EDICT: Simulation of Edge Interactions across IoT-enhanced Environments

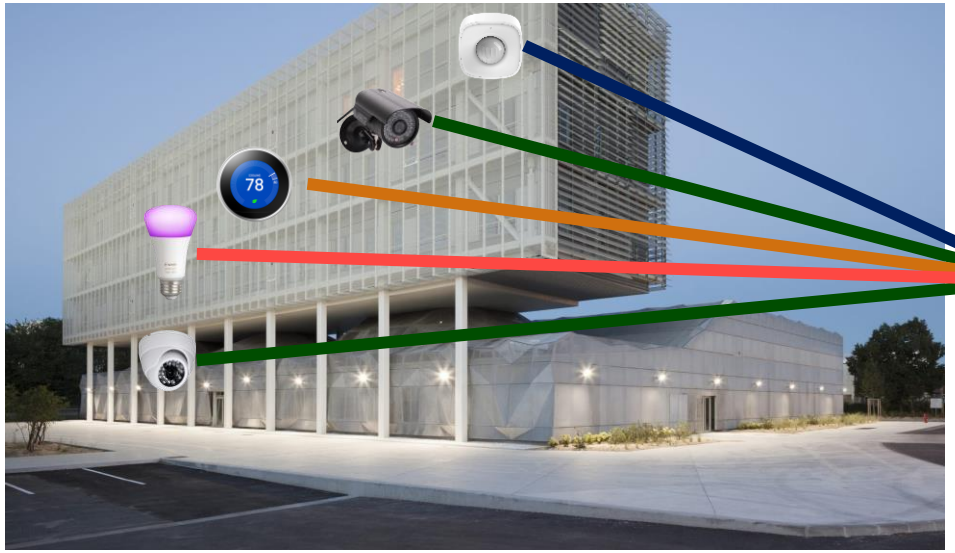
Houssam Hajj Hassan, Georgios Bouloukakis,
Ajay Kattepur, Denis Conan, Djamel Belaïd

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

Motivating Scenario

Smart Spaces' IoT Applications

Télécom SudParis - Évry campus

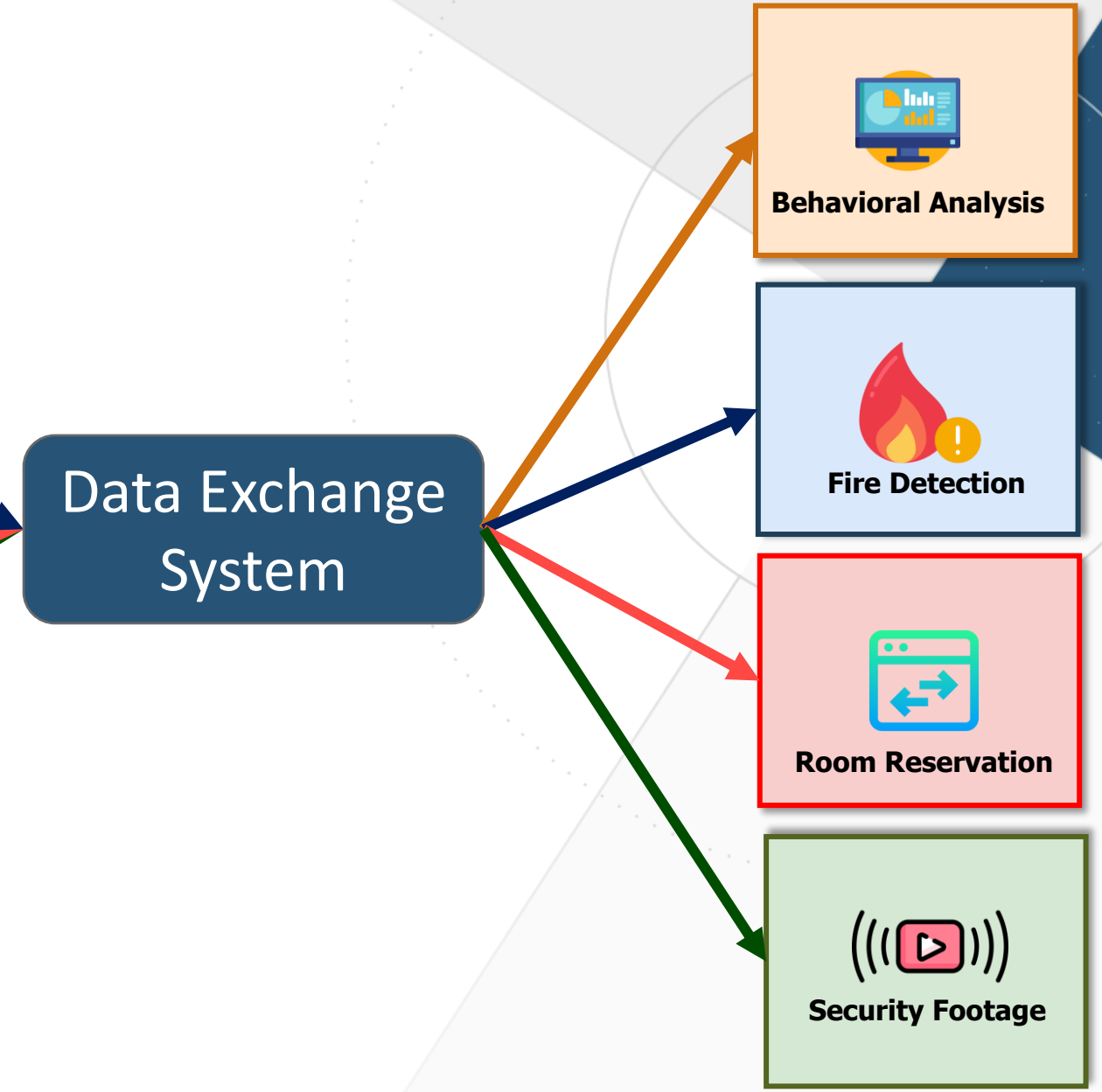


Building occupants:

- Approximately 6,000 occupants / day

Shared rooms:

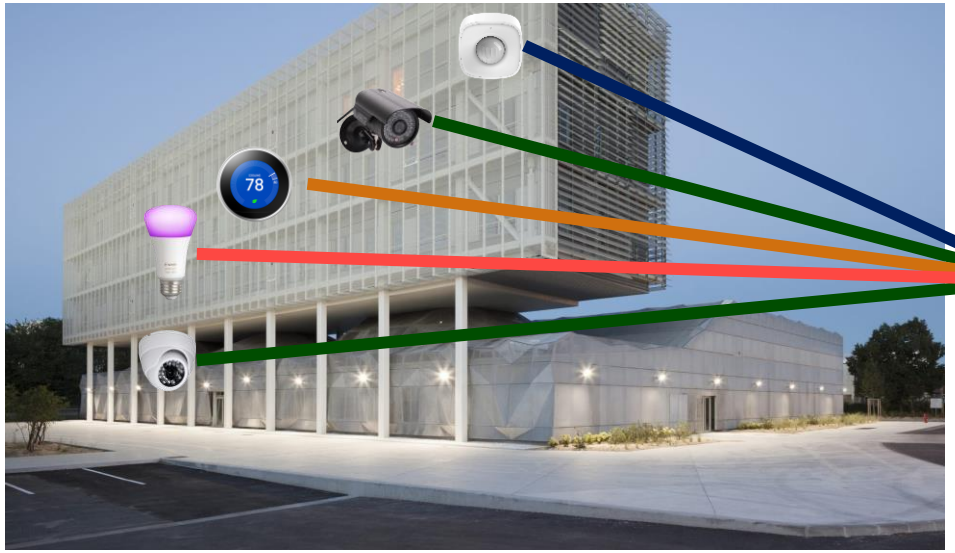
- 10 lecture halls, 220 conference & medium-size rooms
- 4 cafes and 3 restaurants



