IOT Security

Master in Computer Engineering for the Internet of Things
University of Calabria

Giuseppe M. L. Sarne’

Department of Psychology

University of Milan Bicocca

Prof. Giuseppe M. L. Sarnè
giuseppe.sarne@unimib.it
sarne@unirc.it

Department of Psychology
University of Milan Bicocca

Presentation for the exclusive use of students of the course “IoT Security” within the MD in “Intenet of Things” of the University of Calabria for the Academic Year 2020/21

Summary IoT Security: Trust and Blockchain module

- Trust
- Blockchain

Summary Trust

- Introduction
- Trust Notions
- Social Networks
- Social Trust
- Ego-networks

Introduction

Introduction

- The concepts of Trust and Reputation always referred to a social structure connecting social actors (having different nature) among which occur some type of social relationships
  - The social actors can be: real and/or virtual entity.
  - The social actors can be: Individuals, Groups, Organizations, Governments, agents, …
  - The nature of the social relationships can be: Interactions, Trades, Values, …
- The social nature of trust and reputation have been widely investigated.
- These studies provided methods for analyzing social relationships in order to identify local and global patterns, locate influential entities, and examine the underlying dynamics.

Introduction

- These analysis have an intrinsically interdisciplinary nature involving several scientific sciences, among which psychology, sociology, statistics, graph theory, economy, computer science and so on.

- Internet modified traditional relationships on three main dimensions:
  - Space: social interactions are not more connected to physical places
  - Time: social interactions can be carried out in a "time independent" fashion
  - Size: the potential individual social sphere is incomparably greater

- The coming of the Internet make easier to have an overwhelming number of social interactions with (almost) everyone in everyplace and in every time.

Introduction

- Currently, it is possible to exchange sensible information, provide services, make trade arrangements and other activities, which can be affected by the unsafe open nature of the Web.
- Similarly to other Web activities, it is necessary to implement suitable strategies in order to assure authenticity and mutual respect, as well as to avoid (or limit) risks for data, services and so on.
- Two different and complementary approaches exist:
  - Cryptographic systems
  - Trust systems

Trust and Reputation Notions

Trust

- Trust is multi-disciplinary concept that has no single meaning and is subject to high semantic ambiguity.
- The concept of Trust assumes different meanings in relation to the perspective with which it is observed. The perspective depends on the basis of its application context (i.e., sociology, phycology, economics, computer science and so on).
- Moreover, the concept of Trust is generally Multi-dimensional (i.e., rarely a single parameter is enough to define it).

What is Trust?

- Among the various "philosophical" approaches within a rational action we can distinguish, for example, two forms of Trust :
  - Strategic (purpose-oriented): who trusts another entity implements a strategic decision that, based on the information available and the own attitude to risk, reflects the expectations on the entity's behavior
  - Moralistic (ethically oriented): whoever trusts a person implements a moral decision that, assuming others to be trustworthy (since the majority share the same values) reflects expectations about how they should behave

What is Trust?

- Technological development necessarily requires significant daily acts of Trust towards and from virtual systems that are now necessary to carry out common work and/or relationship activities.
- Nowadays also the personal Trust is mediated by the virtual Trust as personal relationships, even with strangers (i.e., horizontal relationships), are mediated through technological tools. This implies both strategic and moral Trust activities
- Every day we carry out actions that imply a basis of trust where granting and obtaining trust can be understood as processes that can simplify our interactions with the surrounding environment.

What is Trust?

- Trust directed towards people is different from the one directed towards virtual systems (moreover, it is now necessary to consider also the Trust between *virtual objects* as in the case of IoT devices).

- The personal Trust is based "on prolonged knowledge that has consolidated the guarantees that make one reliable in the eyes of the other."

- The *virtual* Trust does not require prolonged interactions over time and is born at the intersection of personal needs and systemic interests (e.g., the use of an expert system to solve a problem).

What is Trust?

- In any case Trust is an essential component of any iteration, generally defined as a relationship between a Trustor (subject/object that gives trust to a target) and a Trustee (subject/object that receives trust).

- Trust is essentially a form of knowledge, about someone or something, capable of generating reasonable hope in a positive or, equivalently, negative behavior

- In some respects Trust maintains a balance between knowledge and ignorance (i.e., between what we know and what we do not know about a situation), allowing us to act even where there is not a full knowledge of the context (through a considered hope in an expected behavior).

What is Trust?

- In other words, a "systemic trust" must be created on the basis of which it can be assumed with an adequate degree of reliability that after a certain number of positive or, on the contrary, negative experiences, the system persists in its already known behavior (all things being equal).
- The uncertainty caused by the unknowability of the future is "overcome" by the rational evaluation of the consequences (risks) of one's actions.
- In other words, every time we rely on a Trust assessment we are taking for granted a certain degree of risk

What is Trust?

