Component-based Approach Combining UML and BIP for Rigorous System Design

Presented by **Salim Chehida**
-VERIMAG Lab, University of Grenoble Alpes

GT IDM & IE
TOULOUSE, 09/12/2021

OUTLINE

- BRAIN-IoT PROJECT
- APPROACH
- UML MODEL
- TRANSLATION TO BIP
- SMC ANALYSIS
- CONCLUSION

BRAIN-IoT PROJECT

(2018-2021)

https://www.brain-iot.eu/

Main Objectives

- **Model-based** development and validation of IoT applications
- **Distributed deployment on the Edge** of IoT Services & Applications
- Raw **data analysis** enabling system **resiliency**
- **Ensure secure** data communication and **Privacy Control** in **resource-constraint** & **distributed** environments

BRAIN-IoT AT GLANCE

BRAIN-IoT
Next-Gen IoT Platform

Main Outcome

Meta operating system for the implementation and execution of **decentralized IoT applications** with computing capacity **at the edge**

14 **Integrated Assets** — Exploitable as BRAIN-IoT Platform or stand-alone components

9 **Open Source SW Components** — Released under Eclipse Research Lab

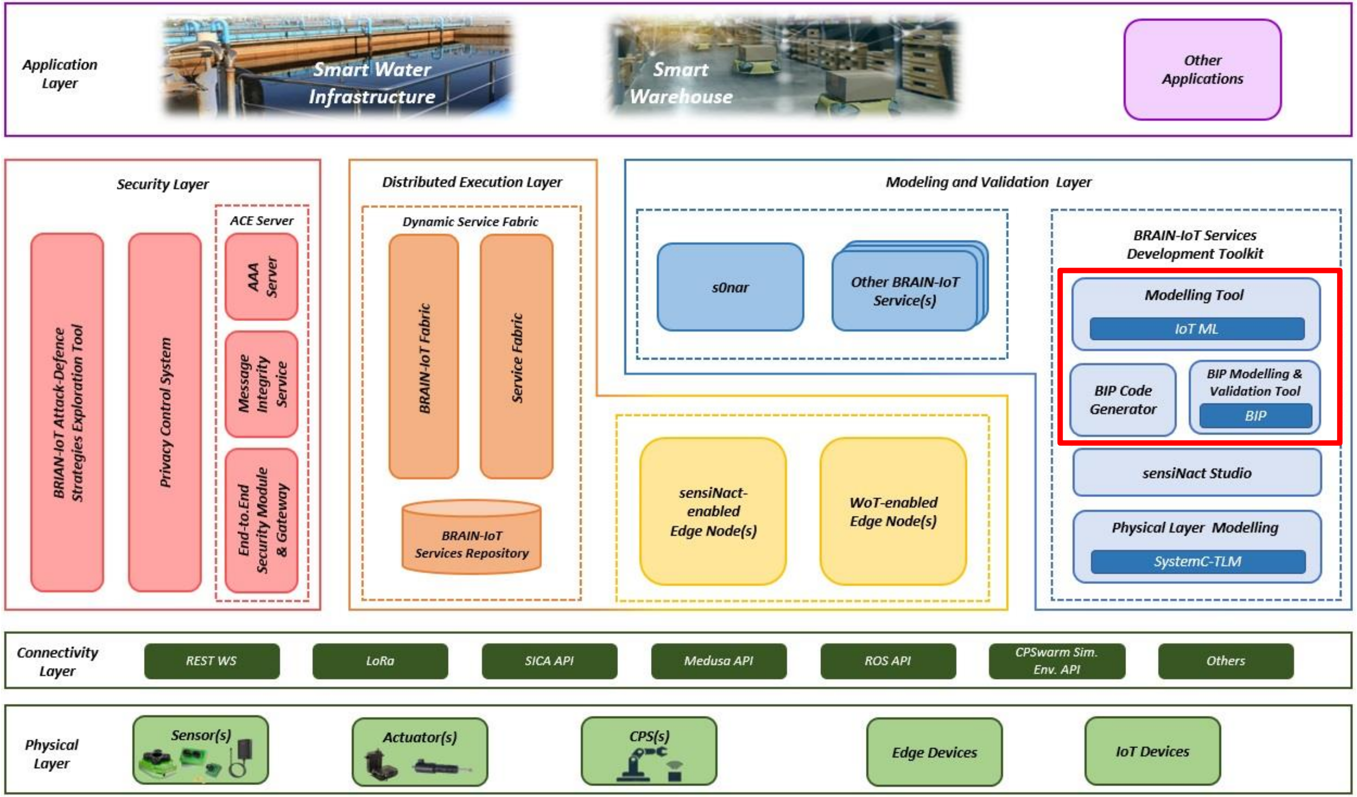
1 **Startup Creation** — Kentyou markets its solutions by leveraging BRAIN-IoT's business-friendly open-source ecosystem

3 **Contributions to Standards** — Contributions to OSGi, OMG MARTE2.0 and W3C WoT specifications and reference use-cases

21 **Scientific Publications** — 14 Conference Papers, 7 Journals and Book Chapters

