INTERNSHIPS OFFERS FOR 2022

CEA-List, Université Paris-Saclay

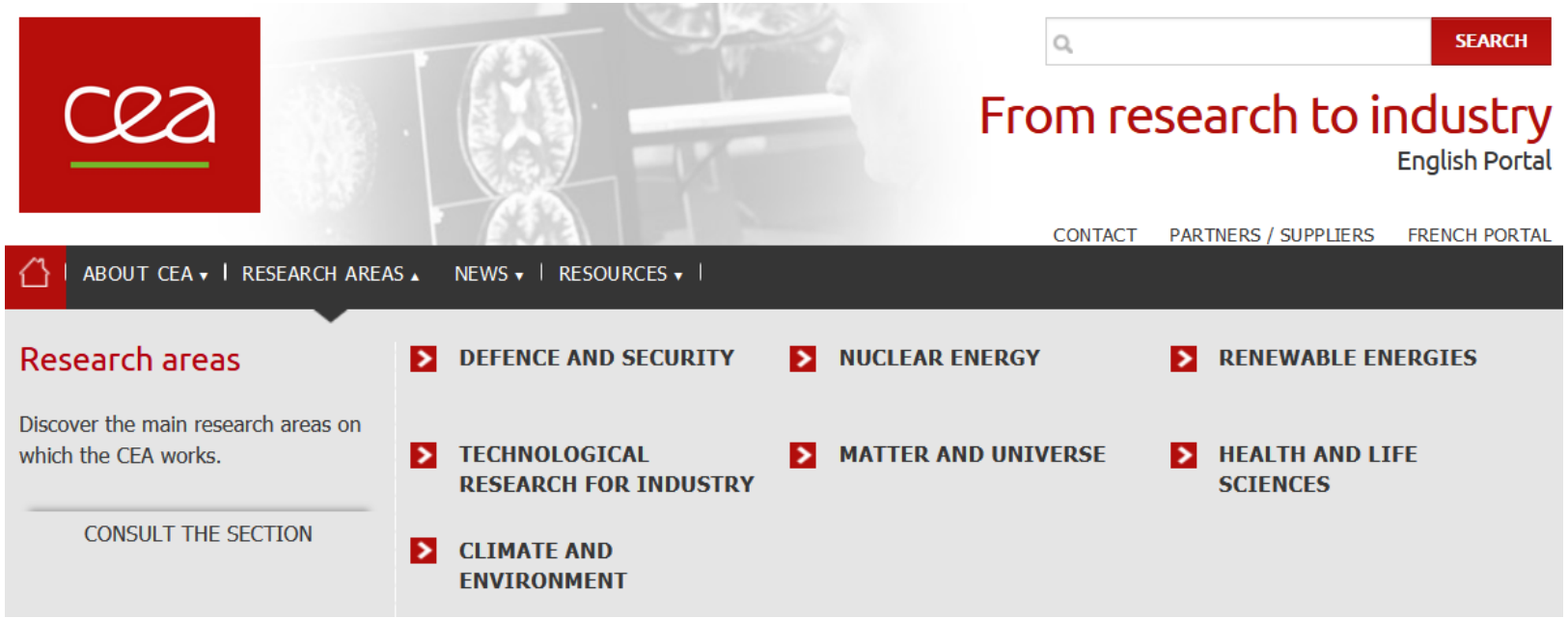
Master Réseaux – Sorbonne Université

LICIA – Trustworthy-, Intelligent-, Auto-organising Information Systems Laboratory

Contact: yackolley.amoussou-guenou@cea.fr | pierre.dubaillay@cea.fr | sara.tucci@cea.fr

CEA, FROM RESEARCH TO INDUSTRY

- **French Alternative Energies and Atomic Energy Commission**
- **CEA is a research institute**
- CEA is not only Atomic Energy