- Finally, Trust is a founding element of the Theory of Social Capital, and similarly Trust accumulates over time with the succession of iterations, but it must not be wasted and, to this end, the trustee must commit to be trustworthy.
- As far as we are concerned, Social Capital can be assumed as the set of resources (material or symbolic) that an entity (individual or collective) can obtain from the network of social relations.
- In the individualistic vision, Social Capital is an individual resource, which produces private goods, for the benefit of the individual or a specific group (Bourdieu; Coleman), other approaches define it with reference to the community.
- In other words, the Trust is strongly characterized by social aspects

Trust and Security

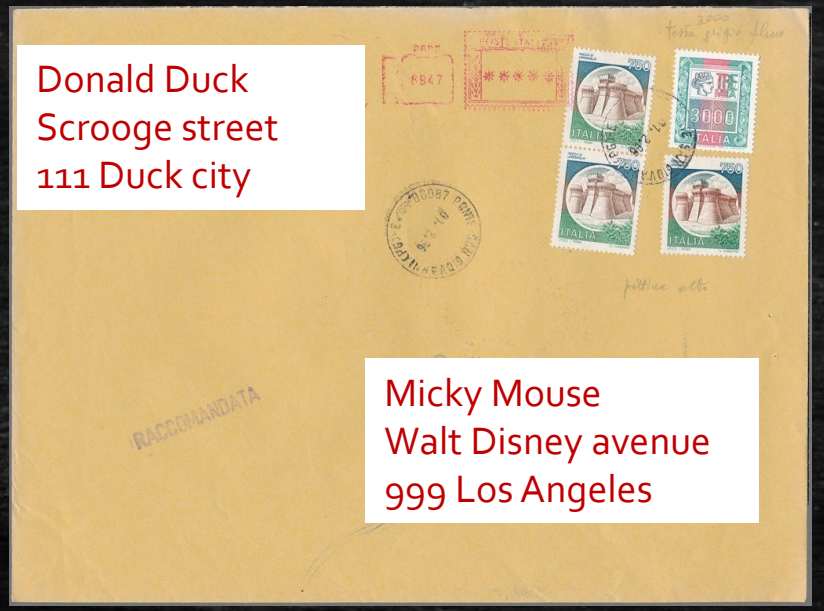
- There are two different and complementary approaches to provide "security" to a system whether it is real or virtual.

- In detail, we refer to:
- Trust-based techniques
- Cryptographic techniques (symmetrical and asymmetrical cryptography)

- These two techniques pursue completely different objectives. Consequently, it is profoundly erroneous to define Trust-based systems as weak forms of security compared to cryptographic techniques, even if Trust is sometimes referred to in such terms being "only" a form of inductive knowledge.

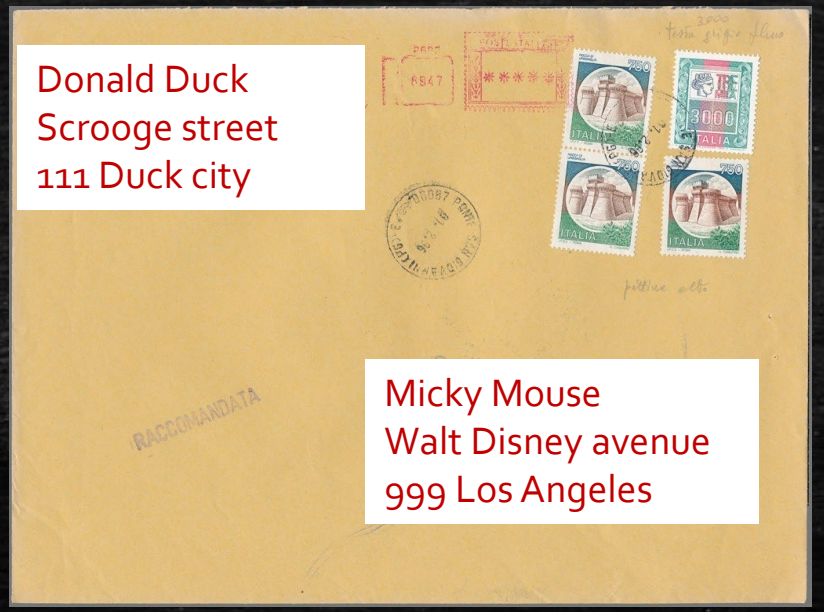
Criptography advantages

- Cryptography has two advantages (i.e, it allows to guaranty the identity and the communication privacy) that, iconically, we can represent with a closed envelope with the addresses of the recipient and the sender



Criptography advantages

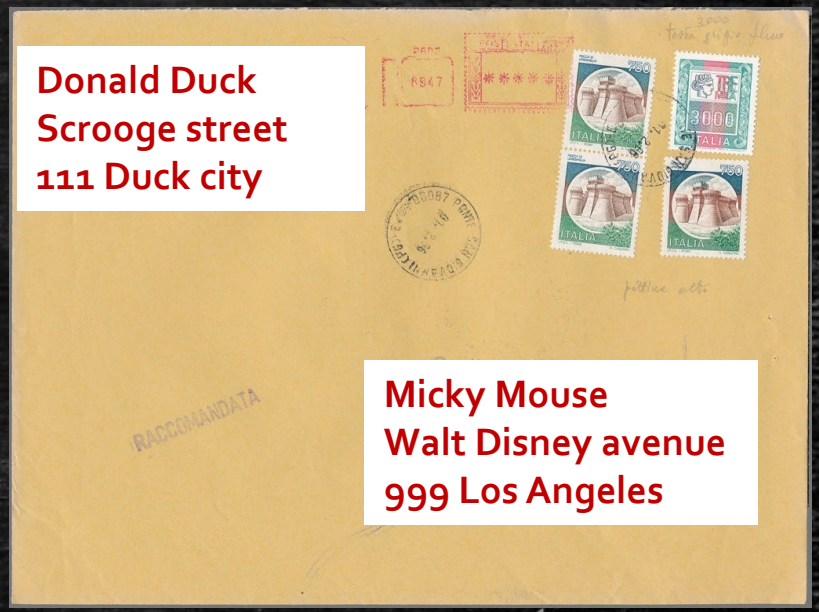
- Cryptography can guarantee:
  - The identification of one or more parties involved in the processes (depending on the parties in possession of cryptographic keys), similarly to the address tags of the sender and the receiver
  - Confidentiality of communications, like a closed envelope



Criptography advantages

- Encryption does not provide the recipient with any kind of information about the possible content of the letter exactly as when the sender is unknown and the envelope does not bear any markings (for example, commercial).

