - topic ?app - application)
   :precondition (and (priority_not_set ?t  ?app))
   :effect (and (priority_not_set ?t  ?app)
                (priority_set ?t  ?app)
                 #prioRT_effects#))
(:action droppingVS1
   :parameters (?t  - topic ?app - application)
   :precondition (and (baseline ?t  ?app))
   :effect (and (not (baseline ?t  ?app))
                (QoS_achieved?t  ?app)
                 #dropVS1_effects# ))
```

PDDL Domain Template

```
(:action prioritize_RT
   :parameters (?t  - topic ?app - application)
   :precondition (and (priority_not_set ?t  ?app))
   :effect (and (priority_not_set ?t  ?app)
                (priority_set ?t  ?app)
                   (increase (latency intrusiondetection app2) 0.62)
                   (increase (latency intrusiondetection app21) 0.27)
…))
(:action droppingVS1
   :parameters (?t  - topic ?app - application)
   :precondition (and (baseline ?t  ?app))
   :effect (and (not (baseline ?t  ?app))
                (QoS_achieved?t  ?app)
                   (increase (latency firedetection app22) 0.2)
                   (increase (latency amazonecho app21) 0.13)
…))
```

PDDL Domain Instance

The PlanIoT Implementation

PDDL Files Generation

response-times.csv

```
(:action prioritize_RT
   :parameters (?t  - topic ?app - application)
   :precondition (and (priority_not_set ?t  ?app))
   :effect (and (priority_not_set ?t  ?app)
                (priority_set ?t  ?app)
                 #prioRT_effects#))
(:action droppingVS1
   :parameters (?t  - topic ?app - application)
   :precondition (and (baseline ?t  ?app))
   :effect (and (not (baseline ?t  ?app))
                (QoS_achieved?t  ?app)
                 #dropVS1_effects# ))
```

PDDL Domain Template

```
(:action prioritize_RT
   :parameters (?t  - topic ?app - application)
   :precondition (and (priority_not_set ?t  ?app))
   :effect (and (priority_not_set ?t  ?app)
                (priority_set ?t  ?app)
                (increase (latency intrusiondetection app2) 0.62)
                (increase (latency intrusiondetection app21) 0.27)
                ...))
(:action droppingVS1
   :parameters (?t  - topic ?app - application)
   :precondition (and (baseline ?t  ?app))
   :effect (and (not (baseline ?t  ?app))
                (QoS_achieved?t  ?app)
                (increase (latency firedetection app22) 0.2)
                (increase (latency amazonecho app21) 0.13)
           ...))
```

PDDL Domain Instance

response-times.csv

IoT system specification

```
(:objects
   #topics# topic_all  - Topic
   #apps# app_all - Application)
(:init
      (baseline topic_all app_all)
      (priority_not_set topic_all app_all)
      #init_predicates#)
(:goal (and  (QoS_achieved topic_all app_all)
      (priority_set topic_all app_all)))
(:metric minimize #metric#)
```

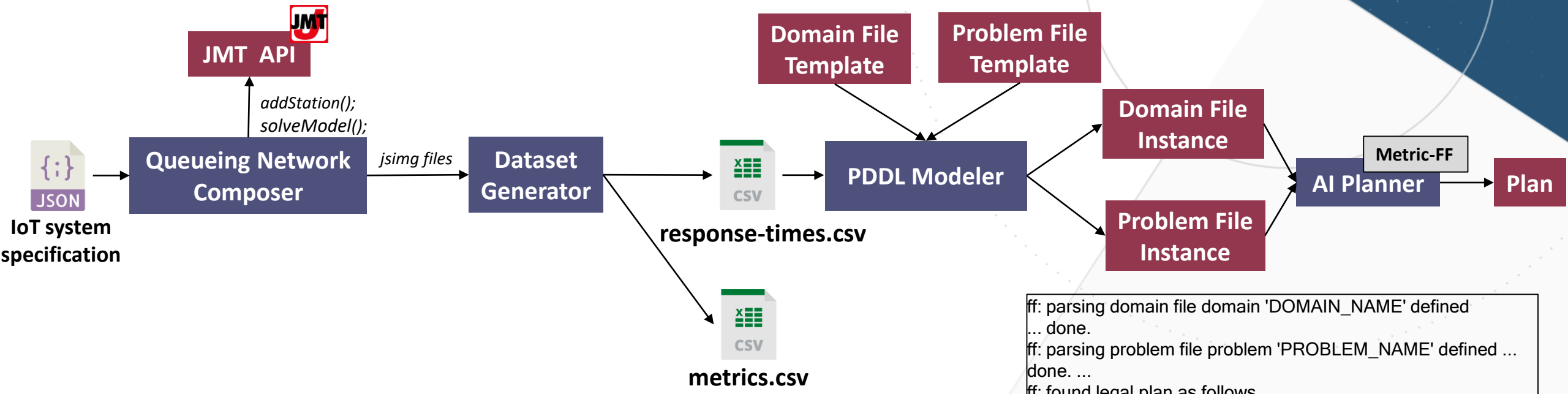
PDDL Problem Template