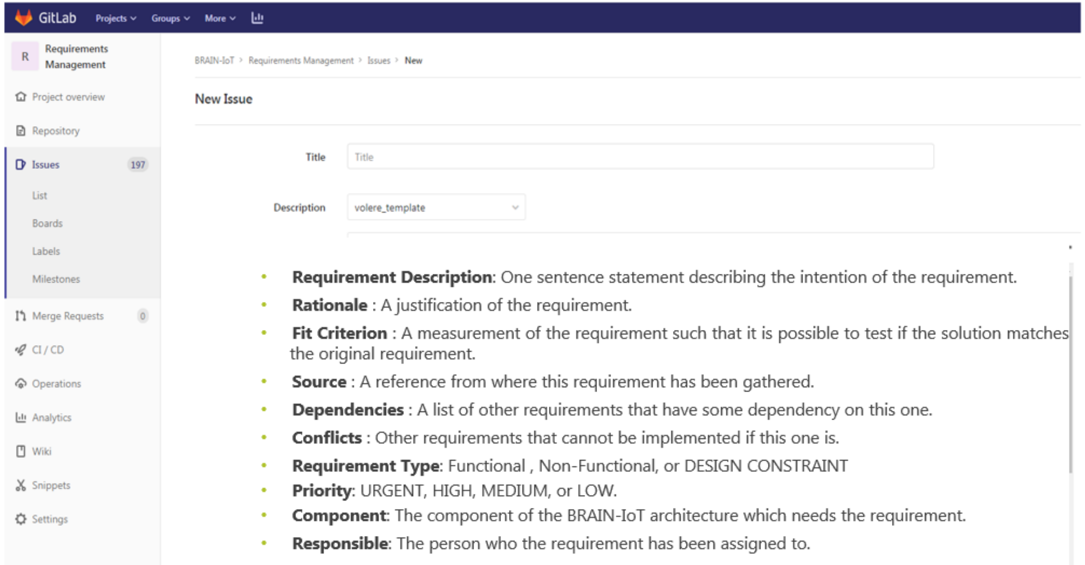
BRAIN-IoT REFERENCE ARCHITECTURE

https://github.com/orgs/eclipse-researchlabs/teams/brain-iot-team/repositories

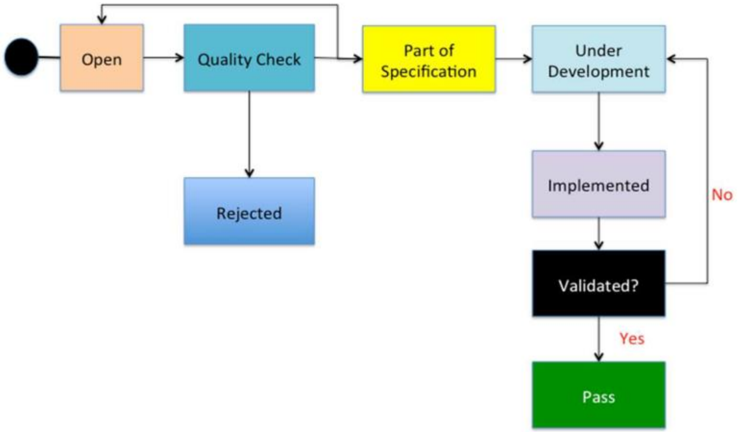


BRAIN-IoT REQUIREMENTS MANAGEMENT

GITLAB TOOL : https://git.repository-pert.ismb.it/BRAIN-IoT/requirements-management

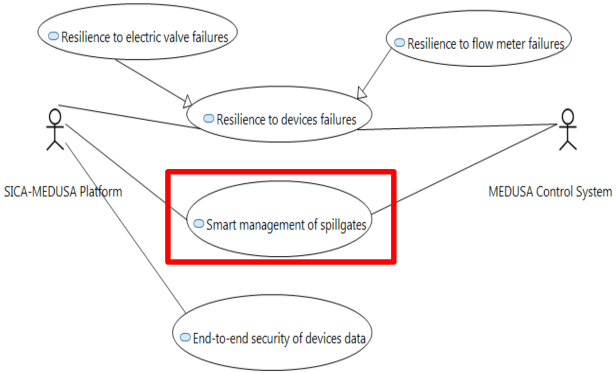


- **Requirement Description**: One sentence statement describing the intention of the requirement.
- **Rationale** : A justification of the requirement.
- **Fit Criterion** : A measurement of the requirement such that it is possible to test if the solution matches the original requirement.
- **Source** : A reference from where this requirement has been gathered.
- **Dependencies** : A list of other requirements that have some dependency on this one.
- **Conflicts** : Other requirements that cannot be implemented if this one is.
- **Requirement Type**: Functional , Non-Functional, or DESIGN CONSTRAINT
- **Priority**: URGENT, HIGH, MEDIUM, or LOW.
- **Component**: The component of the BRAIN-IoT architecture which needs the requirement.
- **Responsible**: The person who the requirement has been assigned to.

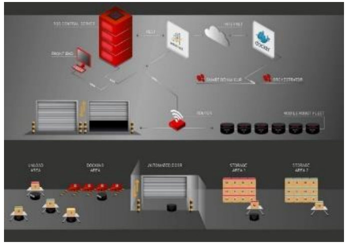


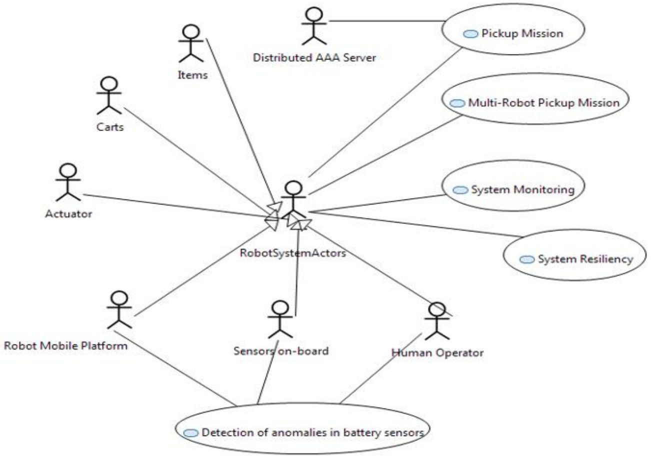
BRAIN-IOT USE CASES

- A scenario related to the management of water distribution network from **EMALCSA**.
- It focuses on **monitoring and control the management of the water urban** cycle in metropolitan environment of the city of *A Coruña* in Spain.