- Remaining on the example of the letter, the Trust-based systems are similar to the case of a closed envelope where the sender has a habit of letters sent to the same recipient (e.g., if I receive a letter from the company that supplies me with electricity, then I can easily imagine its content).



Trust vs Cryptography

- The added value that is given to cryptography is that it is considered inhibitive compared to a "bad" behavior just because it can be identified with certainty through the cryptographic system.

- Undoubtedly, in itself this is not hostile to "bad behavior" and when you have to take some action with a stranger it is equally important to assess the risk factor against which a cryptographic system is totally blind.

- With respect to confidentiality, the Trust overturns the concept. It is no longer keeping the information mechanically confidential, but knowingly deciding who to disclose the information to.

An example

- Consider eBay, the fact that the seller is known and identified (through a third party - in this case by a bank credit institution -) does not exclude possible fraud. Differently, the Trust system (a Reputation system) provides information on past behaviors, but does not give the certainty that they will be repeated.
- In other words, a cryptographic system is similar to an identity card that tells us who the seller is, while a reputation system tells us what others think of the seller.

An example


PASS Proof of Age Card
CITIZENCARD
YOTI
Name
LartStew
DoB
02 Oct 1999
PASS
166 4567 8901
NPCC
National Police Chiefs' Council
sia
Security Industry Authority
18+


200
190
180
170
160
150
140
47A589 POLICE DEPARTMENT
MUGSHOT
HT 168 WT 136 DOB 07.12.1989

Trust definition

- In many relationships, Trust is based on a combination of judgement or opinion derived from face-to-face meetings or recommendations of colleagues, friends and business partners.

- Webster dictionary states that Trust is:
  - an assumed reliance on some persons or things. A confident dependence on the character, ability, strength or truth of someone or something.
  - a charge or duty imposed in faith or confidence or as a condition of a relationship.
  - the placed confidence (in an entity).

- Oxford English Dictionary (1971), is defined as "confidence in or reliance on some quality or attribute of a person or thing, or the truth of a statement"

Trust context

Psychology

Trust is a psychological state of the individual, where the trustor risks being vulnerable to the trustee due to his/her positive expectations about the trustee's intentions or behavior [Rotter, 1967]. It reflects cognitive, emotive and behavioral aspects.

Sociology

When Trust is referred to an individual it can be assumed as a "bet about the future contingent action of a trustee", while when Trust is referred to a group it can be assumed as a "collective psychological state of the group".

Trust context

Probability

"Trust (or, symmetrically, distrust) is a particular level of the subjective probability with which an agent assesses that another agent or group of agents will perform a particular action" [Gambetta, 2000].

Management

Trust is a "the belief that the decision makers will produce outcomes favorable to the person's interests without any influence by the person" [Driscoll, 1978].

Computer Science 1

Trust is a "subjective expectations an entity has about another future expectations" [Mui, 2003].

Trust context

Computer Science 2

Trust is a "the expectations that a device or a system will faithfully behave in a particular manner to fulfil its intended purpose" [Yao et al., 2010].

Multi-Agent Systems

Trust is tightly referred to a delegation concept [Castelfranchi e Falcone].

Trust typology

Computational

"A rational actor will place Trust … if the ratio of the chance of gain to the chance of loss is greater than the ratio of the amount of the potential loss to the amount of the potential gain" [Coleman, 1990].

Relational

Trust is the result of repeated interactions with the same counterpart in order to acquire knowledge about his reliability and dependability [Rousseau, 1998].

Emotional

Security and comfort in relying on a trustee [Kuan and Bock, 2005].

Trust typology

Cognitive

According to the social capital theory, three form of social capital affect cognitive Trust: information channels, norms and sanctions, the trustees' obligations to the trustor [Coleman, 1988].

Institutional

The presence of an institution enable environments which encourage cooperation between members and penalize misbehaviors [Lewis and Weigert, 1985].

Dispositional

People develop generalized expectations about the trustworthiness of other people [Rotter, 1971].

Trust typology

Social

It includes three form of social Trust:

- Trust between members of the network
- Trust between a member and the provided service online
- Trust between a member and the service provider.

Given the multiple facets of the Trust nature, several aspects can affect the last two forms of social Trust, in spite of their apparent irrelevance.

Trust property

Context

Trust is context specific. Therefore, more trust values are referred to the same entity, one for each environment where it acts.

Persistent

Trust is referred to time persistent entities because it has a meaning only for future interactions.

Dynamic

Trust can increase or decrease with new experiences (interactions or observations) [Staab et al., 1971]

Trust property

Propagative

Trust is propagative (different from transitive) and assume more relevance as well as it is spread into a community

Non Transitive

Trust is generally non-transitive [You and Singh, 2000]. Therefore, if A trusts B and B trusts C, then it does not imply that A trusts C.

Composable

Propagation of Trust (and Distrust) along social chains allows a member to form some trust also for member non directly connected. This property is important in presence of large SN.

Trust property

Subjective

In general, Trust is tightly subjective. Therefore, in composing different trust sources, it can happen to have contradictory evaluations.

Asymmetric

Trust is asymmetric and if A trust B, it does not imply that B trusts A (at all or in part)

Self-reinforcing

Members act positively with other members whom they trust and, obviously, positive interactions reinforce mutual Trust.

Trust properties

Event sensitive

A single event can destroy a long time Trust [Nepal et al., 2010].