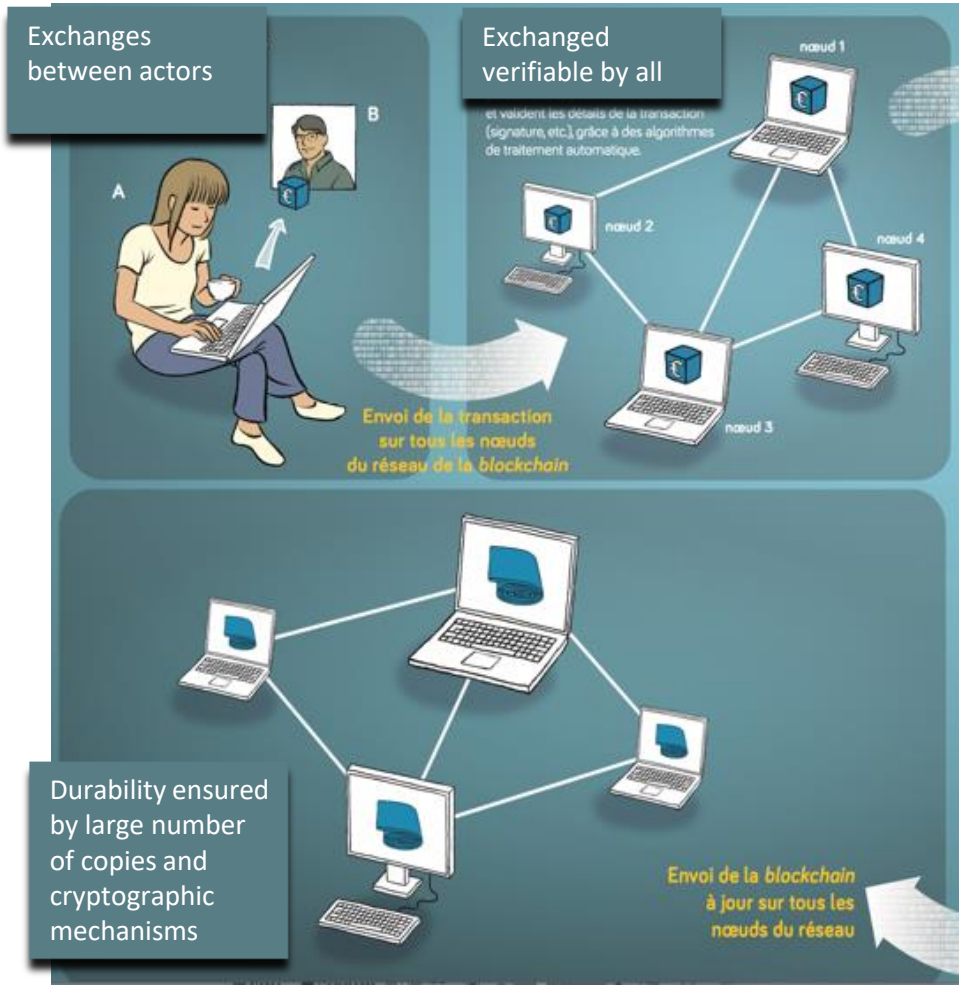


TRUST AT THE INTERNET SCALE: THE BLOCKCHAIN EXAMPLE

- **The blockchain is a distributed ledger registry that contains the history of all exchanges made between its users since its creation**

- **The exchanges are stored in the blockchain in a secure, tamper-proof and transparent way. Exchanges are *verifiables* by all**

- **A distributed ledger that can register any kind of interaction, event or (cryptographic) artefact**



At the Internet level, Blockchain can be seen as providing trust! It ensures agreement *without trust*

OUR MISSION: ENABLING TRUST AT THE INTERNET SCALE

In 2018, CEA creates a new laboratory dedicated to distributed software technologies (e.g. Blockchains)

Trustworthy-, Intelligent-, Auto-organizing Information Systems Laboratory

Head of Lab | Sara Tucci-Piergiovanni

CEA's Focus: GOING BEYOND (first gen) BLOCKCHAINS

- **Next generation blockchains: low-power, scalable, secure**
- **Advanced decentralised trusted services**
- **Methods and Tools for distributed technology design and validation**

THE LICIA TEAM

TUCCI
Sara
Cheffe de laboratoire

GÜRCAN
Önder

LANUSSE
Agnes

DEL POZZO
Antonella

AMOUSSOU-GUENOU
Yackolley

RIEUTORD
Thibault

GARCIA PEREZ
Alvaro

DUBAILLAY
Pierre
CI (Correspondant Informatique)