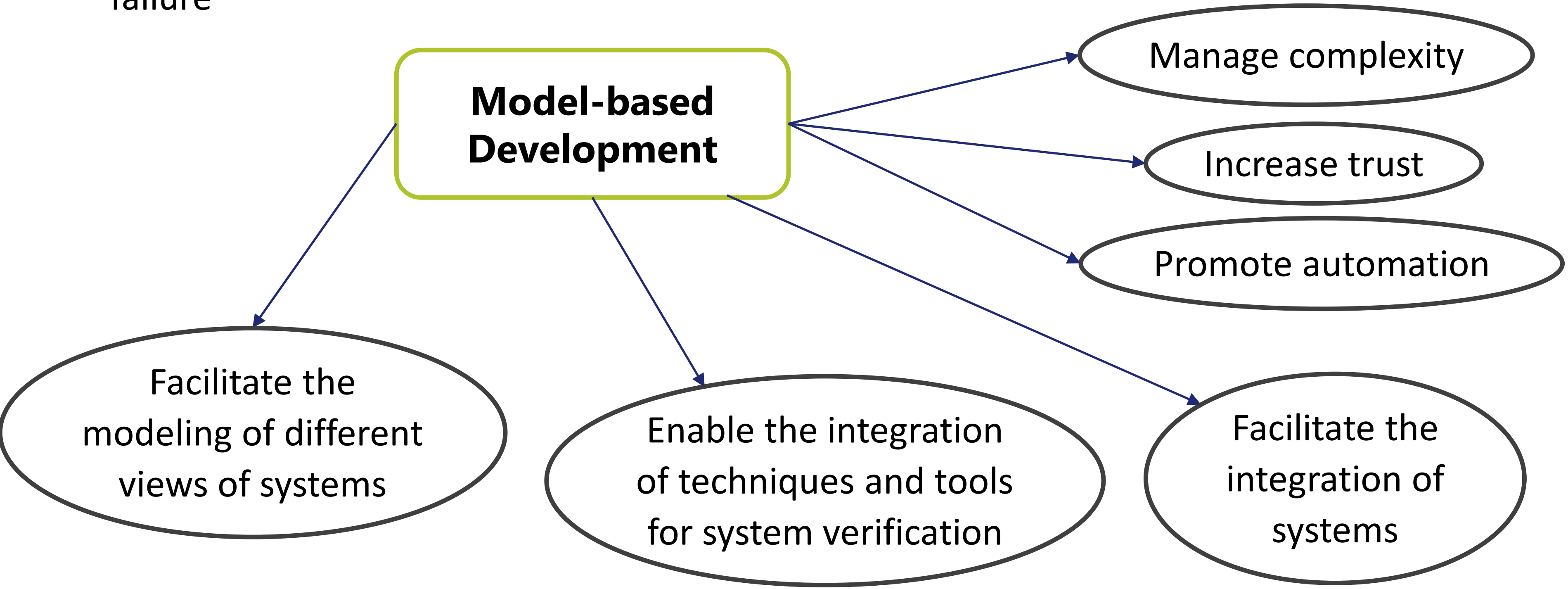
- A logistic service involving several robotic platforms from **ROBOTNIK** which need to collaborate to scan a given warehouse and to assist humans in a logistic domain.
- A fleet of robots **supports the movement of different loads in a warehouse**.





MODEL-BASED DESIGN

- Critical systems become more widespread.
- More rigorous development techniques are needed because of the potential cost of failure



UML & BIP

Model-based Approach

UML
[OMG]

- Standard graphical notation
- Visual diagrams and structural aspect

- Support using external code for specifying components’ behaviors

BIP
[VERIMAG]

- Textual representation for building formal models
- Efficient tool for analysis based on SMC

STATE OF THE ART

UML and B/Event-B

- **Modeling with UML, Translation to B**
- **Animation, Proof**

  - Siyuan, H., Hong, Z.: *Towards Transformation from UML to Event-B*. [QRS-C 2015]
  - Chehida, S., Idani, A., Ledru, Y., Kamel Rahmouni, M.: *Combining UML and B for the specification and validation of RBAC policies in business process activities*. [RCIS 2015]

UML and model checkers

- **Model Checking UML state machines**
  - Knapp, A., Merz, S., Rauh, C.: *Model Checking Timed UML State Machines and Collaborations*. [Formal Techniques in Real-Time and Fault-Tolerant Systems]
  - Zhang, S.J., Liu, Y.: *An Automatic Approach to Model Checking UML State Machines*. [SSIRI-C 2010]
- **UML & UPPAAL for modeling, and verifying component-based real-time systems**

  Muniz, A.L.N., Andrade, A., Lima, G.: *Integrating UML and UPPAAL for designing, specifying and verifying component-based real-time systems*. [Innovations in Systems and Software Engineering 2009] (TANGRAM TOOL)

UML and BIP

- **Safe Incremental Design of UML Architectures**

  Courbis, Anne-Lise et al. "*Safe Incremental Design of UML Architectures*." SEKE (2017).

UML and BIP

- **SMC (Scalable and less memory-intensive)**
- **External code for specifying different types of systems (stochastic, security components, etc.)**
- **Component-based approach (reusability and maintainability of components)**

APPROACH

System Modeling with UML

- System architecture (UML component Diagram)
- Components behavior (UML state machine diagrams)

UML to BIP

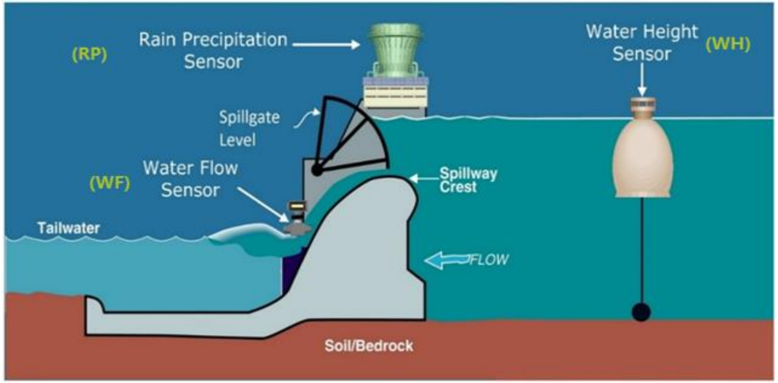
- Mapping rules
- UML_To_BIP prototype (Eclipse Acceleo)

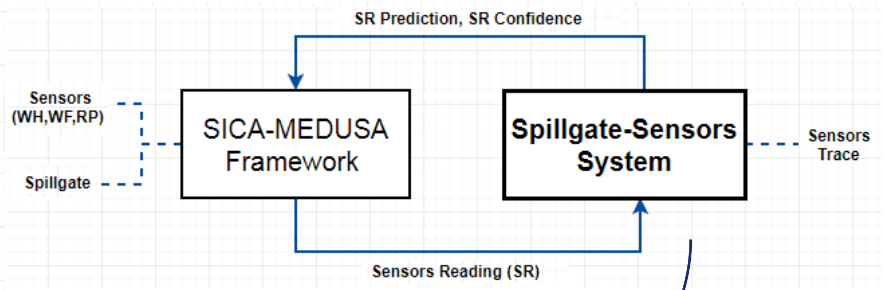
System Analysis with *SBIP*

- SMC (Statistical Model Checking)
- LTL properties

CASE STUDY

Dam of Cecebre (Spain)