Trustor and Trustee properties

Trustor Properties

- objective: Criteria and/or policies specified by the trustor for a Trust decision
- subjective: Predisposition and willingness to trust (propensity to risk)

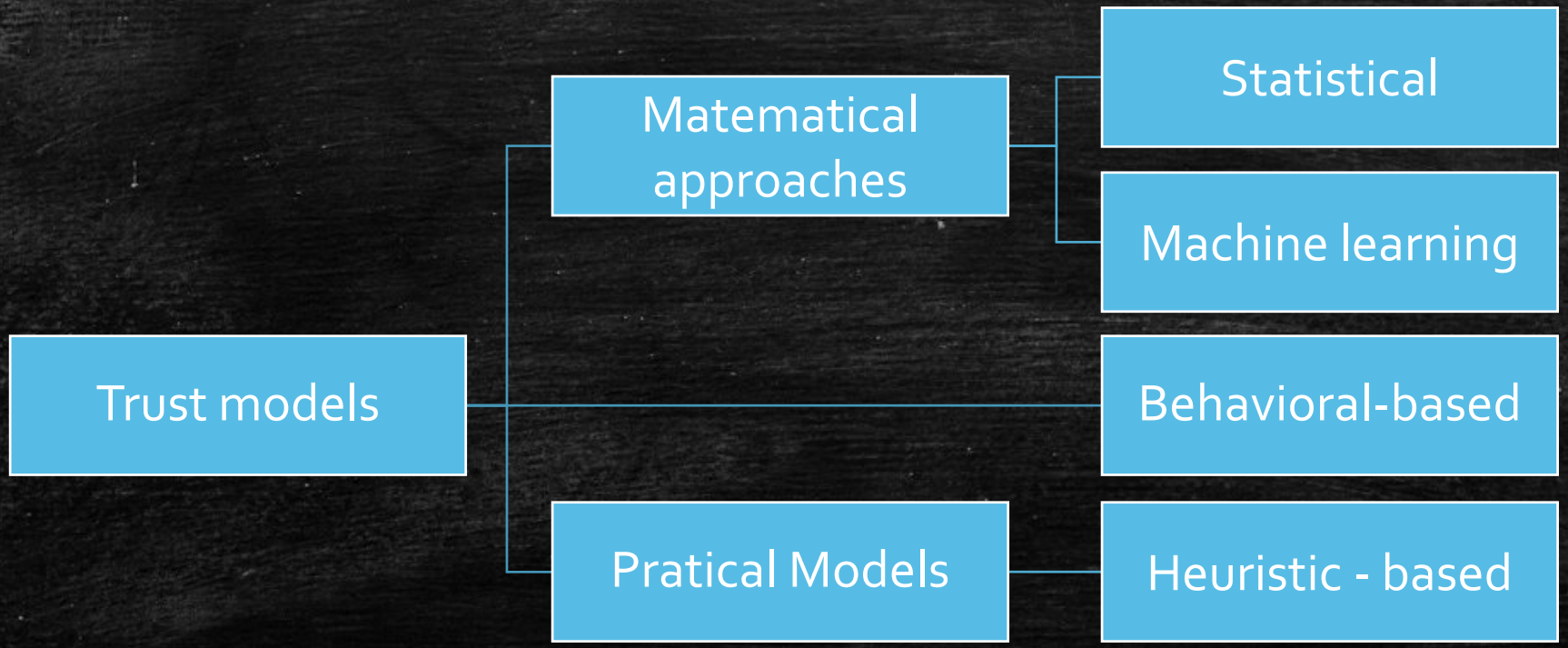
Trustee Properties

- objective: Safety and reliability of the trustee. In particular, the reputation is a public evaluation of the trustee obtained from his past behavior
- subjective: Honesty, benevolence and goodness

Everything is related to the context in which the relationship takes place, the Trust's objective, the Trust's environment in terms of, for example, time, location, activities, devices used, operating mode

Trust models

A rough list of the techniques adopted to evaluate Trust includes:



Trust and IoT

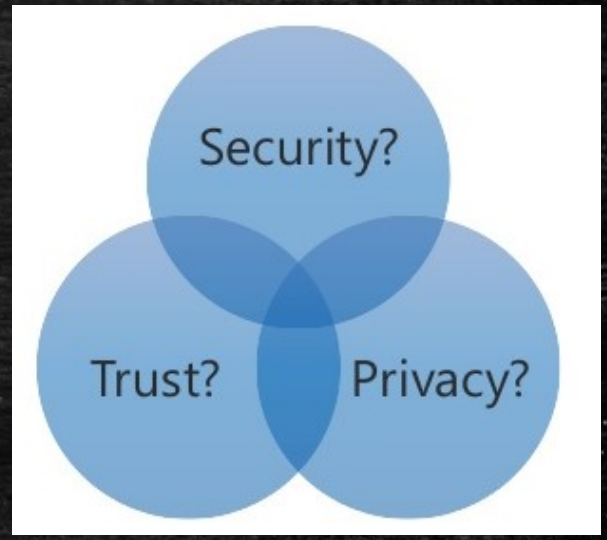
Trust and IoT

- In an IoT environment the concept of Trust is very complicated due to the presence of many properties (both measurable and non-measurable) and the fact that these exist in a mainly Machine-to-Machine context.