RALITERA
Tahina

DJARI
Mohamed-Aimen

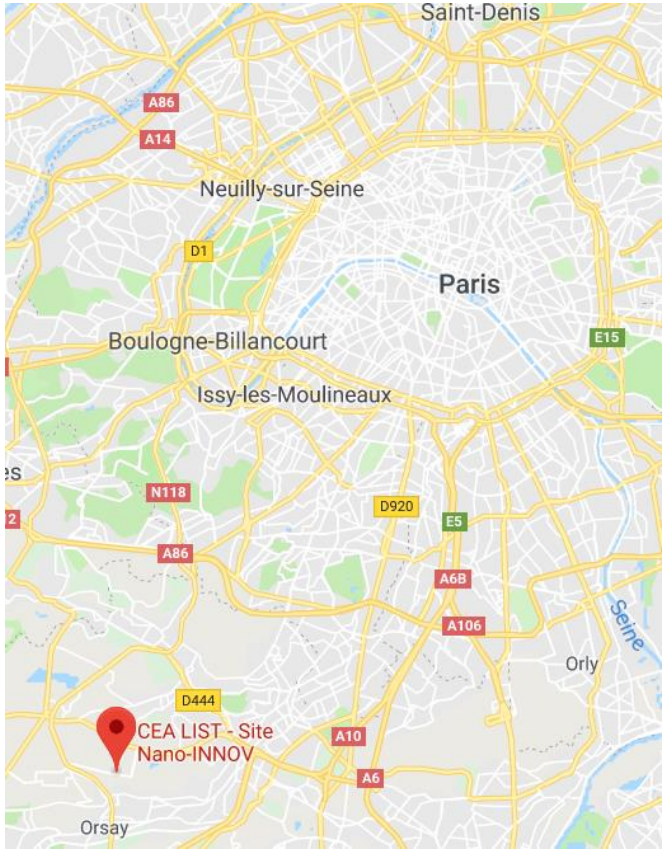
RAPETTI
Alexandre

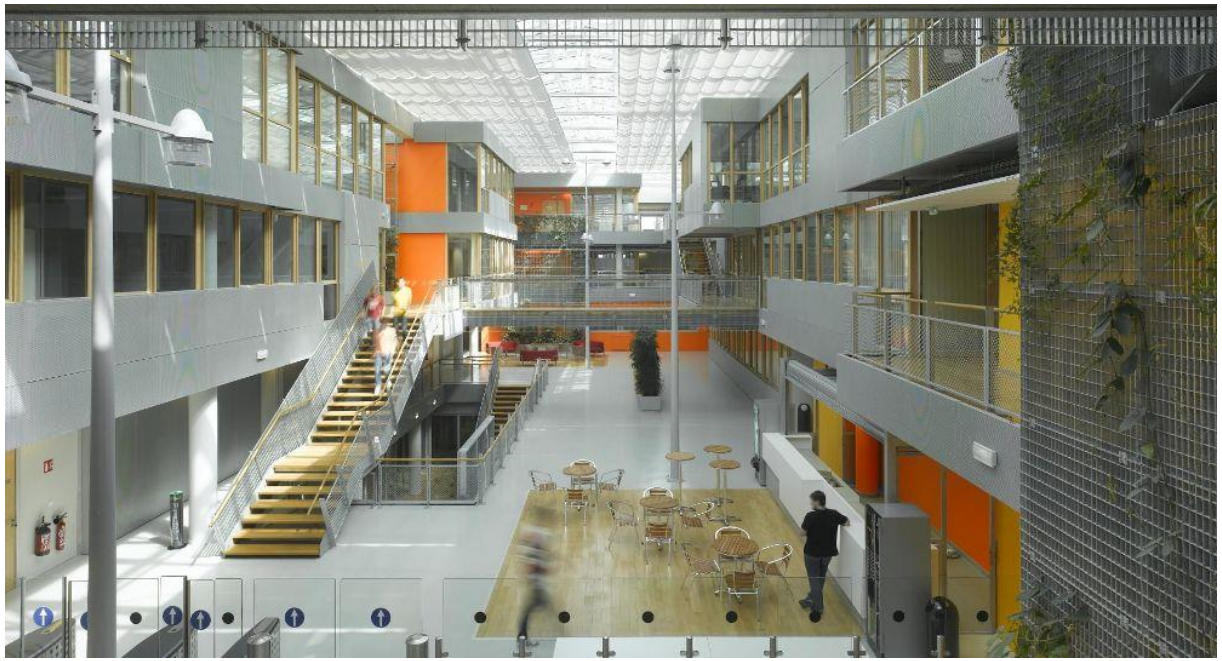
ROUSSILLE
Hector

PAVLOFF
Ulysse

BESSSOLES
Matthieu

WE ARE LOCATED ON THE “PLATEAU DE SACLAY”





ACTIVITIES OF THE LAB (MOSTLY AROUND BLOCKCHAINS)

Design

System specification

Formal specification (algorithms)

Implementation Code

Validation

Formal analysis

Theorems and proofs

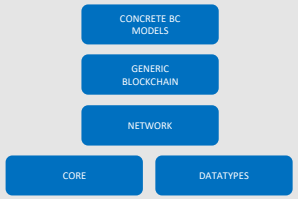
Simulation

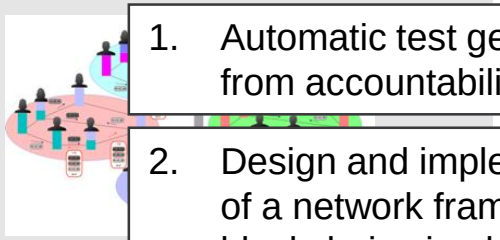
Simulator
Simulation data

CEA'S SKILLS AND TECH OVERVIEW

MAX Simulation Framework

- Dedicated Blockchain models
- Modularity and Extensibility
- Security Attacks testing support

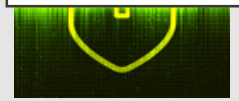




1. Automatic test generation from accountability module
2. Design and implementation of a network framework in a blockchain simulator

Design and validation of distributed ledger core technology

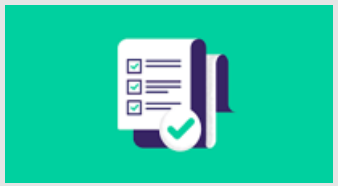
- Green and scalable algorithms
- Link with Physical world
- Proved Attack resilience
- Auditability



3. Blockchain algorithms for federated learning

Design and validation of advanced services

- Identity
- Traceability
- Compliance
- Incentives



4. Analysis of clients rentability in a distributed federated learning system using blockchains
5. Hierarchy of smart contracts for a decentralised escrow platform

IMPORTANT!!!!

Availabilities

The internships can start between February and April, depending on the candidate availabilities, and for a duration of about 6 months.

It is important to know that the instruction phase is **long**! For a beginning in February, the application should be made at before the end of October.

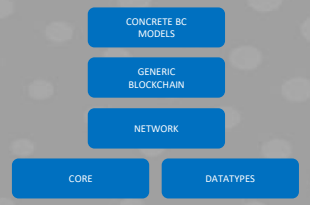
Disponibilité du poste

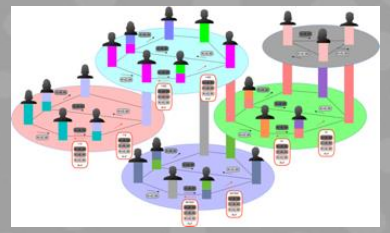
Début de stage possible de février à avril selon les disponibilités des candidat.e.s pour une durée d'approximativement 6 mois

Attention !!! **Les délais d'instruction de dossier sont long**, pour un début en février 2022 les dossier doivent être déposés au plus tard le 31 Octobre 2021.