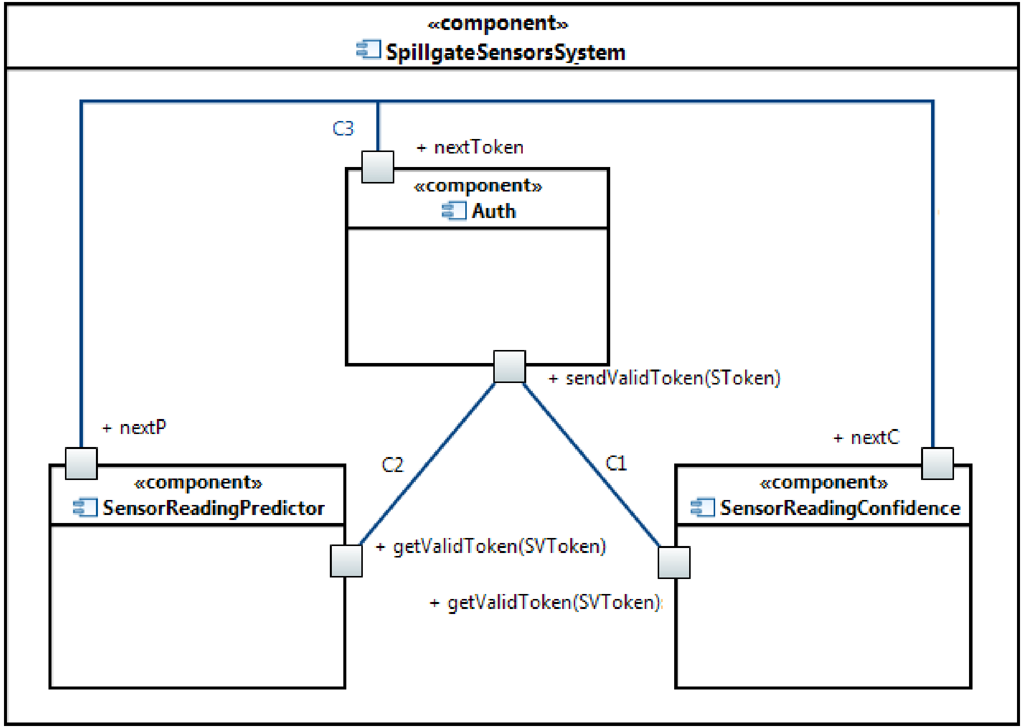
Sensors Data

- Used to Control the opening of the Spillgate.
- Ensure that the water does not reach a maximum level in the dam.

A trace of data recorded by each sensor per day since 1989 to 2016

- Authentication of sensors data
- Evaluating the confidence of sensors readings
- Making predictions

SYSTEM ARCHITECTURE



- **Auth** parses and authenticates the sensors data (JWT)
- **SensorReadingPredictor** predicts the sensor reading level (PSRL)
- **SensorReadingConfidence** evaluates the confidence of sensor reading (SRC)

COMPONENTS BEHAVIOR (Auth)

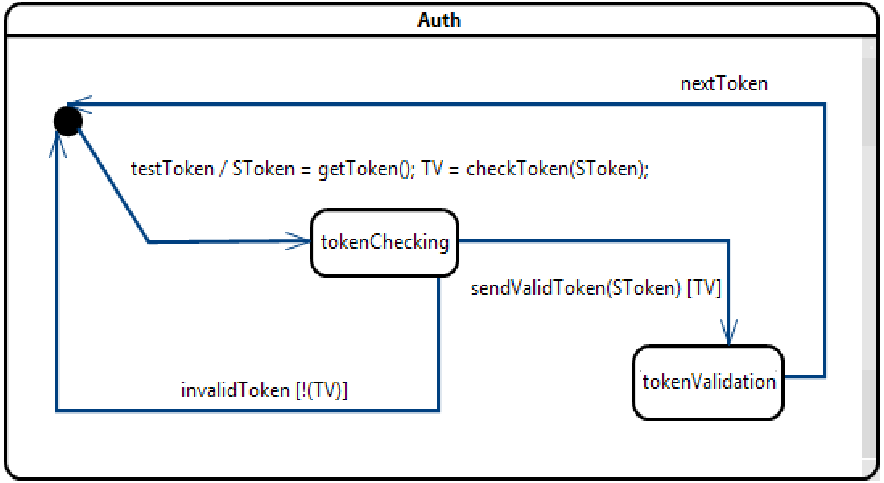
- JWT (JSON Web Token): an authentication system for data transferred between two parties

- Three parts : **Header**, **Payload**, **Signature**

```
{
"typ":"JWT",
"alg":"HS256"
}
{
    "iss": "Online JWT Builder",
    "exp": 1612268278,
    "aud": "Spillgate-Sensors-System",
    "sub": " SICA-MEDUSA",
    "sensorId": "12",
    "sensorType": "WH",
    "sensorReading": "34",
    "sensingTime": "15",
}
{
Signature
}
```