Motivating Scenario

Smart Spaces' IoT Applications

Télécom SudParis - Évry campus

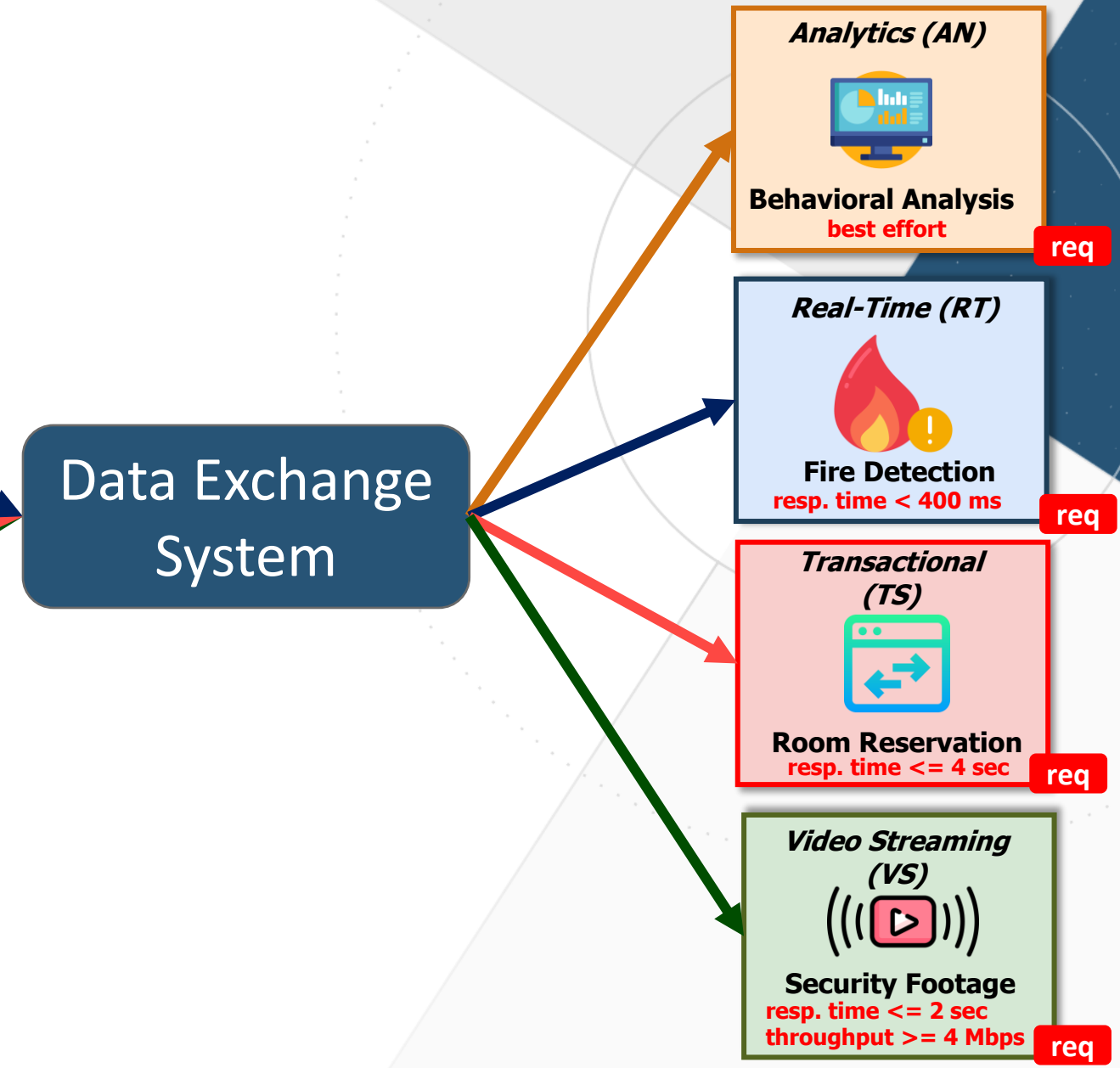


Building occupants:

- Approximately 6,000 occupants / day

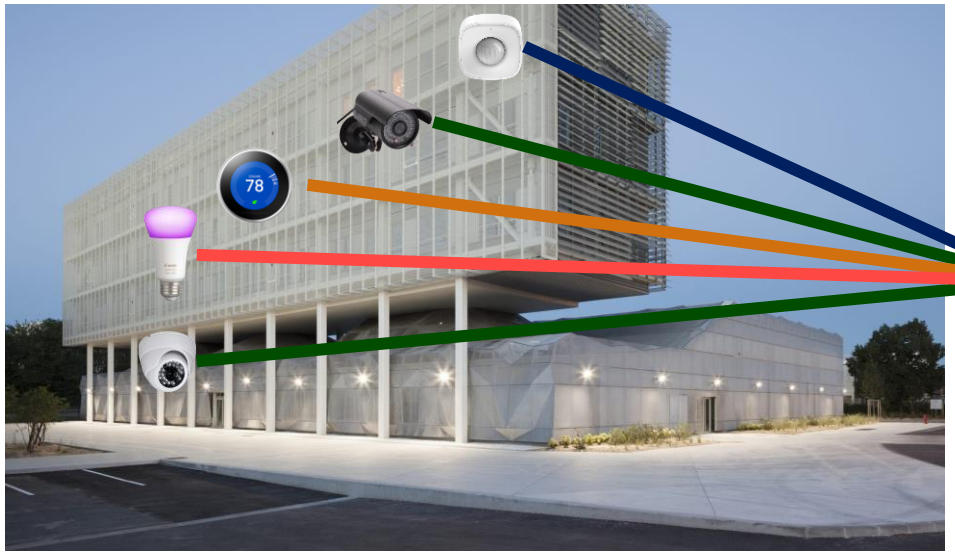
Shared rooms:

- 10 lecture halls, 220 conference & medium-size rooms
- 4 cafes and 3 restaurants



Motivating Scenario

Smart Spaces' IoT Applications



I need to tune the data exchange system to satisfy the QoS requirements of applications

Data Exchange System

Analytics (AN)

Behavioral Analysis

best effort

req

Real-Time (RT)

Fire Detection

resp. time < 400 ms

req

Transactional (TS)

Room Reservation

resp. time <= 4 sec

req

Video Streaming (VS)

Security Footage

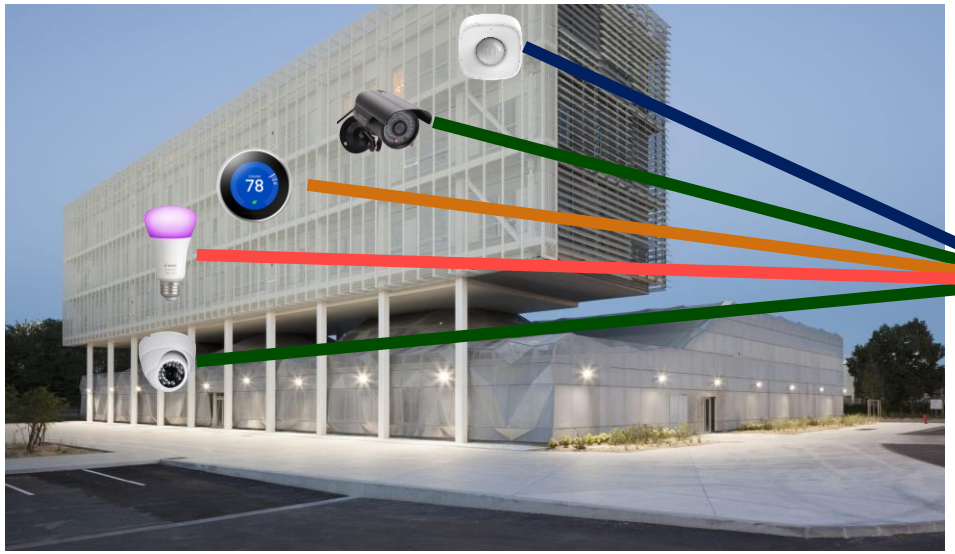
resp. time <= 2 sec

throughput >= 4 Mbps

req

Motivating Scenario

Smart Spaces' IoT Applications



- **ML/AI techniques**
- **Optimization techniques**
- **...**

Data Exchange System

Analytics (AN)

Behavioral Analysis

best effort

req

Real-Time (RT)