MAX Simulation Framework

- Dedicated Blockchain models
- Modularity and Extensibility
- Security Attacks testing support



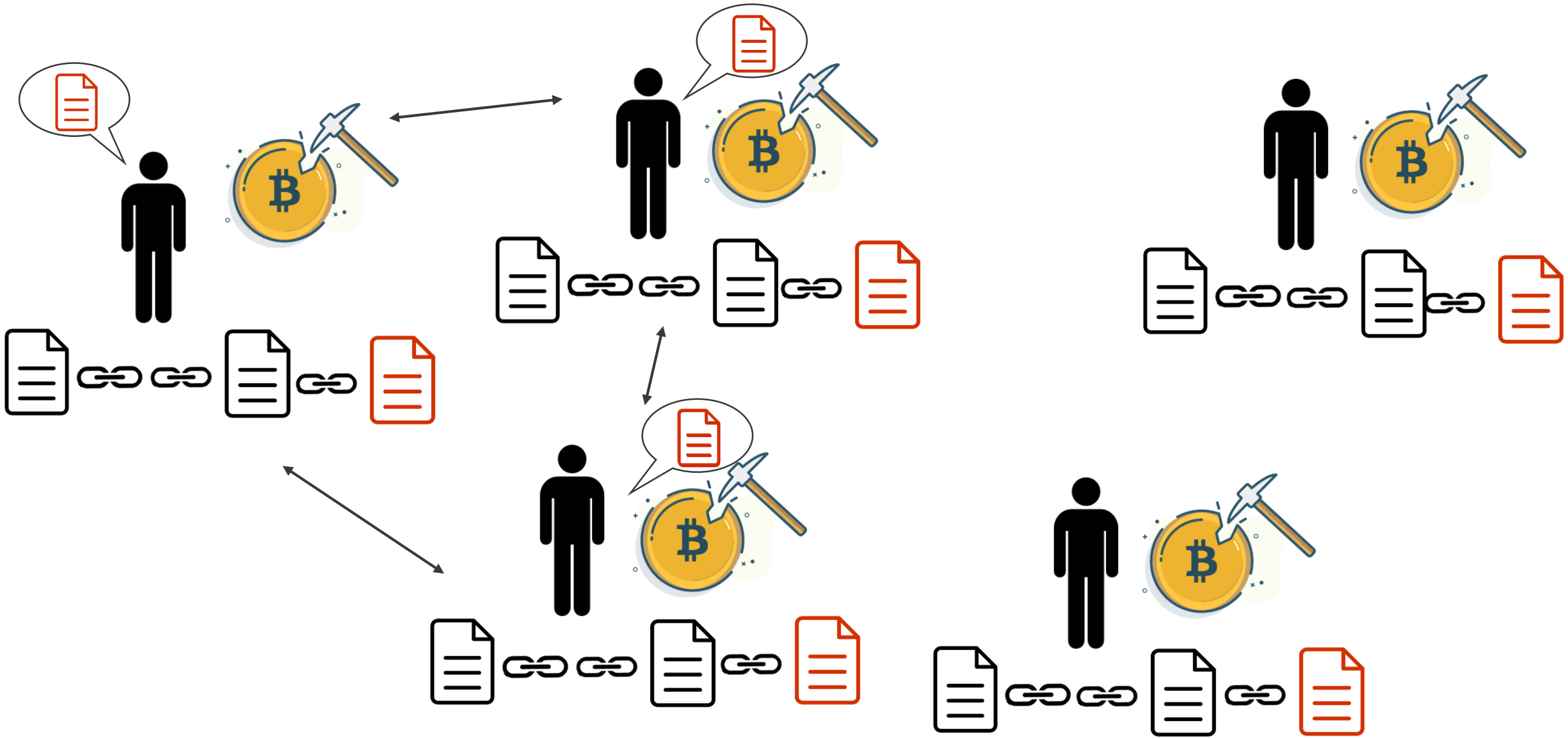


1- AUTOMATIC TEST GENERATION FROM ACCOUNTABILITY MODULE

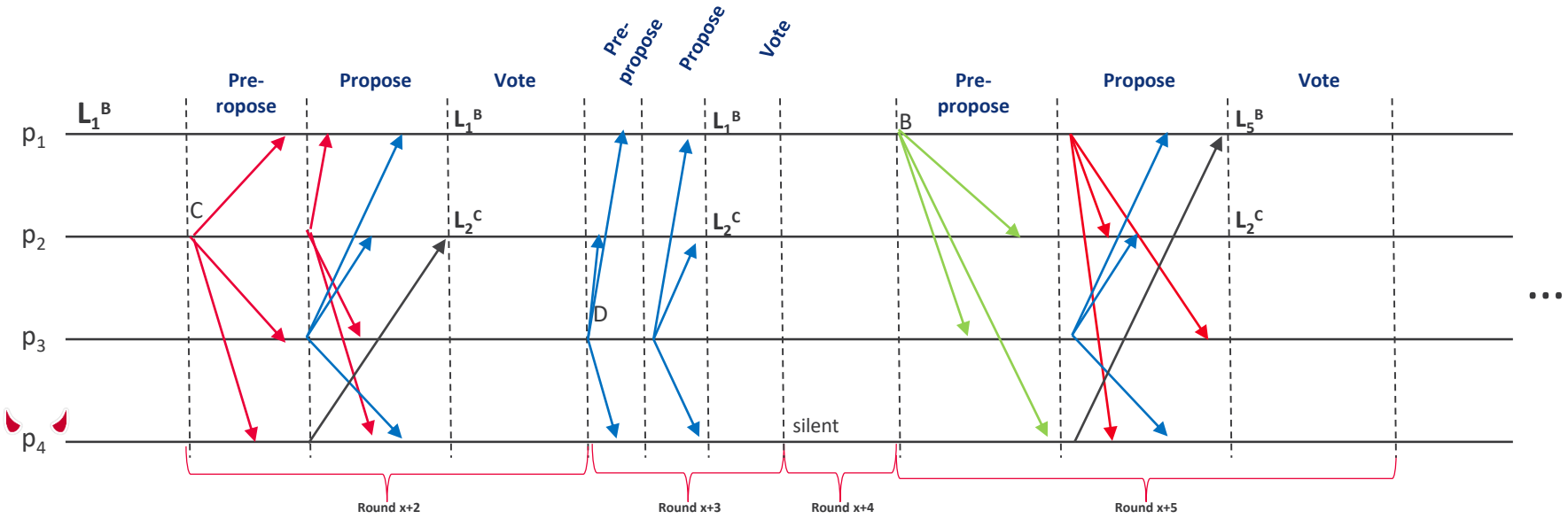
Commissariat à l’énergie atomique et aux énergies alternatives - www.cea.fr

antonella.delpozzo@cea.fr

CONSENSUS-BASED BLOCKCHAIN



AUTOMATIC TEST GENERATION FROM ACCOUNTABILITY MODULE



Processes might misbehave, they diverge from the protocol.

We want to detect them and construct a proof of their misbehavior.

AUTOMATIC TEST GENERATION FROM ACCOUNTABILITY MODULE

Tendermint pseudo-code (BFT protocol)

Algorithm 1 Tendermint consensus algorithm

```
1: Initialization:
2:     h_p := 0                                   /* current height, or consensus instance we are currently executing */
3:     round_p := 0                               /* current round number */
4:     step_p ∈ {propose, prevote, precommit}
5:     decision_p[] := nil
6:     lockedValue_p := nil
7:     lockedRound_p := −1
8:     validValue_p := nil
9:     validRound_p := −1
10: upon start do StartRound(0)
11: Function StartRound(round) :
12:     round_p ← round
13:     step_p ← propose
14:     if proposer(h_p, round_p) = p then
15:         if validValue_p ≠ nil then
16:             proposal ← validValue_p
17:         else
18:             proposal ← getValue()
19:         broadcast {PROPOSAL, h_p, round_p, proposal, validRound_p}
20:     else
21:         schedule OnTimeoutPropose(h_p, round_p) to be executed after timeoutPropose(round_p)

22: upon {PROPOSAL, h_p, round_p, v, −1} from proposer(h_p, round_p) while step_p = propose do
23:     if valid(v) ∧ (lockedRound_p = −1 ∨ lockedValue_p = v) then
24:         broadcast {PREVOTE, h_p, round_p, id(v)}
25:     else
26:         broadcast {PREVOTE, h_p, round_p, nil}
27:     step_p ← prevote

28: upon {PROPOSAL, h_p, round_p, v, vr} from proposer(h_p, round_p) AND 2f + 1 {PREVOTE, h_p, vr, id(v)} while
    step_p = propose ∧ (vr ≥ 0 ∧ vr < round_p) do