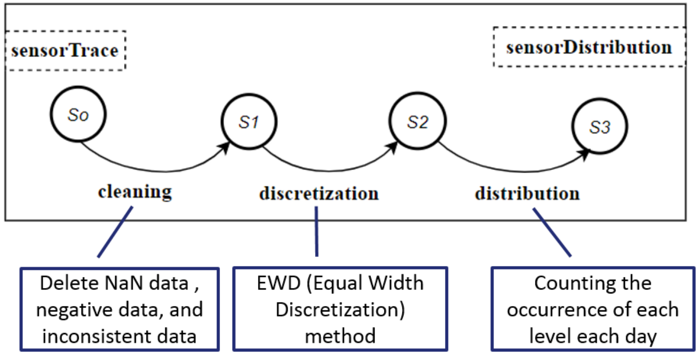
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI
1NiJ9.eyJpc3MiOiJPbmxpbmUgSldUI
EJ1aWxkZXIiLCJpYXQiOjE2MTA3MTMw
NzgsImV4cCI6MTYxMjI2ODI3OCwiYXV
kIjoiU3BpbGxnYXRlLVNlbnNvcnMtU3
lzdGVtIiwic3ViIjoiU01DQS1NRURVU
0EiLCJzZW5zb3JJZCI6IjEyIiwic2Vu
c29yVHlwZSI6IldIIiwic2Vuc29yUmV
hZGluZyI6IjM0Iiwic2Vuc2luZ1RpbW
UiOiIxNSJ9.OYe3I21MrLMt2xgnY5RT
9Iftmh0QWCaaN_7Y31I6s2I

- **checkToken** tests if token is well-formed and signature is correct



COMPONENTS BEHAVIOR (SensorReadingPredictor)

- **Cleaning**, **discretization** of sensor data into 5 levels
- Generation of **sensor distribution** for each day

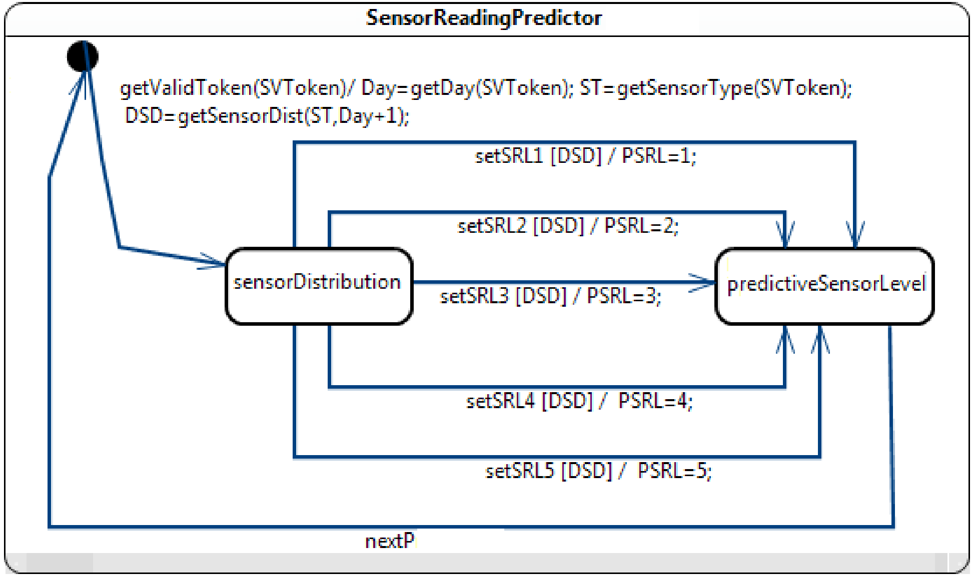


Chehida, S., et al.: Learning and analysis of sensors behavior in IoT systems using statistical model checking. Software Quality Journal (2021)

Day	L1	L2	L3	L4	L5
1	0.76	0.12	0.12	0	0
2					
3					
366					

Distribution

- **getSensorDist** selects distribution (DSD) of a given sensor (ST) for the next day (Day+1)
- Calculation of predictive sensor reading level (**PSRL**) based on DSD distribution.

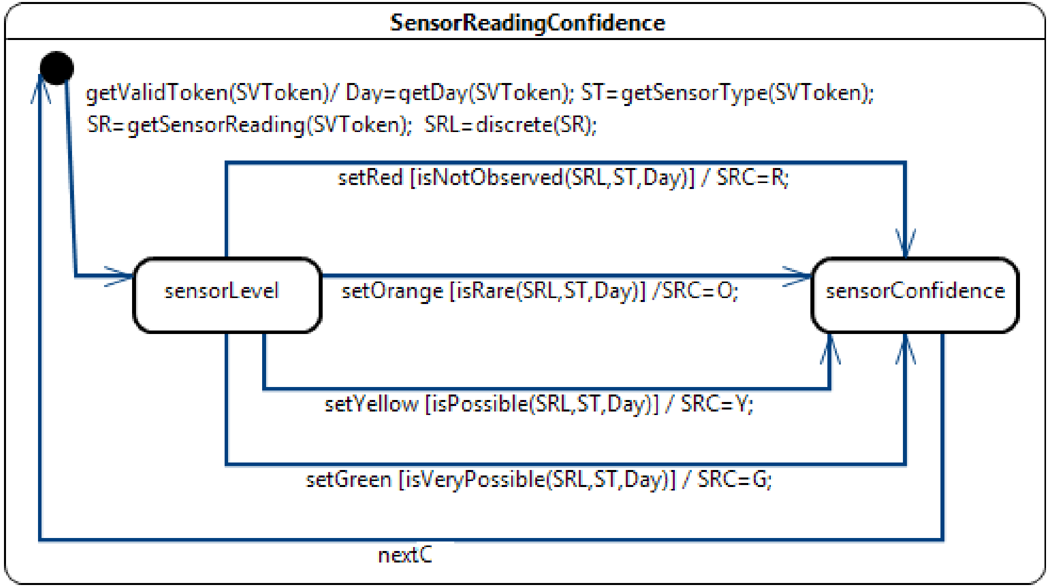


COMPONENTS BEHAVIOR (SensorReadingConfidence)

- ***discrete*** calculates the sensor reading level (SRL) from SR
- Calculation of sensor reading confidence (**SRC**) based on the results of functions:

- isVeryPossible
observed more than 21 times in 28 years for a given day
- isPossible
observed 3 to 21 times in 28 years for a given day
- isRare
observed once or twice within 28 years for a given day
- isNotObserved
never seen in 28 years for a given day