```
(:objects
   printing energymanagement ... topic_all  - Topic
   app1 app2 app3 app4 ... app_all - Application)
(:init
      (baseline topic_all app_all)
      (priority_not_set topic_all app_all)
      (= (latency intrusiondetection app2) 0)
      (= (latency intrusiondetection app21) 0))
(:goal (and  (QoS_achieved topic_all app_all)
      (priority_set topic_all app_all)))
(:metric minimize ( + ( + (* 1 (latency videosurveillance app5 )))
                      (* 1 (latency energymanagement app2 )))
          ...)
```

PDDL Problem Instance

The PlanIoT Implementation

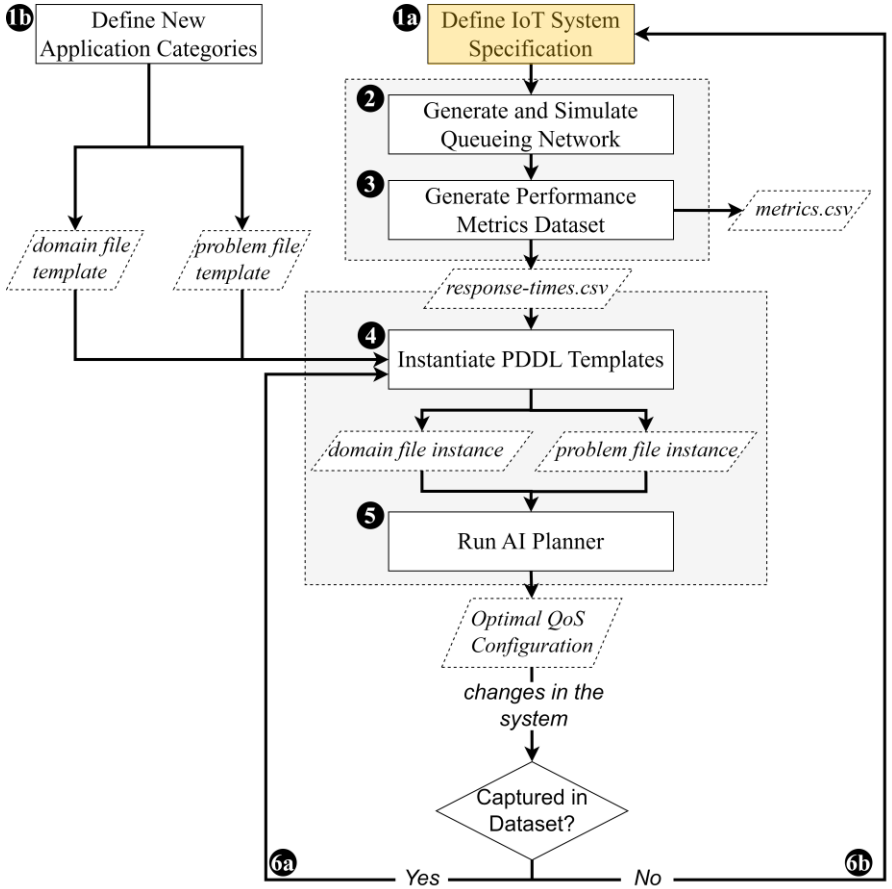
Plan Generation Process