29:     if valid(v) ∧ (lockedRound_p ≤ vr ∨ lockedValue_p = v) then
30:         broadcast {PREVOTE, h_p, round_p, id(v)}
31:     else
32:         broadcast {PREVOTE, h_p, round_p, nil}
33:     step_p ← prevote

34: upon 2f + 1 {PREVOTE, h_p, round_p, ∗} while step_p = prevote for the first time do
35:     schedule OnTimeoutPrevote(h_p, round_p) to be executed after timeoutPrevote(round_p)

36: upon {PROPOSAL, h_p, round_p, v, ∗} from proposer(h_p, round_p) AND 2f + 1 {PREVOTE, h_p, round_p, id(v)} while
    valid(v) ∧ step_p ≥ prevote for the first time do
37:     if step_p = prevote then
38:         lockedValue_p ← v
39:         lockedRound_p ← round_p
40:         broadcast {PRECOMMIT, h_p, round_p, id(v)}
41:         step_p ← precommit
42:         validValue_p ← v
43:         validRound_p ← round_p
44: upon 2f + 1 {PREVOTE, h_p, round_p, nil} while step_p = prevote do
45:     broadcast {PRECOMMIT, h_p, round_p, nil}
46:     step_p ← precommit

47: upon 2f + 1 {PRECOMMIT, h_p, round_p, ∗} for the first time do
48:     schedule OnTimeoutPrecommit(h_p, round_p) to be executed after timeoutPrecommit(round_p)

49: upon {PROPOSAL, h_p, r, v, ∗} from proposer(h_p, r) AND 2f + 1 {PRECOMMIT, h_p, r, id(v)} while
    decision_p[h_p] = nil do
50:     if valid(v) then
51:         decision_p[h_p] = v
52:         h_p ← h_p + 1
53:         reset lockedRound_p, lockedValue_p, validRound_p and validValue_p to initial values and empty message log
54:         StartRound(0)
55: upon f + 1 {∗, h_p, round, ∗, ∗} with round > round_p do
56:     StartRound(round)
57: Function OnTimeoutPropose(height, round) :
58:     if height = h_p ∧ round = round_p ∧ step_p = propose then
59:         broadcast {PREVOTE, h_p, round_p, nil}
60:         step_p ← prevote
61: Function OnTimeoutPrevote(height, round) :
62:     if height = h_p ∧ round = round_p ∧ step_p = prevote then
63:         broadcast {PRECOMMIT, h_p, round_p, nil}
64:         step_p ← precommit
65: Function OnTimeoutPrecommit(height, round) :
66:     if height = h_p ∧ round = round_p then
67:         StartRound(round_p + 1)
```