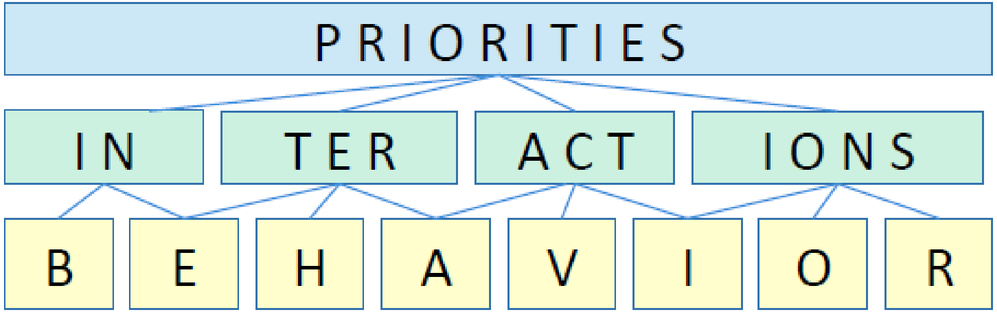
Day	L1	L2	L3	L4	L5
1	G	Y	Y	R	R
2					
3					
366					

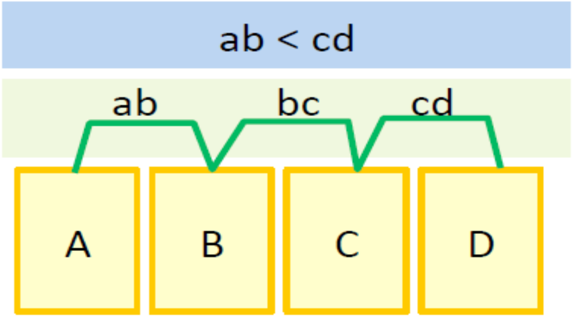


BIP

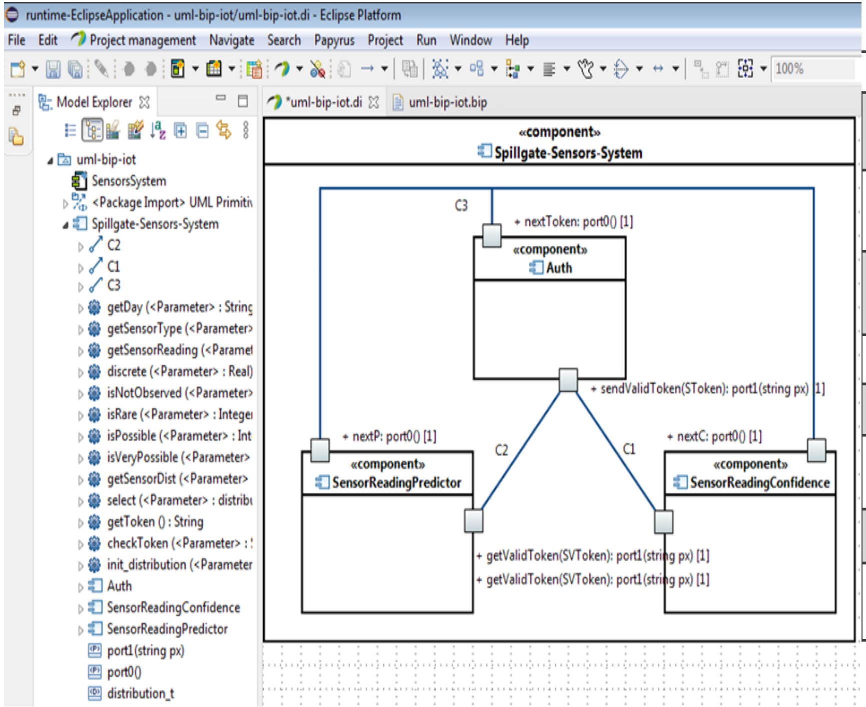
- BIP is component-based language for formal specification of critical systems

Components = layered composition of
– Behavior, atomic functional units (automata + code + timing constraints, stochastic semantics)
– Interactions, cooperation between actions of behavior
– Priorities, conflict resolution between interactions





UML TO BIP (1)

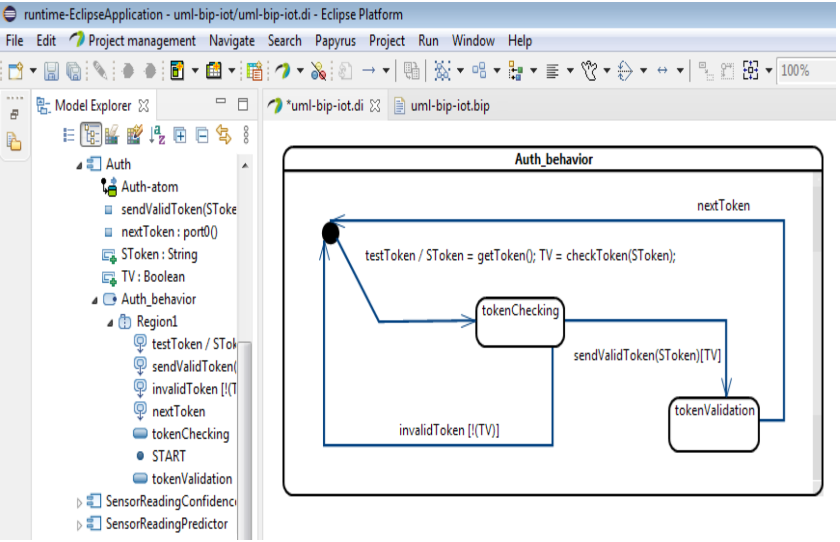


UML	BIP
Component Diagram	Package
Composite component	Compound
Atomic component	Atom
Operation	Extern function
Data type	Extern data type
Primitive type	Port type
Connection	Connector
Opaque bahavior	Connector type