- Some typical features of the Trust in the IoT are:
  - Large (or even very large) populations of IoT devices
  - Relationships between persistent (even if intermittent) devices over time
  - Presence of nomadic devices (mobile between different federated environments that form a single community)
  - Interaction capability between IoT devices
  - Ability to evaluate the quality of the relationship according to its own targets (Human-to-Machine or Machine-to-Machine)
  - Computational and storage capacity (often limited for many IoT devices)

Trust and IoT

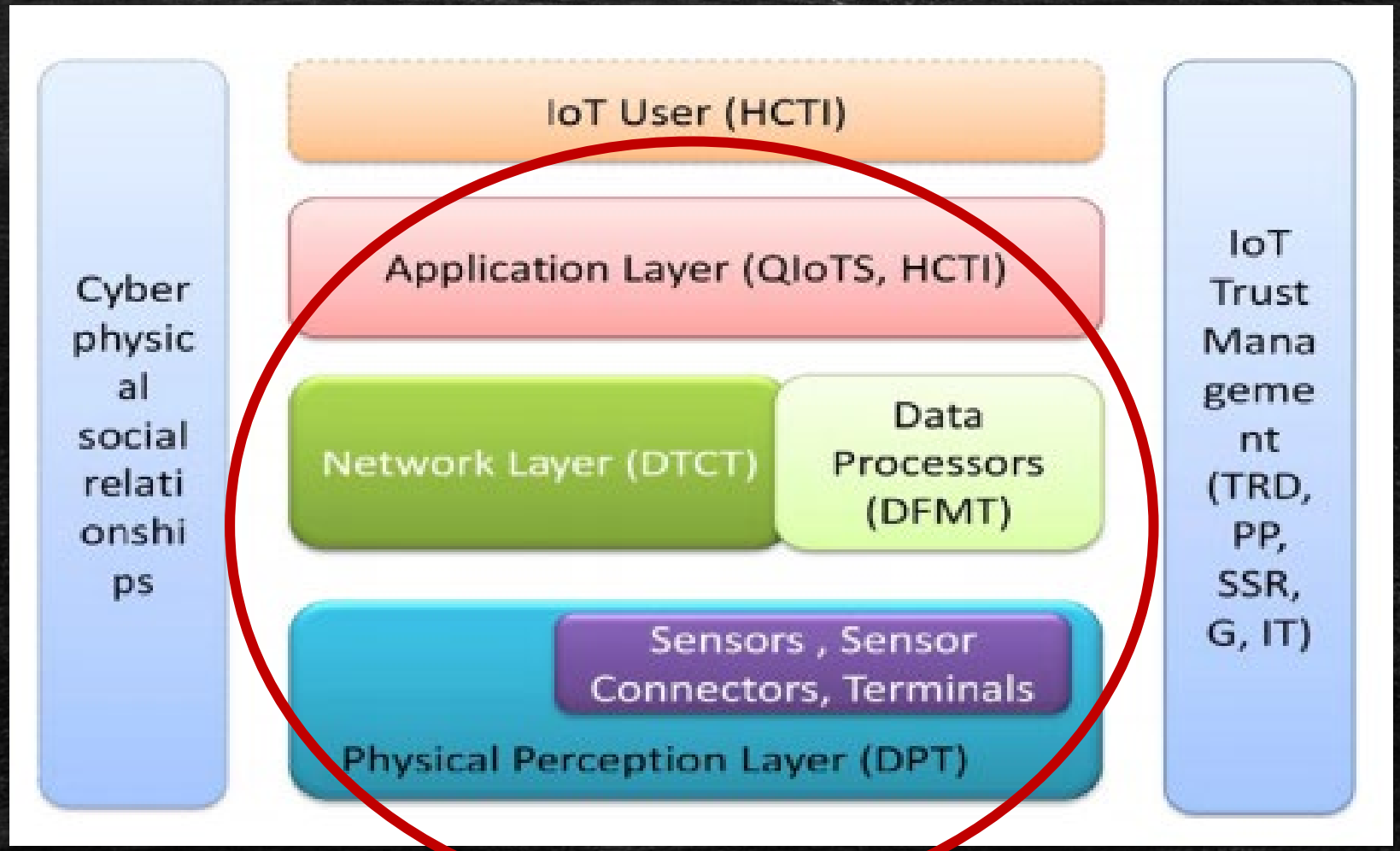
- As already said Trust and encryption are different and complementary
- By due to its nature, it is more complicated to manage the Trust than cryptography, and even more so when referring to an IoT environment.
- One aspect already highlighted concerns not only the concept of reliability in services, but also to establish to whom and when to communicate information



Trust and IoT

- A reliable digital system should preserve the privacy of users, which is one of the ways to gain user confidence
- Trust, security and privacy are crucial, highly related issues for IoT
- In general, the management of the Trust in the IoT covers part or all of the above Trust properties in different contexts for different purposes
- Below a IoT model will illustrate Trust properties should be implemented to achieve holistic Trust management

A model for the IoT



Trust and IoT

- Physical Perception Layer - (DPT): it can contain a large number of sensors, actuators, mobile terminals, connections and application of sensing technologies to perceive objects and humans in social environments by collecting large amounts of data.

- Network Layer - (DTCT): includes all components related to the network (regardless of the type of network, e.g. wireless sensor networks, ad hoc networks, cellular mobile networks and the Internet) is responsible for encoding data, joining them, discovering information (mining) analyze the data, all in order to provide the information essential to the operation of the application layer

- Application Layer - (QIoTS) and (HCTI): offers the desired services or applications to IoT end users in a pervasive and intelligent way

Trust and IoT

- In the IoT environment (as in other virtual environments) there are various cyber-physical relationships that variously involve the three layers introduced and that must be explored in search of useful information to offer advanced services to end users

- The management of the IoT Trust can be described in:
  - *gathering all the information necessary to make a trust decision; evaluating the criteria for analyzing trust relationships, monitoring and reviewing existing ones; ensuring the possibility of changing these relationships dynamically by automatically updating all the processes that are linked in the IoT System*

Goal 1

- To ensure an atmosphere of trust for the IoT in relation to the model presented above, the management of the Trust in the IoT should pursue the following 10 objectives:

- **Trust Relationship and Decision (TRD):**
  - Trust management provides an effective way to assess the relationships between IoT entities and assists them in making good decisions to communicate and collaborate with other IoT entities.
  - Trust principles must involve the whole system of IoT devices and all layers of the model (Vertical Trust) as they play a fundamental role for an intelligent and autonomous management of relationships based on the Trust.