Tendermint State Machine

Automatic test generator

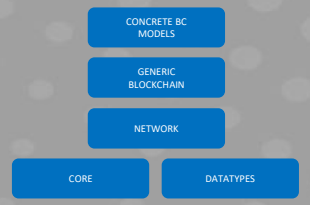
Test 1
Test 2
Test 3
....

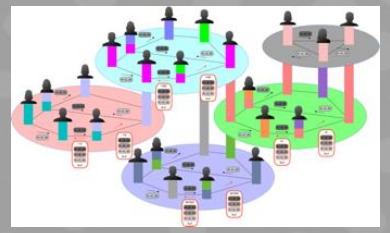
Goal :

The goal of this internship is to define and developp a tool for automatically generating test scenarios for the implementation of Tendermint. The candidate will have access to a formalization of the algorithm, as well as a first methodology allowing to define (pen and paper) tests.

MAX Simulation Framework

- Dedicated Blockchain models
- Modularity and Extensibility
- Security Attacks testing support





2- DESIGN AND IMPLEMENTATION OF A NETWORK FRAMEWORK IN A BLOCKCHAIN SIMULATOR

Commissariat à l’énergie atomique et aux énergies alternatives - www.cea.fr

pierre.dubaillay@cea.fr

DESIGN AND IMPLEMENTATION OF A NETWORK FRAMEWORK IN A BLOCKCHAIN SIMULATOR

Context

- Emergence of new blockchains
- Different consensus protocols : PoW, PoS, PBFT …
- Lack of perspective aboutsecurity in blockchain applications

Objective

- State of the art :
  - Communication paradigms
  - Network properties
  - Classical attacks
- Design and implementation of the network framework in our simulator
  - Influence the network properties
  - Swap communication paradigms
  - Set up attacks easily

DESIGN AND IMPLEMENTATION OF A NETWORK FRAMEWORK IN A BLOCKCHAIN SIMULATOR

Program schedule

- State of the art
- Design and modeling
- Implementation
- (Analysis)

Technologies

- Java 11
- Maven
- Git
- Gitlab
- IDE (choice of the candidate)

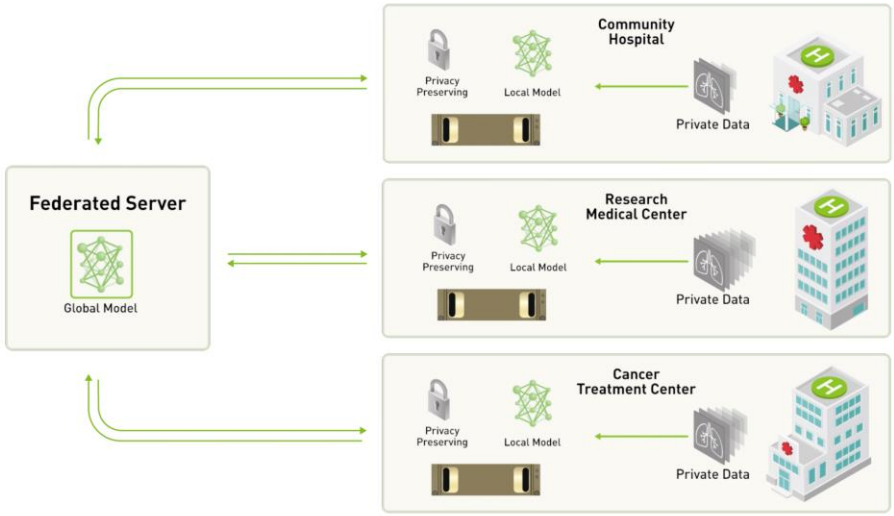
Design and validation of distributed ledger core technology

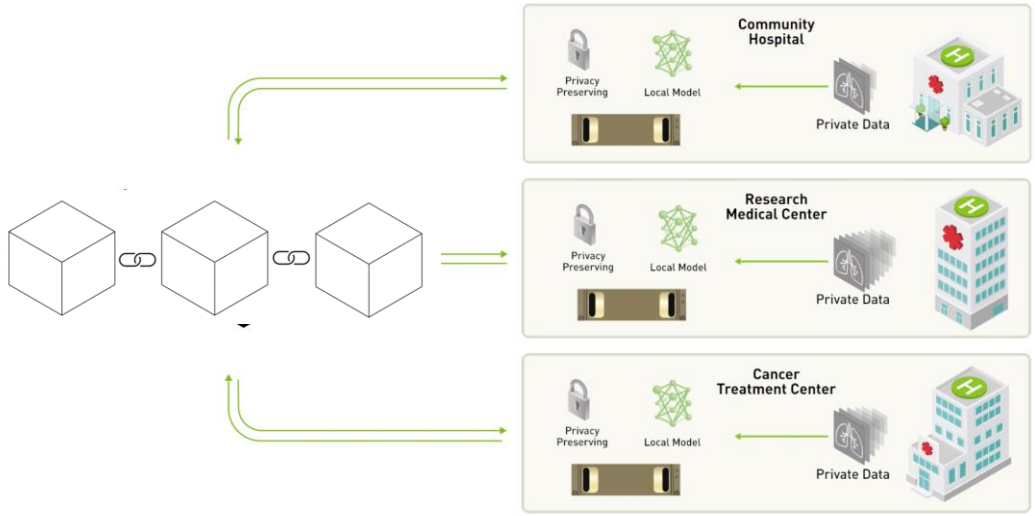
- Green and scalable algorithms
- Link with Physical world
- Proved Attack resilience
- Auditability