Listing 1.1: BIP code of the main system component

```
@cpp(src="ext-cpp/senSysFunc.cpp",include="senSysFunc.cpp")
package SensorsSystem
//--- external functions and data
extern function string getToken()
extern function bool checkToken(string)
extern function int getDay(string)
extern function string getSensorType(string)
extern function float getSensorReading(string)
extern function int discrete(float)
extern function bool isNotObserved(int,string,int)
extern function bool isRare(int,string,int)
extern function bool isPossible(int,string,int)
extern function bool isVeryPossible(int,string,int)
extern function string getSensorDist(string,int)
extern function int select(distribution_t,int)
extern function distribution_t init_distribution(string,int)
extern data type distribution_t
// --- Ports and connectors types definitions
port type port0()
port type port1(string px)
connector type connect0(port1 p1, port1 p2)
define p1 p2 on p1 p2 down { p2.px= p1.px; } end
connector type connect1(port0 p1, port0 p2, port0 p3)
define p1 p2 p3 end
// --- Atom types definitions
atom type Auth()
//.. see Listing 1.2
end
atom type SensorReadingPredictor()
//.. see Listing 1.3
end
atom type SensorReadingConfidence()
//..
end
//--- Compound types definitions
compound type SpillgateSensorsSystem()
component Auth Auth1()
component SensorReadingPredictor SRP1()
component SensorReadingConfidence SRC1()
//--- Connector instantiations
connector connect0 C2(Auth1.sendValidToken, SRP1.getValidToken)
connector connect0 C1(Auth1.sendValidToken, SRC1.getValidToken)
connector connect1 C3(Auth1.nextToken, SRP1.nextP, SRC1.nextC)
end
end
```

UML TO BIP (2)



UML	BIP
State	Place
Transition	BIP External Port/ Internal Port
Property	BIP Data
Transition action	BIP expression
Transition guard	BIP guard expression (provided)

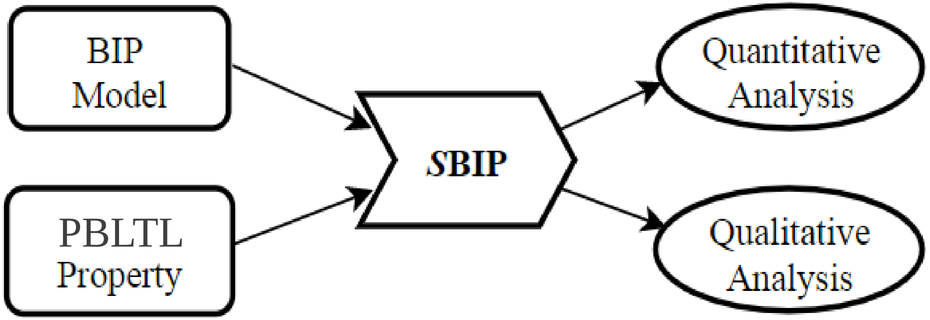
Listing 1.2: BIP code of Auth atom

```
atom type Auth()
data string SToken
data bool TV
port port0 testToken, invalidToken
export port port1 sendValidToken(SToken)
export port port0 nextToken
place START, tokenChecking,tokenValidation
initial to START
on testToken from START to tokenChecking
do {SToken = getToken(); TV = checkToken(SToken);}
on sendValidToken from tokenChecking to tokenValidation
provided (TV)
on invalidToken from tokenChecking to START provided (!TV)
on nextToken from tokenValidation to START
end
```

SMC ANALYSIS

*S*BIP framework

- Compile and simulate BIP models
- Automate the different statistical analysis
- Generate specific curves and/or plots of results



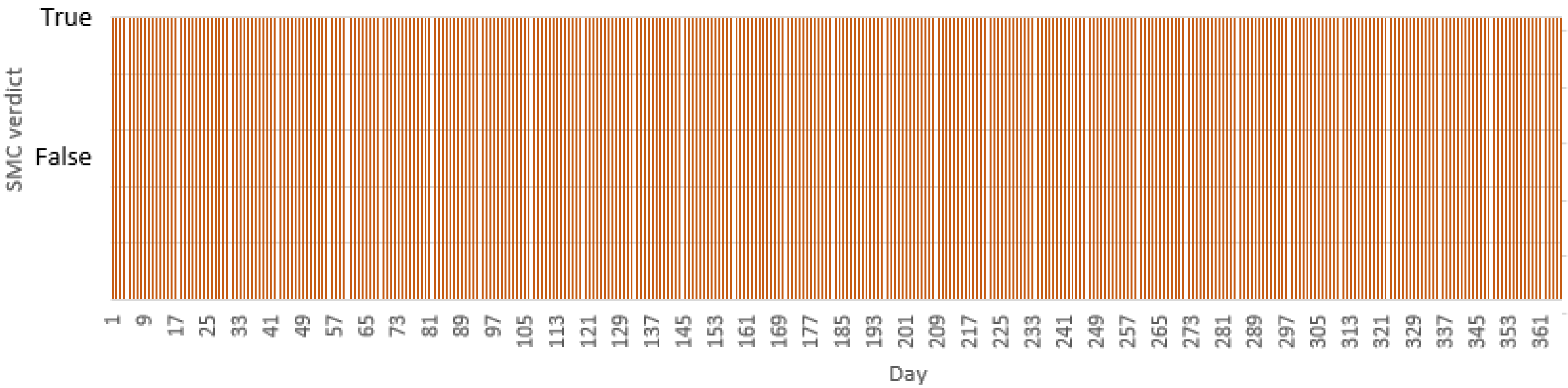
- PE (Probability Estimation)
- Estimate the probability that the system satisfies a given property

- HT (Hypothesis Testing)
- Check whether the probability of a given property being satisfied is greater or equal to a certain threshold.

PROPERTY 1

Check whether the tokens received by the system in 2017 are valid

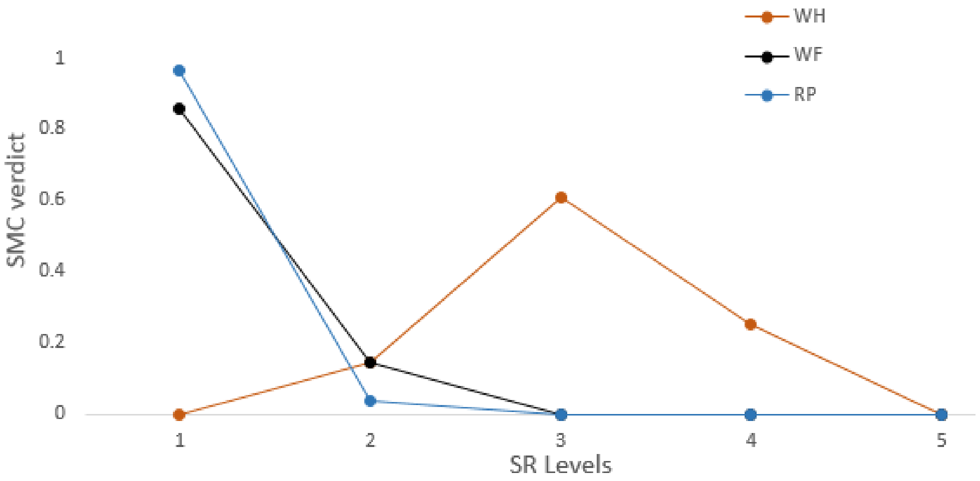
$$P_{=1}[F^{10000}\ (TV = true\ \wedge\ Day = D)];\quad D = 1 : 366 : 1;$$