Fire Detection

resp. time < 400 ms

req

Transactional (TS)

Room Reservation

resp. time <= 4 sec

req

Video Streaming (VS)

Security Footage

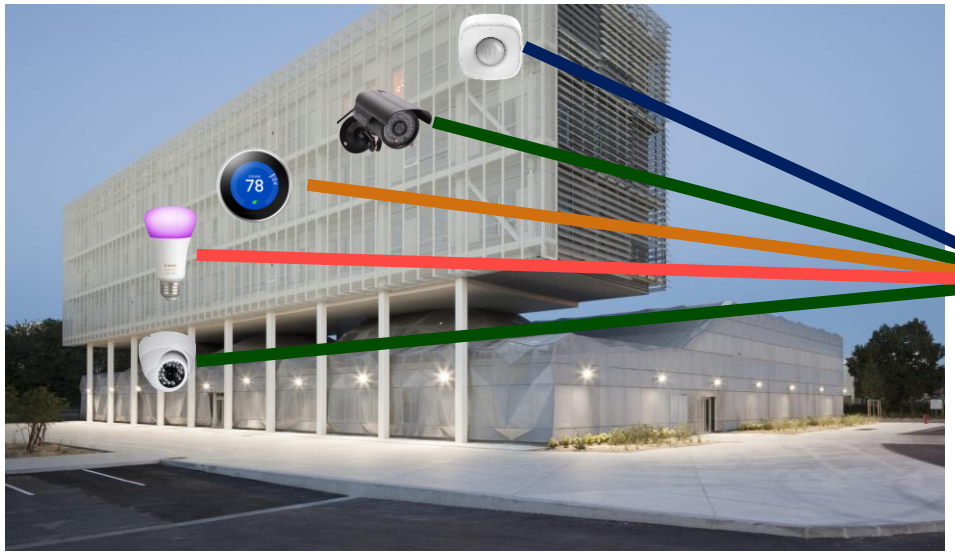
resp. time <= 2 sec

throughput >= 4 Mbps

req

Motivating Scenario

Smart Spaces' IoT Applications



How to get data that captures the performance of the data exchange system?

Data Exchange System

Analytics (AN)

Behavioral Analysis
best effort
req

Real-Time (RT)

Fire Detection
resp. time < 400 ms
req

Transactional (TS)

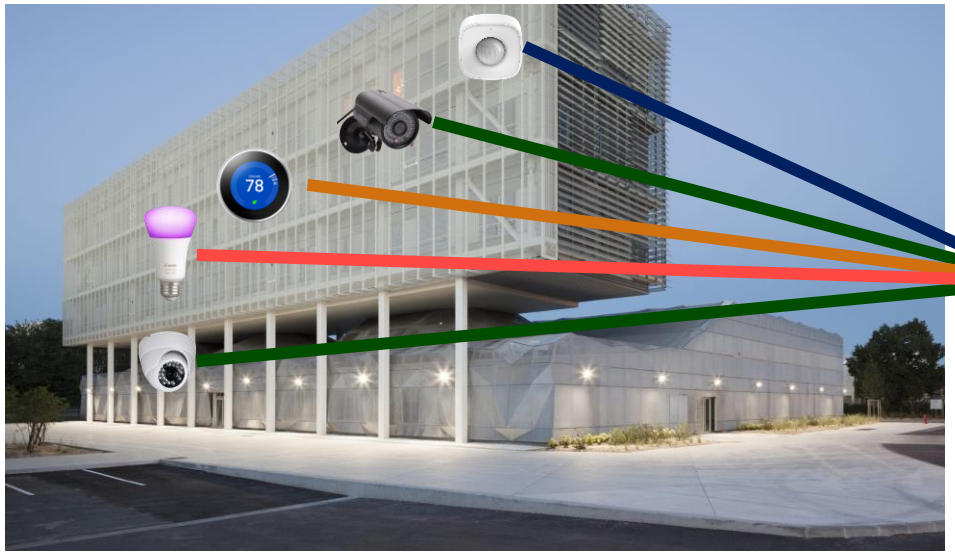
Room Reservation
resp. time <= 4 sec
req

Video Streaming (VS)

Security Footage
resp. time <= 2 sec
throughput >= 4 Mbps
req

Motivating Scenario

Smart Spaces' IoT Applications



How to get data that captures the performance of the data exchange system?

Data Exchange System

Analytics (AN)

Behavioral Analysis

best effort

req

Real-Time (RT)

Fire Detection

resp. time < 400 ms

req

Transactional (TS)

Room Reservation

resp. time <= 4 sec

req

Video Streaming (VS)

Security Footage

resp. time <= 2 sec

throughput >= 4 Mbps

req

Publicly available datasets

- Privacy/security concerns
- Finding datasets that accurately represent the IoT environment to be tuned

Implement the IoT system

- Expensive to deploy and test multiple scenarios
- Lengthy process

Existing Simulators

IoT Simulators:

- Resource allocation and provisioning: iFogSim[1], IoTSim-Edge [2]
- Cloud computing: IoTSim [3], CloudSim [4], Kaala [5]
- IoT Microservices: DPWSim [6]
- Smart city: IoTIFY [7], CupCarbon [8]

Network Emulators

- Evaluation of network protocols: OMNeT++ [9], QualNet [10]
- Discrete-event Simulation: NS-3 [11]

[1] H. Gupta, A. Dastjerdi, S.K. Ghosh et al. Software: Practice and Experience. 2017.
[2] D.N. Jha, K. Alwasel, A. Alshoshan. Software: Practice and Experience. 2020.
[3] X. Zeng, S.K. Garg, P. Strazdins et al. Journal of Systems Architecture. 2017.
[4] R.N. Calheiros, R. Ranjan, A. Beloglazov et al. Software: Practice and Experience. 2011.
[5] U.K. Kumar, R.A. Fezeu, T.J. Salo et al. NET4us. 2022.
[6] S.N. Han, G.M. Lee, N. Crespi et al. IEEE WFIoT. 2014.
[7] https://iotify.io
[8] K. Mehdi, M. Lounis, A. Bounceur et al. SIMUTools. 2014.
[9] A. Varga and R. Hornig. SIMUTools. 2010.
[10] S. Dinesh and G. Sonal. IJICT. 2014.
[11] G.F. Riley and T. R. Henderson. Modeling and Tools for Network Simulation. 2010.

Existing Simulators

IoT Simulators:

- Resource allocation and provisioning: iFogSim[1], IoTSim-Edge [2]
- Cloud computing: IoTSim [3], CloudSim [4], Kaala [5]
- IoT Microservices: DPWSim [6]
- Smart city: IoTIFY [7], CupCarbon [8]

Usability: Developers must learn and get used to the specifics of the simulator they choose to use
System Reconfiguration: It's complicated to simulate the same environment with multiple configuration parameters.

Network Emulators

- Evaluation of network protocols: OMNeT++ [9], QualNet [10]
- Discrete-event Simulation: NS-3 [11]

Automated System Tuning: Simulation results are not always provided in suitable formats.
Runtime Adaptation: At runtime, if changes occur in the Edge infrastructure, IoT designers must re-run the simulations.