```
ff: parsing domain file domain 'DOMAIN_NAME' defined
... done.
ff: parsing problem file problem 'PROBLEM_NAME' defined ...
done. ...
ff: found legal plan as follows
step    0: DROPPINGVS2AN2 TOPIC_ALL APP_ALL
        1: PRIORITIZE_RT_VS_TS_AN TOPIC_ALL APP_ALL
plan cost: 59.439987
```

Using PlanIoT

Defining IoT System Specifications




```
 "iotDevices": [
{    "deviceId": "temp_r324",
     "deviceName": "temperature_sensor",
     "publishFrequency": 5,
     "messageSize": 200,
     "publishesTo": ["temperature_r324"],
     "distribution": "exponential" },
…
```

```
 "applications": [
{    "applicationId": "app1",
     "applicationName": "dashboard",
     "applicationCategory" : "AN",
     "priority": 0,
     "subscribesTo": ["temperature_r324",
"smoke_r324"] },
…
```

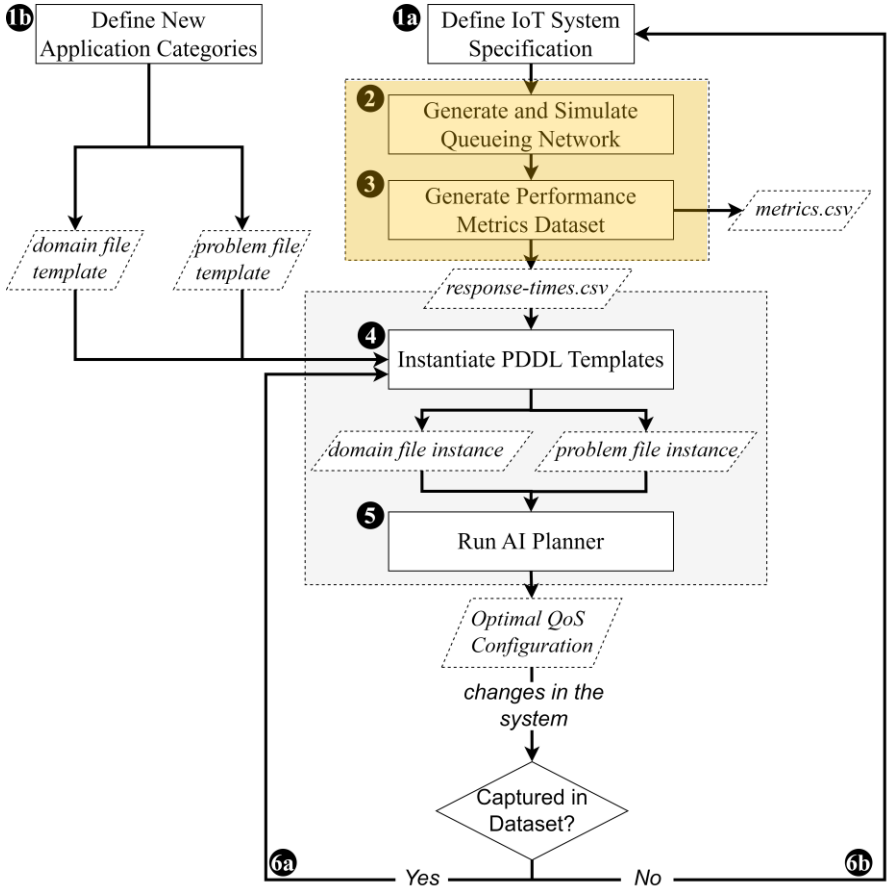
```
"systemBandwidth": 70,
"bandwidthPolicy": "default",
"priorityPolicy": "apps",
"dropRateAN": 0,
"dropRateRT": 0,
"dropRateTS": 0,
"dropRateVS": 0,
"brokerCapacity": 10000
```