Goals 2 and 3

Data Perception Trust (DPT):

- The acquisition and collection of data must ensure their reliability (trustworthiness)
- With reference to this aspect, attention must be paid to the Trust's properties such as sensor sensitivity, accuracy, security, reliability, persistence, as well as efficient data collection.
- All of these objectives subject to the Trust are referred to the IoT Physical Perception Layer

Privacy Preservation (PP):

- The user's privacy includes his data and personal information, however, should be flexible enough to the context without compromising the rules and expectations of IoT users.
- These objectives refer to general properties of the IoT System

Goal 4

Data Fusion and Mining Trust (DFMT):

- The large amount of data available in IoT environments should be analyzed and processed reliably, accurately, and in a privacy-friendly manner.
- This objective also refers to Social Trust Computing in relation to discovering the needs of users based on their social behaviors and social relations
- DFMT is about objective properties of the data analysis in the IoT Network Layer

Goal 5

Data Transmission and Communication Trust (DTCT):

- The data should be transmitted securely within the IoT System.
- Unauthorized entities must not access the user's private data during the transmission of this data
- This objective refers to the security and privacy properties of the IoT system.
- Trusted routing and, in general, management of all aspects of communication security in IoT networks are important and necessary instances to achieve this goal.

Goal 6

Quality of IoT Services (QIoTS):

- The quality of IoT services should be assured
- The paradigm to be adopted with reference to a user's expectations for a service should be "Only here, only me and only now" (Chen, 2012), all this implies that services should be personalized and offered in exactly the right place, at the right time and to the right person.
- This goal is mainly related to Trust Management in the IoT Application Layer, but in fact it must be supported by the other layers as well.
- The goal of QIoTS Trust Management is not only related to the trustee but also to the trustor, as well as to the context.

Goals 7 and 8

System Security and Robustness (SSR):

- Trust management in the IoT should effectively counter attacks to gain the trust of IoT users
- This goal is related to all layers of the model, focusing on system security (including reliability and availability, which are properties of the trustee).

Generality (G):

When the Trust system needs to serve multiple IoT systems and embrace multiple services, then it is preferable that it is generic enough to embrace all the requirements

Goals 9 and 10

Human–Computer Trust Interaction (HCTI):

- Trust management does not have to be complex and support human-computer interactions in a reliable way to be easily accepted.
- This requirement mainly refers to the trustor (i.e., IoT user) at the Application Layer level.

Identity Trust (IT):

- The identifiers within the IoT System must be well managed to strengthen the reliability of the System.
- In particular, this aspect must enjoy the properties of Scalability and Efficiency
- This objective involves all levels of the model in terms of the objective properties of the IoT System (e.g., identity privacy), the IoT entity (e.g., user expectations) and the context that may influence Identity Management policies.

Vertical Trust (Trust Manegement)

The **Vertical Trust** serves to achieve the global reliability of an IoT System with respect to all levels of the IoT model and not just level by level by strengthening security, privacy and Trust.

The **Vertical Trust** involves the management of:

- Trust Relationship and Decision (TRD)
- Privacy Preservation (PP)
- System Security and Robustness (SSR)
- Generality (G)
- Identity Trust (IT)

Trust management techniques that operate at each level (and may adopt different techniques) must be harmonized

An inclusive and holistic management of the IoT Trust requires that all of the above requirements can and should be achieved at the same time.

Trust Taxonomy

Some proposals of literature can be classified in:

1. Trust Evaluation
2. Trust Framework
3. Data Perception Trust
4. Identity Trust and Privacy Preservation
5. Transmission and Communication Trust
6. Secure Multi-Party Computation
7. User Trust
8. IoT Application Trust

Trust Techinques for IoT

The approaches presented will be compared in terms of versatility with respect to the objectives listed above

With reference to the taxonomy previously presented will be exposed the strengths and weaknesses of each individual proposal

Note that a number of the proposals presented below, which are classified in the literature as Trust-based, make significant (or exclusive) use of cryptographic techniques, thereby perpetuating the semantic ambiguity about the use of the term Trust

Trust Techinques for IoT

	TRD	DPT	PP	DFMT	DTCT	QIoTS	SSR	G	HCTI	IT
Bao and Chen (2012), Bao et al. (2013)	x					x (no context considered)	x (considered)			
Chen et al. (2011)	x					x				
Nitti et al. (2012)	x									
Liu et al. (2010)						x				
Ning et al. (2013)		x	x		x	x				x
Suo et al. (2012)			x		x		x	x		x
Zhou et al. (2012)		x			x					
Xiong et al. (2011)		x			x			x		x
Gessner et al. (2012)	x		x		x	x partially				x
de Leusse et al. (2009), Alam et al. (2011)					x support					
Yan et al. (2010), Jun et al. (2011)			x							
Roman et al. (2013)			x partially		x partially		x partially			x (some aspects)
Kothmayr et al. (2013)			x partially		x		x			x
Javed and Wolf (2012)		x								
Ukil et al. (2011)		x			x partially					
Khoo (2011)		x	x		x					
Feng and Fu (2010)		x	x		x					x
Sicari et al. (2013)	x	x		x (fusion only)			x partially			
Huang et al. (2013)		x								
Fongen (2012)										x
Hu et al. (2011)			x location privacy							x
Jara et al. (2011)										x
Sarma and Girão (2009)			x identity privacy							x
Zhang and Tian (2010)			x role privacy							x
Isa et al. (2012)					x					
Granjal et al. (2012)					x					
Raza et al. (2013)			x somehow		x					
Evans and Eyers (2012)			x	x considered						
(Huang et al. (2012), Gusmeroli et al. (2013)			x							
SMC			x (partially)	x						
Mishra and Chandwani (2008) in SMC				x			x (zero hacking)			
Pass (2004), Yao et al. (2008), Tang et al. (2011) in SMC			x (partially)	x			x			
Køien (2011)									x	
Li and Wang (2012)			x			x integrity support				
Petroulakis et al. (2013)			x		x					x
Thoma et al. (2012)			x	x						
Portelo et al. (2012)				x						