3- BLOCKCHAIN ALGORITHMS FOR FEDERATED LEARNING

alvaro.garciaperez@cea.fr

BLOCKCHAIN ALGORITHMS FOR FEDERATED LEARNING





Goal :

The goal of this internship is to help at the conception and the implementation of a blockchain platform for a distributed federated learning application. This work will be part of the CEA-List Carnot projet « Fantastyc ».

A high level specification and a preliminary prototype will be available to the candidate at the beginning of the internship.

Design and validation of advanced services

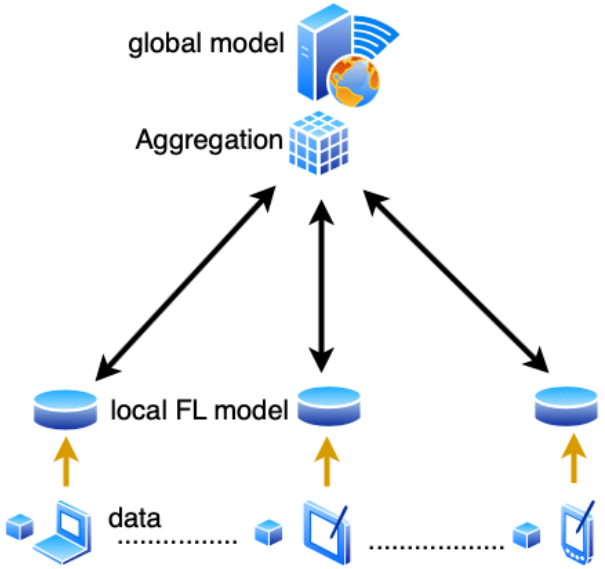
- Identity
- Traceability
- Compliance
- Incentives

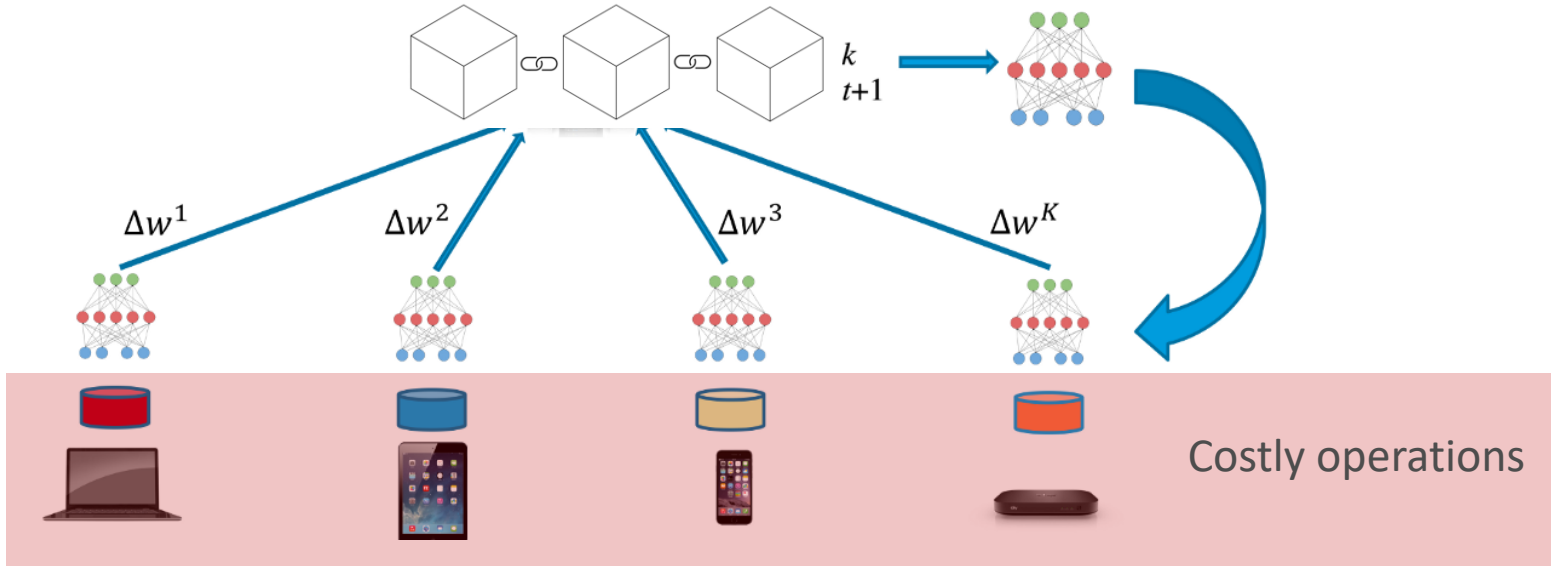
4- ANALYSIS OF CLIENTS RENTABILITY IN A DISTRIBUTED FEDERATED LEARNING SYSTEM USING BLOCKCHAINS