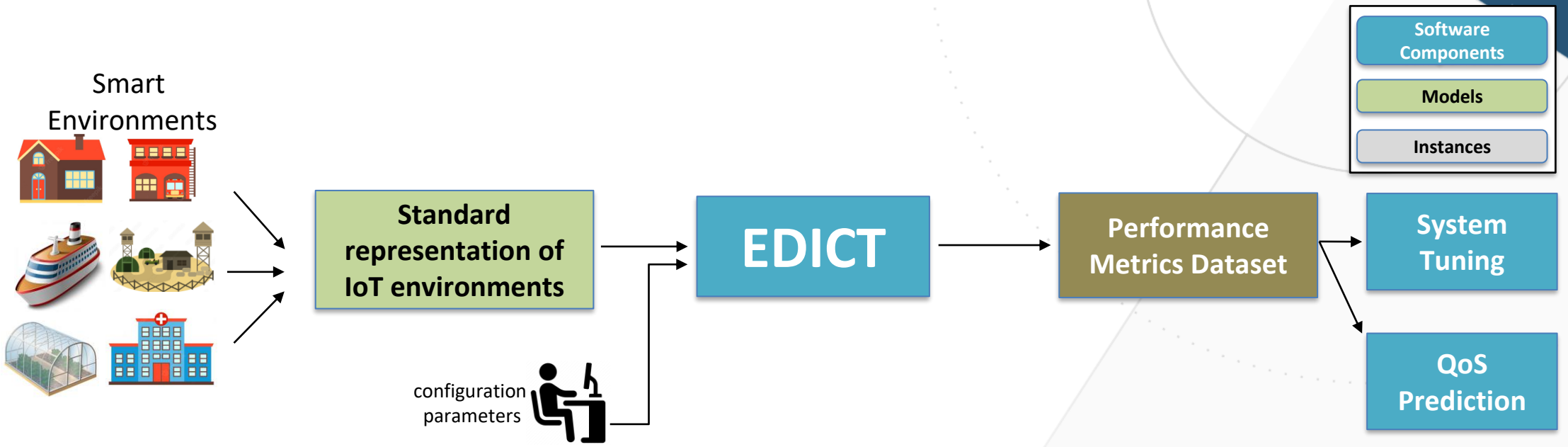
[1] H. Gupta, A. Dastjerdi, S.K. Ghosh et al. Software: Practice and Experience. 2017.
[2] D.N. Jha, K. Alwasel, A. Alshoshan. Software: Practice and Experience. 2020.
[3] X. Zeng, S.K. Garg, P. Strazdins et al. Journal of Systems Architecture. 2017.
[4] R.N. Calheiros, R. Ranjan, A. Beloglazov et al. Software: Practice and Experience. 2011.
[5] U.K. Kumar, R.A. Fezeu, T.J. Salo et al. NET4us. 2022.
[6] S.N. Han, G.M. Lee, N. Crespi et al. IEEE WFIoT. 2014.
[7] https://iotify.io
[8] K. Mehdi, M. Lounis, A. Bounceur et al. SIMUTools. 2014.
[9] A. Varga and R. Hornig. SIMUTools. 2010.
[10] S. Dinesh and G. Sonal. IJICT. 2014.
[11] G.F. Riley and T. R. Henderson. Modeling and Tools for Network Simulation. 2010.

The EDICT Simulator

High-Level Overview

EDICT is a tool for simulating **message-layer interactions at the Edge**. EDICT differs from existing tools by:

- Using **standard data models** for representing IoT-enhanced environments
- Abstracting hardware and network-layer implementation details to focus on **application-layer interactions**.
- Generating performance metrics datasets in user-friendly formats for **automated system tuning and performance prediction.**



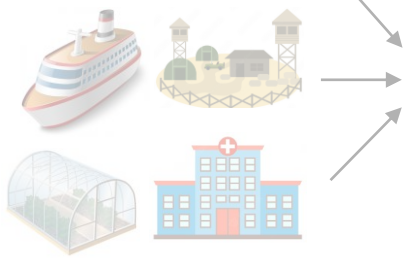
The EDICT Simulator

High-Level Overview

EDICT is a tool for simulating **message-layer interactions at the Edge**. EDICT differs from existing tools by:

- Using **standard data models** for representing IoT-enhanced environments
- Abstracting hardware and network-layer implementation details to focus on **application-layer interactions**.
- Generating performance metrics datasets in user-friendly formats for **automated system tuning and performance prediction.**

How can we represent generic IoT environments in a structured manner?



Standard representation of IoT environments

configuration parameters

EDICT

Performance Metrics Dataset

Software Components

Models

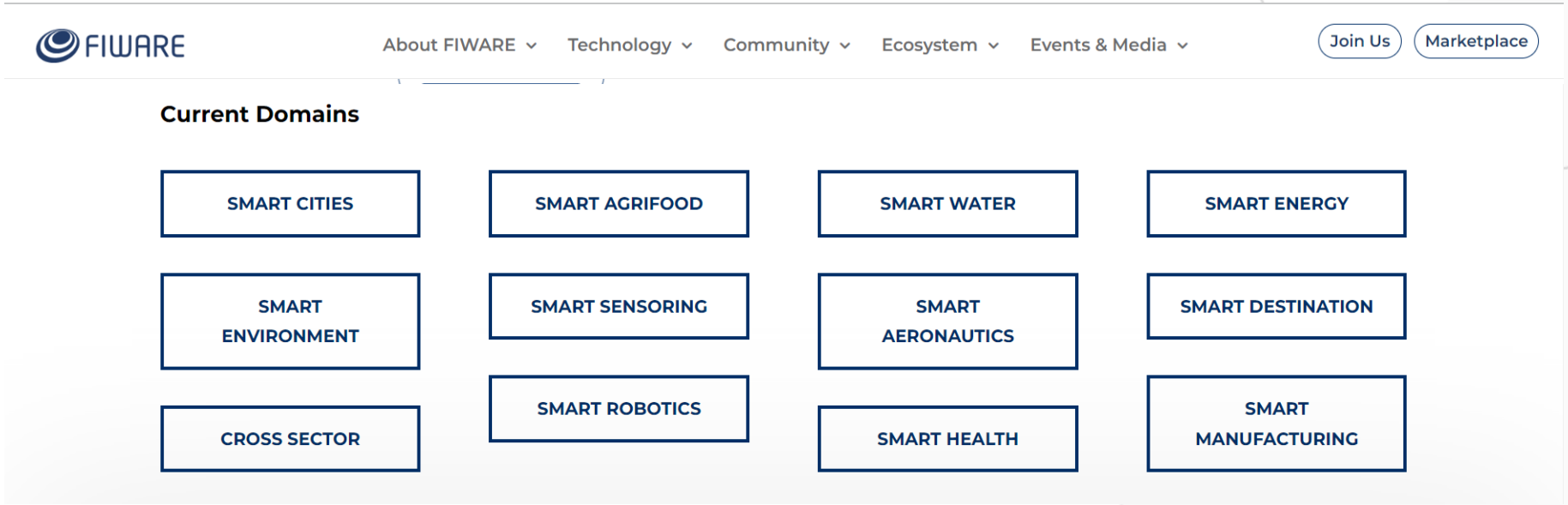
Instances

System Tuning

QoS Prediction

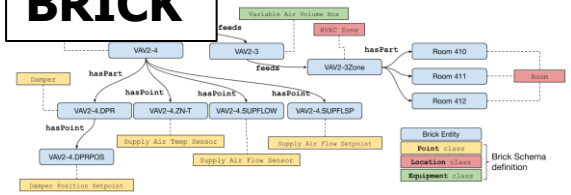
NGSI-LD Representation of IoT-enhanced Environments

- NGSI-LD (*Next Generation Service Interfaces – Linked Data*) is a specification for data representation based on **property graphs**.
- The FIWARE foundation defines smart data models for IoT-related domains: smart cities, smart agriculture, smart manufacturing, etc.
- However, there aren't data models that focus on **data exchange in IoT environments.**