Trust Evaluation

Bao and Chen (Bao and Chen, 2012) proposed a Trust protocol that adopts social trust mechanisms and metrics measuring the Trust's QoS, using both direct (i.e. reliability) and indirect (i.e. reputation) observations to update a Trust score.

In practice, this proposal evaluates the Trust of IoT nodes ("things") based on three properties:

- Honesty - expresses whether a node is honest or not
- Cooperativity - expresses whether the Trust is socially cooperative with the trustor or not
- Common interest - consider whether the trustor and the trustee belong to the same communities/social groups (e.g. co-location or co-work relationships) or have similar capabilities

Trust Evaluation

Bao and Chen's proposal was one of the first to consider social relationships in the management of the IoT Trust.

The same authors then proposed further studies on the scalability, adaptability and resilience of this Trust protocol in dynamic IoT environments (i.e., subject to change).

The properties that this proposal satisfies are:

- Trust Relationship and Decision (TRD)
- Quality of IoT services (QIoTS) - non context
- System Security and Robustness (SSR)

Trust Evaluation: Discussion

Some objective and subjective properties of the trustee (e.g., Quality of IoT services - QIoTS) in relation to Trust Evaluation and Decision (TRD) processes are taken into account, always with reference to IoT contexts.

However, TRD processes based on social computing have not been studied yet

The TRD has not yet been applied to obtain QIoTS in relation to services like "*only here, only me and only now*", but only in relation to System Security and Robustness (SSR).

Finally, no proposal supports QIoTS, TRD, G and SSR instances simultaneously.

Trust Framework

Suo et al. (2012) have briefly analyzed the state of the art in the field of IoT with particular attention to security (cryptographic) aspects.

Through careful analysis of the architectures and functionality required for security at each level of the IoT have been added, for example:

- Phisycal Perception layer: lightweight encryption algorithms and protocols, identity authentication sensor data integrity and authenticity verification, anti-DoS systems
- Network layer: communication encryption and authentication mechanisms
- Data Processing layer: secure multi-party computing, cloud computing and anti-virus protocols
- Application layer: authentication, key management, privacy protection, security and management education

Trust Framework

The following key technologies are identified as key technologies: encryption; communications security; sensor data protection and encryption algorithms; while key management, data security laws and regulations and security management are only introduced

This work does not consider privacy protection in the layers of Physical Perception and Network (e.g., DPT), TRD, DFMT, QIoTS, and HCTI.

Trust Framework: Discussion

No proposal for secure IoT facilities is able to achieve holistic Trust management behavior with respect to all of the above objectives

All proposals focus only on some aspects of the Trust in the IoT, such as security and privacy of communications and data transmission (some of them support Trust Identity instances) mainly based on cryptography

However, many neglect targets such as TRD, DFMT, QIoTS, SSR, G and HCTI (DFMT and HCTI are systematically ignored by almost all proposals).

In practice, it still lacks a proposal for a Trust Framework including all the above mentioned objectives

Data Perception Trust

This approach is related to the "IoT Data Trust" for the collection and preprocessing of data at the Physical Perception Layer level.

Javed and Wolf (Javed and Wolf, 2012) studied how to verify the information of multiple sensors managed by different entities using tools external to devices

A technique has been developed to automatically model a physical phenomenon measured by IoT sensors.

This model can compare sensor readings through temporal and spatial interpolation and used in the context of any phenomenon having a continuous nature

This approach only meets Data Perception Trust (DPT) requirements.

Data Perception Trust: Discussion

Various types of connections can be conducted to compromise the DPT

Currently, research is oriented to solve specific problems in specific contexts without worrying about coordinating these actions with the various Trust mechanisms that could act in the different levels of the IoT model of reference, always to achieve other Trust related objectives

Some jobs also partially support TRD, DFMT, SSR, G, IT

The objective of future research should be the search for "light" solutions suitable to be adopted by IoT devices with limited computational and information processing capabilities.

Identity Trust e Privacy Preservation

Safeguarding privacy is an important aspect for the Trust in IoT

Evans and Eyers (2012) apply techniques to control the flow of information and "tagging" data in IoT while maintaining their privacy

This allows control over access to data based on the adoption of privacy policies and, in addition, this study considered the problem that sensors with limited resources might have in tagging data.

This work is significant for Data Trust and Privacy Preservation, but its complexity means that its easy adoption at the Physical Perception Layer level cannot be taken for granted.

Identity Trust e Privacy Preservation: Discussion

Most of the research is aimed at achieving IT and/or PP-related objectives, and some of them also support DTCT

However, no one considers DFMT, DPT, HCTI and QIoTS

A model that enables interoperability or integration with other Trust management mechanisms and that can be operational in several contexts has not yet been proposed in the literature.