```
"topics": [
{   "topicId": "topic_topic1",
    "topicName": "topic_topic1",
    "publishers": ["topic1_source"],
    "subscribers": ["app1", "app3",
                    "app5", "app_app7"]
},
…
```

```
"applicationCategories": [
  { "categoryId": "AN",
    "categoryName": "analytics"},
  { "categoryId": "RT",
    "categoryName": "realtime"},
  { "categoryId": "TS",
    "categoryName": "transactional"},
  { "categoryId": "VS",
    "categoryName": "videoStreaming"}
```

Using PlanIoT

Generating Performance Metrics Datasets



Queueing Network Composer + Dataset Generator

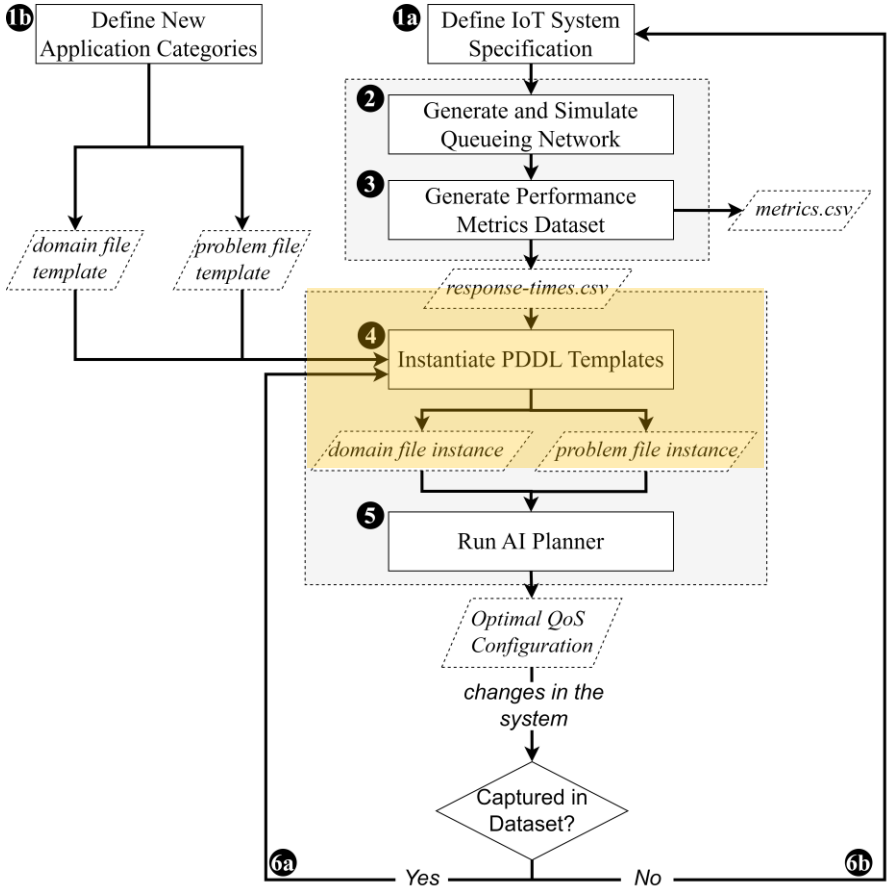
```
$ run_simulation.sh <system-specifications> <dataset> <simulation-duration> <alias>

$ run_simulation.sh …/default.json …/response-times.csv 300 default
$ run_simulation.sh …/prioritize_RT.json …/ response-times.csv 300
$ run_simulation.sh …/prioritize_VS.json …/ response-times.csv 300 prioVS