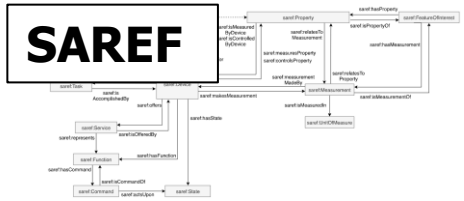


NGSI-LD Representation of IoT-enhanced Environments

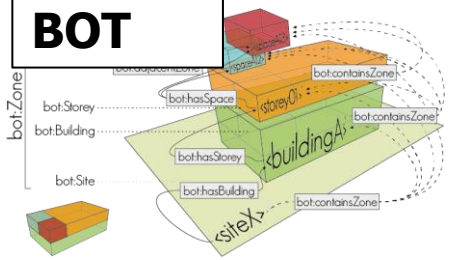
BRICK



SAREF

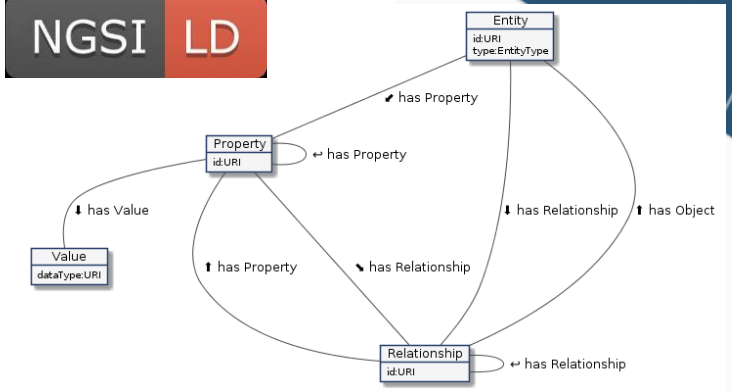


BOT



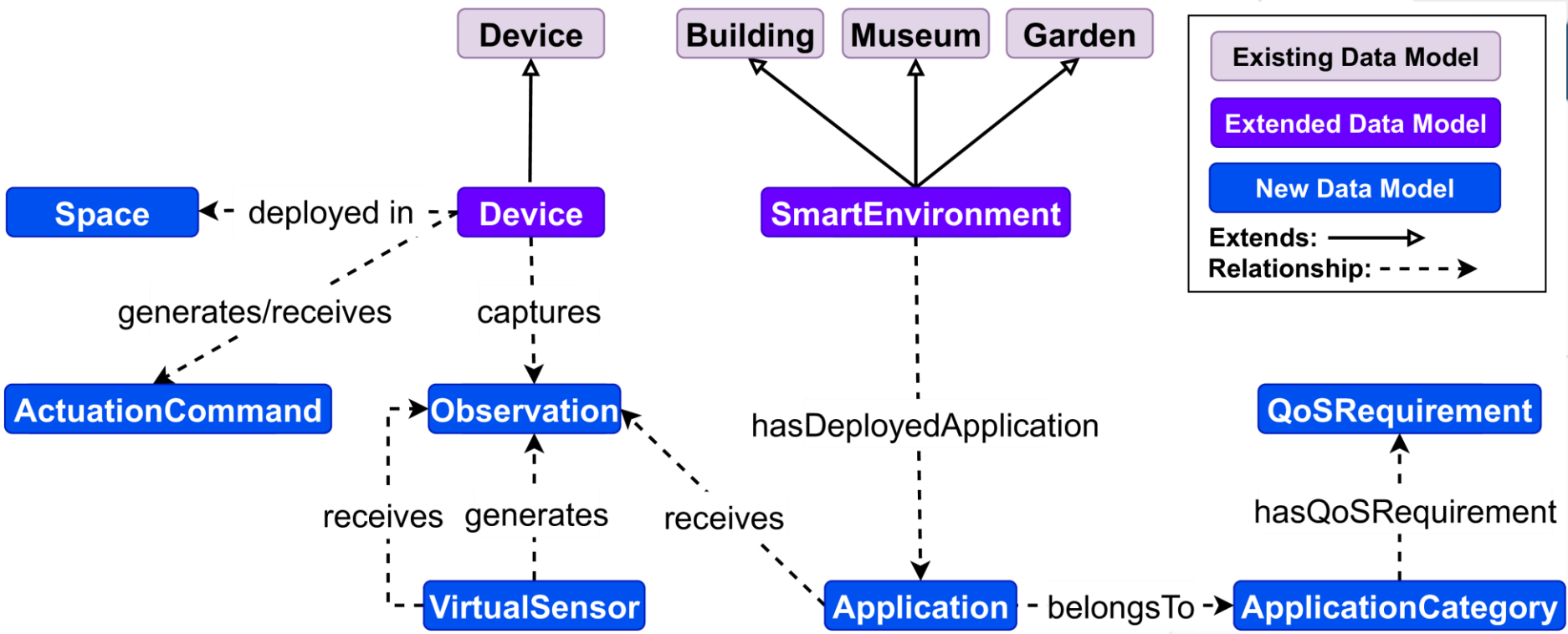
Convert[1]

NGSI LD



[1] A. Abid, J. Lee, F. Le Gall, and J. Song. 2022. Toward Mapping an NGSI-LD Context Model on RDF Graph Approaches: A Comparison Study. Sensors 22, 13 (2022). https://doi.org/10.3390/s22134798

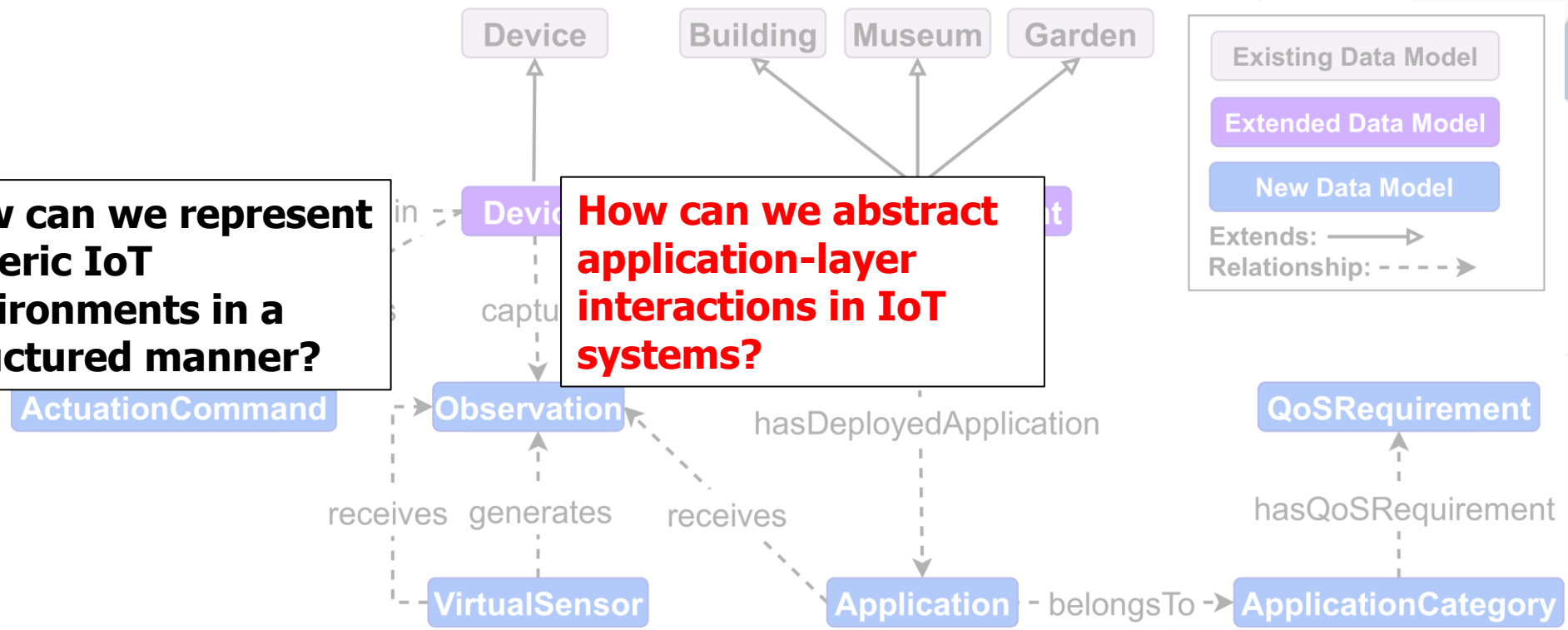
NGSI-LD Representation of IoT-enhanced Environments