PROPERTY 2

Compute the probability of WH, WF and RP levels sensors on January 27

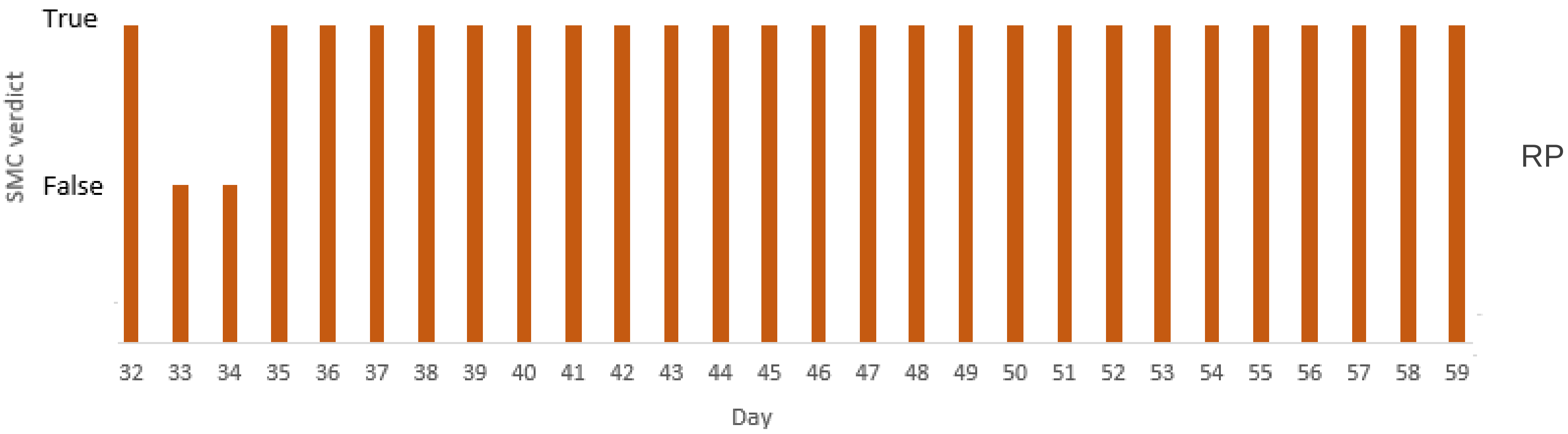
$$P_{=?}[F^{10000}\ (ST = T\ \wedge\ PSRL = L\ \wedge\ Day = 26)];\quad L = 1:5:1;$$
$$T \in \{WH, WF, RP\};$$



PROPERTY 3

Check the absence of rare levels of sensors readings in February 2017

$$P_{=1}[F^{10000}\ (ST = T\ \wedge\ \neg(SRC = O)\ \wedge\ Day = D)]; \quad D = 32:59:1;$$
$$T \in \{WH, WF, RP\};$$



CONCLUSION

Advantage of our approach

- Component-based modelling with reusability, and maintainability features.
- UML graphical models facilitate understanding the system.
- BIP provides a rigorous specification.
- Rigorous analysis using statistical model checkers that consume less memory and can check models with large state spaces.
- Support external code for specifying different types of components such as stochastic and security components.
- Prototype that automatically translates the UML models to BIP.

Future Work

- Validation of the translation of UML models into BIP

PUBLICATIONS

- Salim Chehida, Abdelhakim Baouya, Saddek Bensalem. (2021) " ***Component-Based Approach Combining UML and BIP for Rigorous System Design***". In Formal Aspects of Component Software. FACS 2021. Lecture Notes in Computer Science, vol 13077. Springer, Cham.
- Salim Chehida, Abdelhakim Baouya, Saddek Bensalem, Marius Bozga (2021) "***Learning and Analysis of Sensors Behavior in IoT Systems using Statistical Model Checking***". Software Quality Journal - Springer Journal.
- Abdelhakim Baouya, Salim Chehida, Samir Ouchani, Saddek Bensalem, Marius Bozga (2021) " ***Generation and Verification of Learned Stochastic Automata using k-NN and Statistical Model Checking***". Applied Intelligence - Springer Journal.
- Salim Chehida and Jean-Claude Tshilenge Mfumu. (2021) " ***Analysis and Prediction of Viral Infections using Statistical Model Checking*** ". In Proceedings of the 13th International Conference on Management of Digital EcoSystems (MEDES '21). Association for Computing Machinery, New York, NY, USA, 43–48.
- Salim Chehida, Abdelhakim Baouya, Saddek Bensalem, Marius Bozga (2020) "***Applied Statistical Model Checking for a Sensor Behavior Analysis***". In Quality of Information and Communications Technology. QUATIC 2020. Communications in Computer and Information Science, vol 1266. Springer, Cham.
- Tao, X., Conzon, D., Ferrera, E., Li, S., Götz, J., Maillet-Contoz, L., ... & Chehida, S. (2020). ***Model Based Methodology and Framework for Design and Management of Next-Gen IoT Systems***. In SAM IoT (pp. 80-90).
- Baouya, A., Chehida, S., Cantero, M., Millet, M., Bensalem, S., & Bozga, M. (2020). ***Formal modeling and simulation of collaborative intelligent robots***. In *European Conference on Service-Oriented and Cloud Computing* (pp. 41-52). Springer, Cham.

CONTACTS

SALIM CHEHIDA

RESEARCHER, UNIVERSITY OF GRENOBLE ALPES

Salim.Chehida@univ-grenoble-alpes.fr