$ run_simulation.sh …/drop_VS_2.json …/ response-times.csv 300 dropVS2
```

CSV

response-times.csv

Using PlanIoT

Instantiating PDDL Files



PDDL Modeler

```
$ python InstantiatePddlTemplates.py <dataset> <domain-template> <problem-template>
```

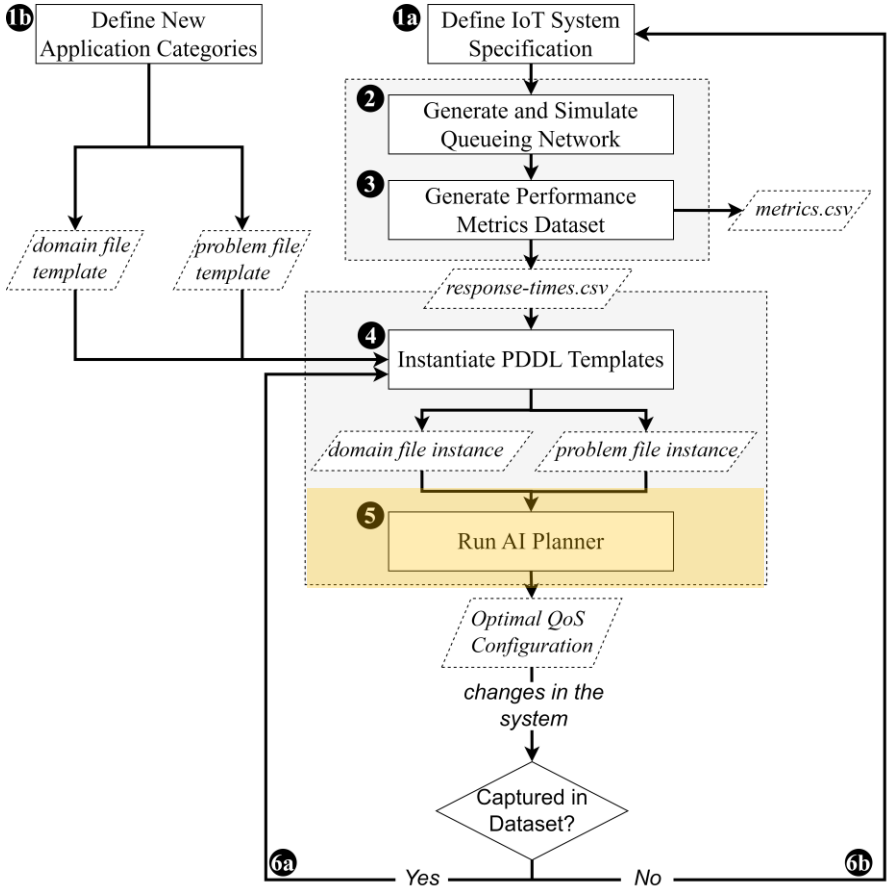
```
$ python InstantiatePddlTemplates.py …/response-times.csv \
 …/domain-template.pddl …/problem-template.pddl
```

`domain-generated.pddl`

`problem-generated.pddl`

Using PlanIoT

Running the AI Planner



Metric-FF