NGSI-LD Representation of IoT-enhanced Environments

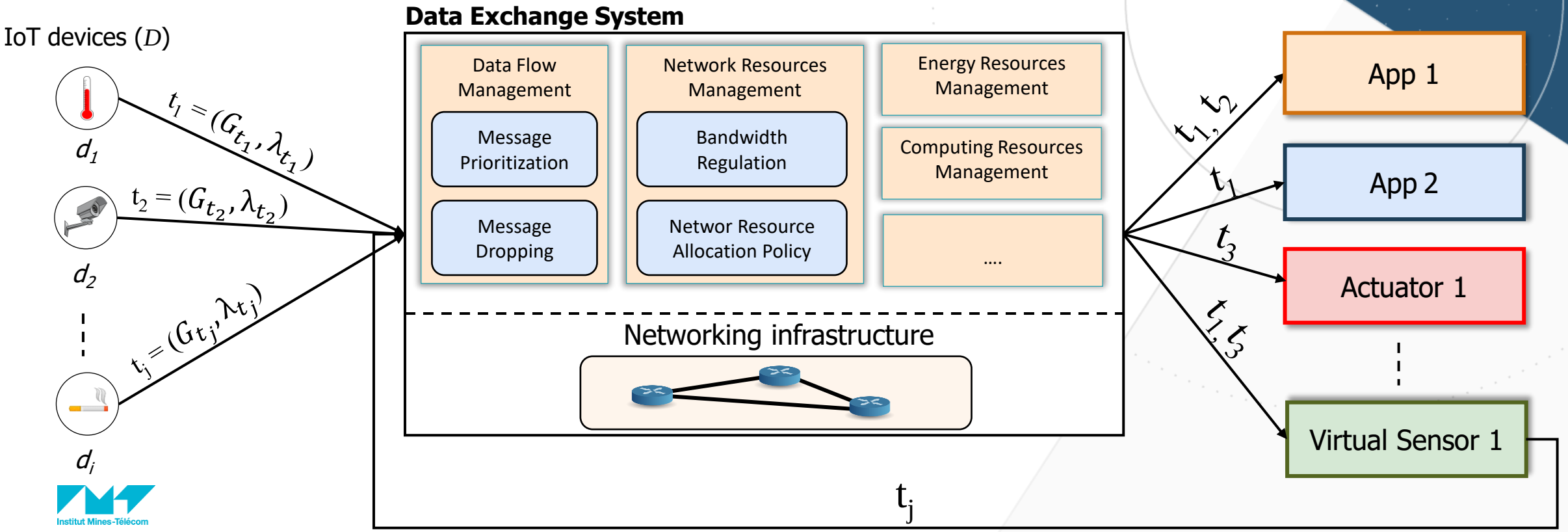
How can we represent generic IoT environments in a structured manner?

How can we abstract application-layer interactions in IoT systems?



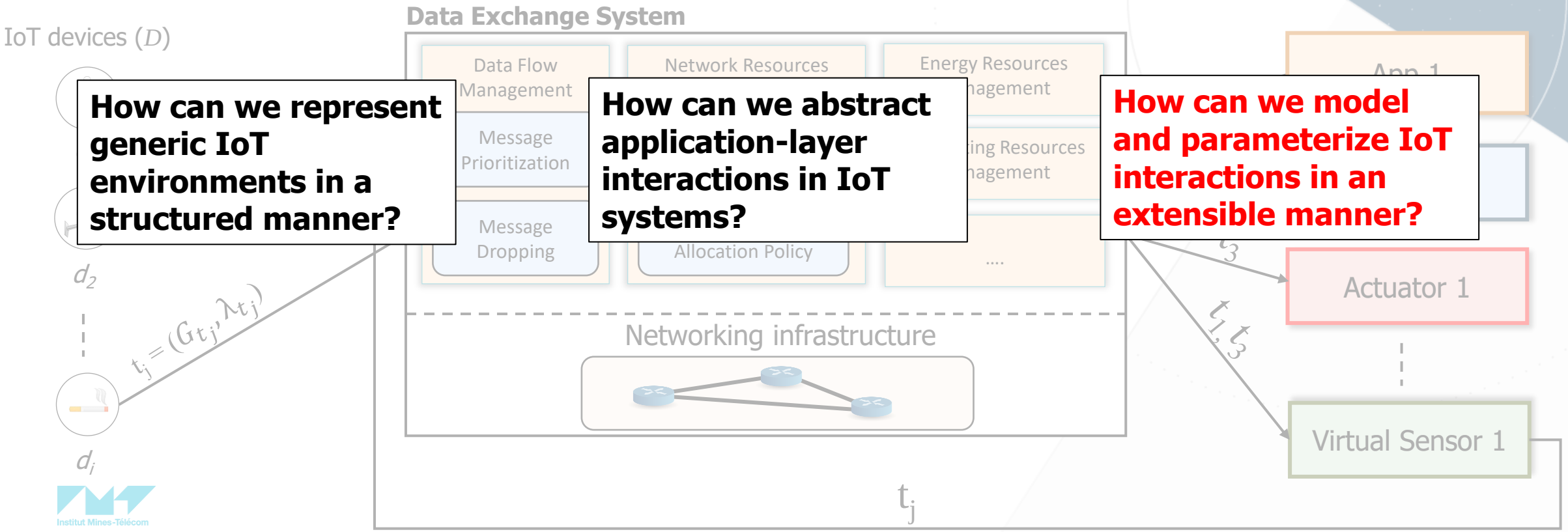
IoT Data Exchange Representation

Notation	Description
$d_i \in D$; $a_i \in A$	IoT devices; IoT applications
$t_j \in T$	topics
G_{t_j}	message size
λ_{t_j}	publication rate

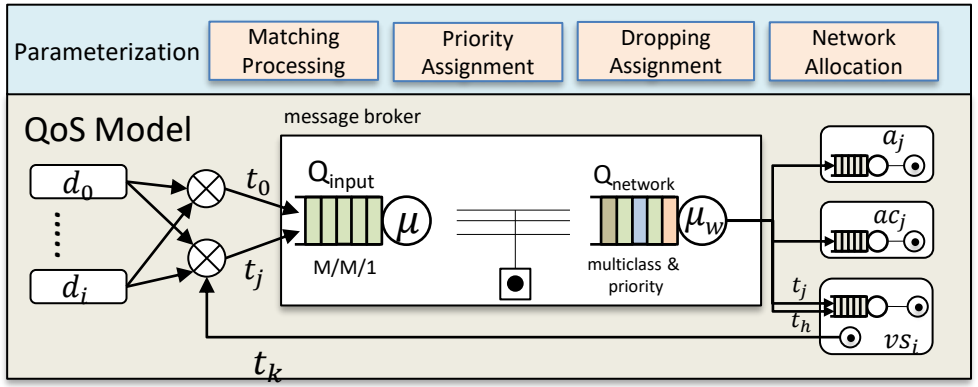


IoT Data Exchange Representation

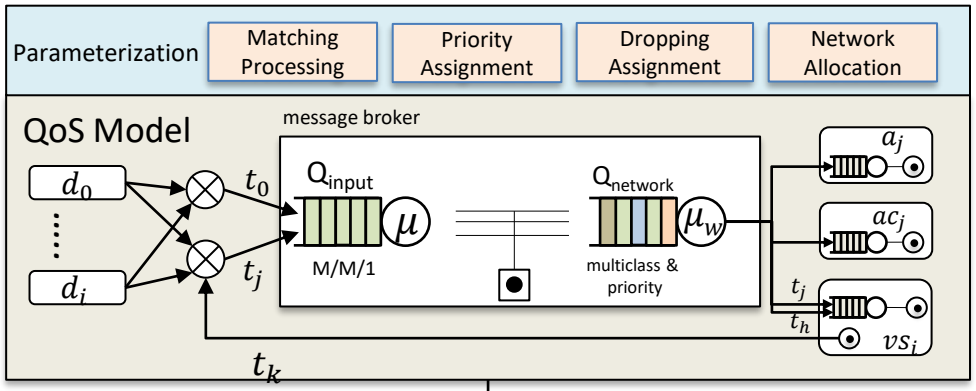
Notation	Description
$d_i \in D$; $a_i \in A$	IoT devices; IoT applications
$t_j \in T$	topics
G_{t_j}	message size
λ_{t_j}	publication rate



Queueing Network Composition



Queueing Network Composition

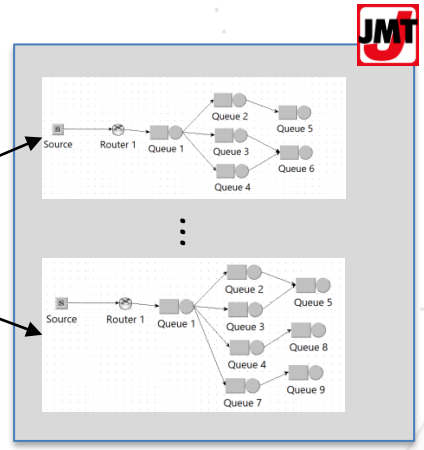


The Java Modelling Tools[1] (JMT) simulator is used to compose and simulate queueing models.