Commissariat à l'énergie atomique et aux énergies alternatives - www.cea.fr

yackolley.amoussou-guenou@cea.fr

ANALYSIS OF CLIENTS RENTABILITY IN A DISTRIBUTED FEDERATED LEARNING SYSTEM USING BLOCKCHAINS

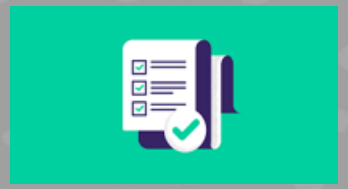




- **Understand and analyze the gain of the clients in such a system.**
- **Design incentive mechanisms allowing clients for gaining from doing these operations.**

Design and validation of advanced services

- Identity
- Traceability
- Compliance
- Incentives

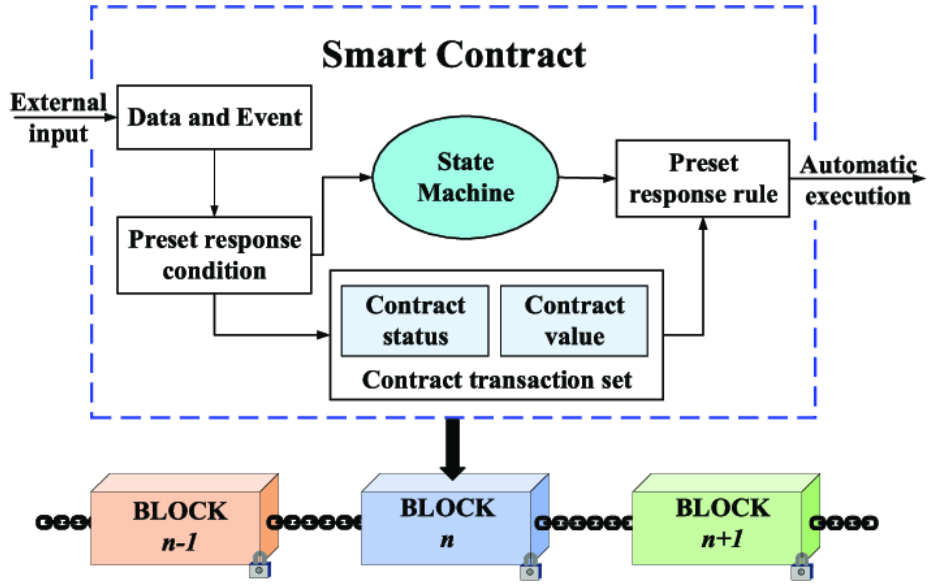


5- HIERARCHY OF SMART CONTRACTS FOR A DECENTRALISED ESCROW PLATFORM

alvaro.garciaperez@cea.fr

HIERARCHY OF SMART CONTRACTS FOR A DECENTRALISED ESCROW PLATFORM

HIERARCHY OF SMART CONTRACTS FOR A DECENTRALISED ESCROW PLATFORM

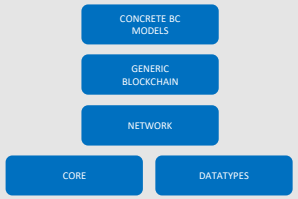


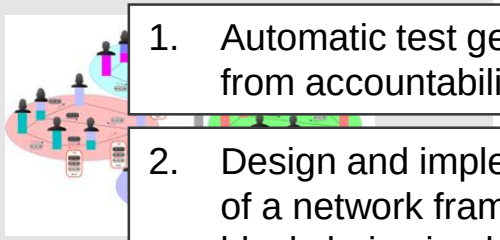
Goal :

The goal of this internship is to define a hierarchy of smart contracts and to define the life cycle of the smart contracts, allowing to have a continuous integration of new smart contracts for the DeFi application which is a decentralized escrow platform.

SUMMARY OF OUR OFFERS

- Dedicated Blockchain models
- Modularity and Extensibility
- Security Attacks testing support

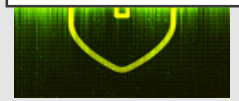




1. Automatic test generation from accountability module
2. Design and implementation of a network framework in a blockchain simulator

Design and validation of distributed ledger core technology

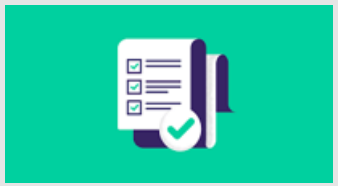
- Green and scalable algorithms
- Link with Physical world
- Proved Attack resilience
- Auditability



3. Blockchain algorithms for federated learning

Design and validation of advanced services

- Identity
- Traceability
- Compliance
- Incentives



4. Analysis of clients rentability in a distributed federated learning system using blockchains
5. Hierarchy of smart contracts for a decentralised escrow platform

list
cea tech

DE LA RECHERCHE À L'INDUSTRIE

THANK YOU | MERCI

Yackolley Amoussou-Guenou

Pierre Dubaillay

LICIA – Trustworthy-, Intelligent-, Auto-organizing Information Systems Laboratory