```
$ run_planner.sh <domain-file> <problem-file> <solution-file>
```

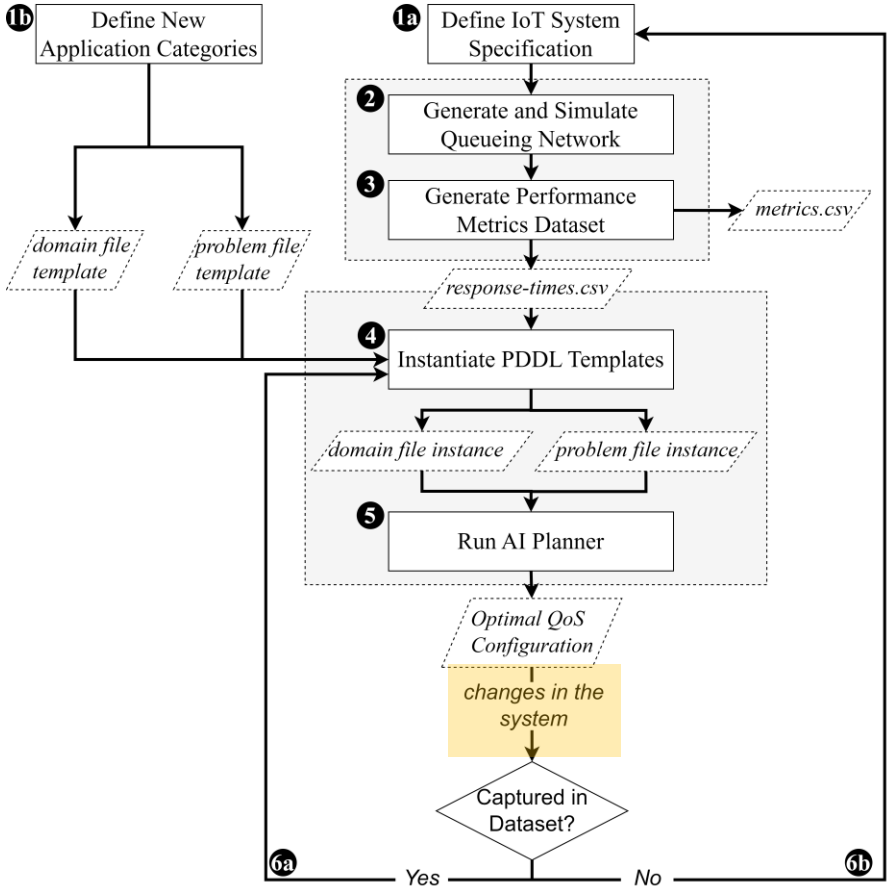
```
$ run_planner.sh …/domain-generated.pddl …/problem-generated.pddl …/solution.pddl
```

solution.pddl

```
ff: parsing domain file domain 'DOMAIN_NAME' defined
... done.
ff: parsing problem file problem 'PROBLEM_NAME'
defined ...
done. ...
ff: found legal plan as follows
step    0: DROPPINGVS2AN2 TOPIC_ALL APP_ALL
        1: PRIORITIZE_RT_VS_TS_AN TOPIC_ALL APP_ALL
plan cost: 59.439987
```

Using PlanIoT

Generating Adaptation Plans




```
$ python InstantiatePddlTemplates.py <dataset> <domain-template> <problem-template>
$ run_planner.sh <domain-file> <problem-file> <solution-file>

$ python InstantiatePddlTemplates.py …/response-times.csv \
 …/domain-template.pddl …/problem-template.pddl -e
$ run_planner.sh …/domain-emergency.pddl …/problem-emergency.pddl …/solution.pddl
```

```
ff: parsing domain file domain 'DOMAIN_NAME' defined
... done.
ff: parsing problem file problem 'PROBLEM_NAME' defined
...
done. ...
ff: found legal plan as follows
step    0: DROPPINGV102AN5 TOPIC_ALL APP_ALL
        1: PRIORITIZE_EM_RT TOPIC_ALL APP_ALL
plan cost: 67.142951
```

Takeaways and Conclusions

- PlanIoT can be used to (re)adapt data flows of Edge infrastructure in IoT-enhanced spaces by providing optimal plans for satisfying QoS requirements of deployed applications
- We implement a queueing network composer to generate a dataset that captures the performance of the IoT system.
- A set of automated planning components are used for choosing a configuration plan for the adaptive management of IoT flows at the Edge.

Thank you!

Artifact: Implementation of an Adaptive Flow Management Framework for IoT Spaces

Houssam Hajj Hassan, Georgios Bouloukakis, Ajay Kattepur, Denis Conan, Djamel Belaïd

Télécom SudParis, IP Paris, France
Ericsson AI Research, India

TRY IT OUT!

Contact: houssam.hajj_hassan@telecom-sudparis.eu