JMT Library

addStation()
setConnection()

Queuing Composer



NGSI-LD Instance

Dataset

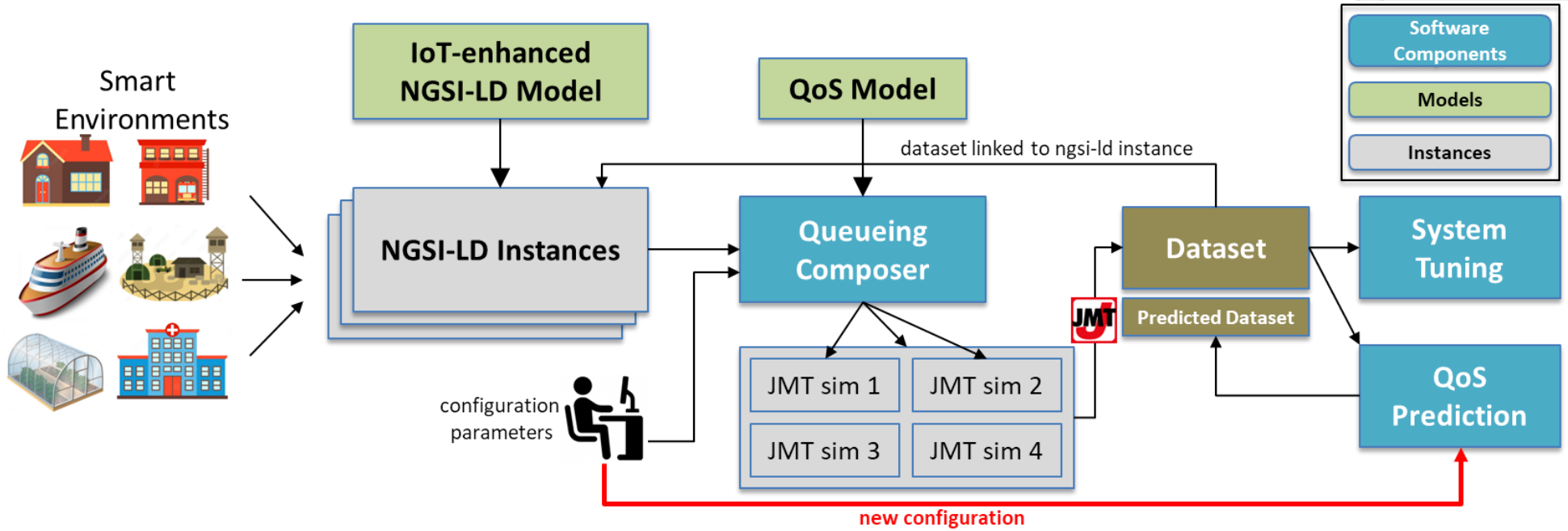
Response times
Throughput
Dropping
….

System Tuning

QoS Prediction

[1] https://jmt.sourceforge.net/

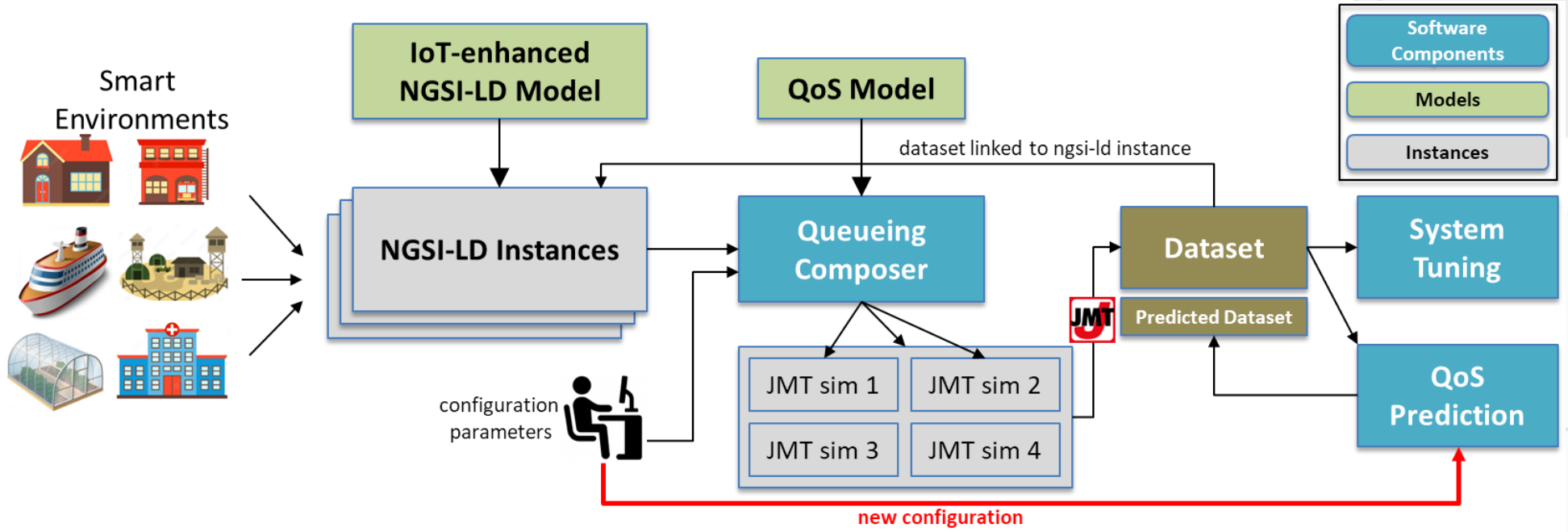
The EDICT Architecture



Because of the extensibility of queueing models, EDICT can be extended to support simulating additional aspects of IoT environments:

- disconnections of IoT nodes (through ON/OFF queues)
- energy consumption (through energy packet networks)
- …

The EDICT Architecture



Datasets generated by EDICT have been used in PlanIoT[1,2] to support autonomous IoT systems in dynamic environments through AI planning techniques.

[1] H. Hajj Hassan, G. Bouloukakis, A. Kattepur, D. Conan, D. Belaïd. PlanIoT: A Framework for Adaptive Data Flow Management in IoT-enhanced Spaces. SEAMS'23.
[2] H. Hajj Hassan, G. Bouloukakis, A. Kattepur, D. Conan, D. Belaïd. Artifact: Implementation of an Adaptive Flow Management Framework for IoT Spaces. SEAMS'23.

EDICT Evaluation

Experimental Setup

IoT-enhanced Environment Properties				*QoS Requirements*		
App. categories	**|A|**	**|R|**	W_{DX}	δ_{max}	θ_{min}	ω_{max}
AN	6	21	650 MB /s	best effort	best effort	best effort
RT	9	17		<400 ms	384 KB / s	0%
TS	6	12		<4 s	-	0%
ST	9	10		<2 s	384 KB / s	<2%
Total	30	60	650 MB/s			

30 apps / 60 subscriptions

based on ETSI TS 1212 105

EDICT Evaluation

Experimental Setup

IoT-enhanced Environment Properties				*QoS Requirements*		
App. categories	**|A|**	**|R|**	W_{DX}	δ_{max}	θ_{min}	ω_{max}
AN	6	21	650 MB /s	best effort	best effort	best effort
RT	9	17		<400 ms	384 KB / s	0%
TS	6	12		<4 s	-	0%
ST	9	10		<2 s	384 KB / s	<2%
Total	30	60	650 MB/s			

30 apps / 60 subscriptions

based on ETSI TS 1212 105

Use case 1: System Tuning

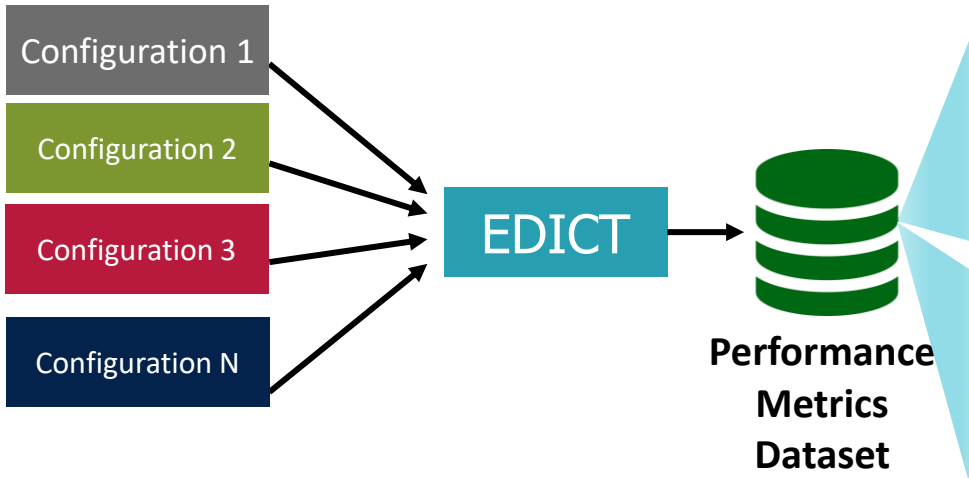
- Evaluation of different tuning policies to find the one that best satisfies the QoS of applications

Use case 2: QoS Prediction

- Demonstrate how EDICT can be used for runtime QoS prediction.
- Evaluation of different ML prediction models.

EDICT Evaluation

Performance Metrics Dataset Generation



App	Topic	Prioritize RT	Prioritize TS	...	Drop 2% VS	Drop 5% AN	...	Shared	Max-Min	...
app 1	smoke	0.2015	0.560679		0.4761	0.4651		0.81563	0.498188	
app 1	temp	0.515479	0.13125		0.5193	0.4950		0.4980	0.131307	
app 3	temp	0.633439	0.5532327		0.4624	0.3168		0.51384	0.23485	
app j	occupancy	0.134651	0.345628		0.2156	0.5138		0.31564	0.154152	

Response times per subscription

App	Topic	Prioritize RT	Prioritize TS	...	Drop 2% VS	Drop 5% AN	...	Shared	Max-Min	...
app 1	smoke	452.141	262.432		450.14	453.51		363.43	479.532	
..	...	...	...		...	...		...	...	

Throughput per subscription

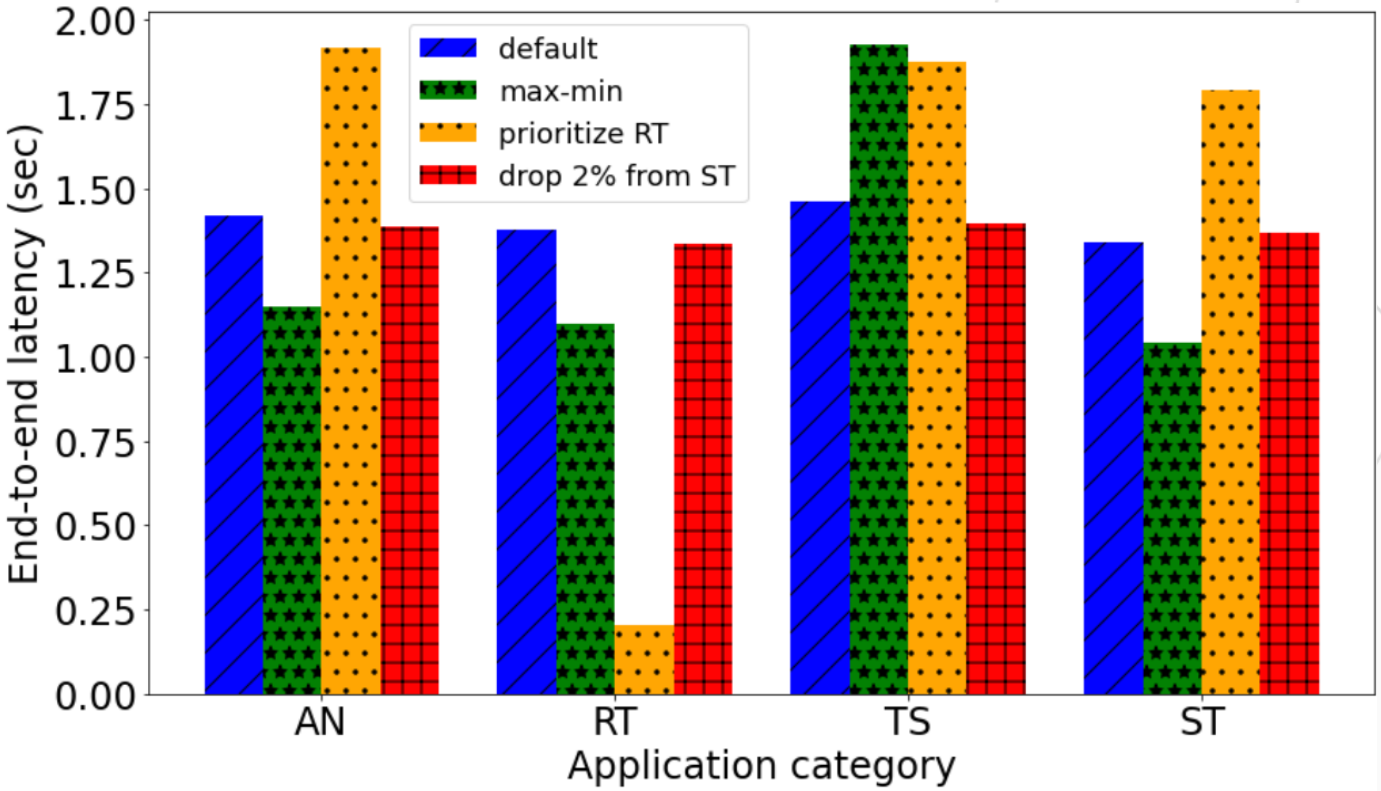
App	Topic	Prioritize RT	Prioritize TS	...	Drop 2% VS	Drop 5% AN	...	Shared	Max-Min	...
app 1	smoke	0.00	0.008243		0.0014	0.0005		0.000	0.000	
..	...	...	...		...	...		...	...	

Drop rate per subscription

EDICT Evaluation

Use Case 1: System Tuning

Response time per application category



EDICT Evaluation

Use Case 2: QoS Prediction

- Evaluated 4 ML models for predicting QoS metrics: KNN, Linear Regression, Decision Trees, and AWS's DataWig[1] Library.
- Scenario 1: prediction of QoS metrics when the number of subscriptions increases.
- Scenario 2: prediction of QoS metrics when we change the configuration parameters of the data exchange system.

	Prediction when adding subscriptions				*Prediction when reconfiguring*			
Dataset size	**KNN**	**LR**	**DT**	**DW**	**KNN**	**LR**	**DT**	**DW**
220	0.035	0.036	0.031	0.035	0.0203	0.021	0.022	0.024
440	0.069	0.191	0.072	0.205	0.235	0.195	0.233	0.162
660	0.101	0.648	0.110	0.706	0.090	0.27	0.066	0.118
880	0.199	2.882	0.193	3.165	0.247	3.973	0.062	1.424
1100	0.195	3.487	0.204	3.875	0.346	7.975	0.202	3.027

Comparison of RMSE (sec) of QoS predictions

[1] https://github.com/awslabs/datawig

Conclusion and Future Extensions

- We propose EDICT, a simulation tool for generating performance metrics datasets in IoT-enhanced environments.
- EDICT leverages the **NGSI-LD specification** as a structured representation of IoT environments.
- EDICT **abstracts application-layer interactions** through a publish/subscribe-based representation of IoT data flow interactions.
- EDICT relies on **queueing network modelling** as a flexible and extensible framework for modelling and parameterizing IoT interactions.
- The EDICT code is publicly available on: https://github.com/SAMSGBLab/edict.
- Future extensions of EDICT include modelling energy consumption and machine learning IoT applications.

Thank you!

EDICT: Simulation of Edge Interactions across IoT-enhanced Environments

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