Therefore, the current mechanisms for the protection of privacy and identification are imperfect and partial.

Transmission and Communication

Ensuring trusted data transmission and communication is crucial to achieving a Trust-based IoT

Progress in networking and communication can be applied to achieve DTCT and new (cryptographic) protocols will improve performance in this area and address new challenges.

A protocol to transfer data between IoT devices has been proposed by Isa et al. (2012), together with a *security framework* to improve the security and privacy of communication infrastructures.

A symmetric encryption protocol for data and asymmetric encryption for key exchange has been proposed in the Trivial File Transfer Protocol (TFTP) which is recommended for IoT applications.

Transmission and Communication: Discussion

The DTCT plays the role of the backbone to achieve the Trust in the IoT

Research in this area is focused on proprietary protocols or security schemes for specific IoT scenarios based on well-defined requirements and constraints.

In all this, the integration and interoperability of these protocols with other Trust management systems is simply ignored, making vertical management of the Trust impossible.

The objective of future studies should be to support vertical management of the Trust.

Secure Multi-party and Discussion

The *Secure Multi-party Computation* (SMC) aims to solve problems related to the execution of processing between participants that are not trusted with all the others, with particular regard to privacy aspects

The SMC refers to those who participate in the processing with their own confidential inputs to "*cooperatively calculate a function*" and receive, when processing is finished, its own correct output without knowing that of others

In this field there are about one hundred scientific contributions referring to different IoT contexts that support DFMT (*Data Fusion and Mining Trust*), some works also partially cover instances of PP (*Privacy Preservation*) and SSR (*System Security and Robustness*).

User Trust

Køien (2011) has carefully analyzed the Trust in the IoT by presenting its multi-faceted vision of software, hardware, devices and services

He considered aspects of transitivity, reflexivity, propensity and risk assessment, distrust, deception, retaliation, altruism, reputation, associationism and brands, human mind

Køien points out that no one can blindly trust any of the IoT components (*e.g., software, hardware, communications, etc.*), but that does not mean that IoT services are not trustworthy.

Human heuristics allows you to manage risks, threats and opportunities even if it is subject to failure but the use of *trusted proxy devices* and trust in brands and/or companies allows you to have an acceptable level of Trust

User Trust: Discussion

In the literature a limited number of papers have analyzed aspects related to *Human-Computer Trust Interaction* (HCTI) in the IoT and studied the user's behavior with respect to the Trust.

Obviously, User Trust and HCTI are decisive for the acceptance and success of IoT services and their applications.

Trust and IoT

There is a sufficient number of IoT applications in many areas of our lives where the Trust is relevant even though PP, DFMT, DTCT and IT are usually partially considered in relation to the Trust's objectives specified above.

Obviously, the IoT applications do not yet consider all aspects related to the Trust instances and all the objectives related to the management of the Trust in the IoT and in particular not only based on cryptographic techniques

It is important that in the future all aspects related to the Trust operate in a holistic (i.e., integrated and rational) way.

Discussion

There are a number of unresolved situations in relation to the Trust

- As a first point, there is a general lack of awareness in assessing the Trust in relation to the context and properties of the trustor.

- Trust evaluation is poor compared to the level of customization and so often it becomes complex to provide customized IoT services according to the paradigm "Only here, only now and only for me". (which represents a goal still far away)

Discussion

Research is inadequate in proposing integrated Trust management systems that support all Trust objectives in the IoT.

Existing proposals focus especially on security and privacy to support DPT, PP, DTCT, QIoTS and IT instances; however, TRD, DFMT, SSR, G and HCTI are rarely considered when designing the *Trust Framework*.

Current progress in the development of DFMT has not yet been implemented in practice or used in real products, services or systems.

Instances of SSR, Generality and HCTI are generally artfully ignored, but aspects of Trust management are essential. Obviously, Trust management covers more objectives than just security management, even if some people think that the simple IoT management of security and privacy is the panacea for everything.

Discussion

Although *Trusted Computing* platforms based on DPT solutions have been proposed, they may be too heavy or complicated due to the limited capabilities of the wireless sensors that can be adopted.

Basic security and trust mechanisms adapted to non-sophisticated devices is still to be developed in the IoT area, in particular to protect against DoS attacks.

The cooperation of the TPD with other *Trust Management* mechanisms should be designed to support the vertical management of the Trust.

Discussion

Privacy protection in the IoT is not adequate

Only some research work proposes complete PP solutions that operate on different levels of the model

Real cross-layer solutions should be developed to integrate and harmonize PP mechanisms in different IoT levels with particular reference to identity, location, time, behavior, business processes and other types of information.

Since few studies involve DFMT and DPT and none HCTI and QIoTS, one objective of future research should be the interoperability or integration of PP techniques with the vertical management mechanisms of the Trust in different contexts.

Discussion

The interoperability or integration of *Data Transmission and Communication Trust* (DTCT) technologies with other Trust Management mechanisms has not yet been studied and should be a topic to be developed in the future for a vertical Trust in different contexts.

A strong *Secure Multi-party Computation* (SMC) that supports IoT's vertical Trust goals is still an open challenge

HCTI is almost ignored by current research, but instead it is one of the decisive aspects for the success of IoT. A holistic IoT Trust management with a good *User Experience* determines the ultimate success of the IoT.

Next Challenges

In environments denoted by the heterogeneity of IoT devices, Trust instances are most important

Calculating and transferring trusts between different networks is a difficult problem

The management of the Trust on different networks adopts the same objective/subjective criteria referring to trustors/trustees already exposed

In the transfer of the Trust between different *Wireless Sensor Networks* the use of Internet could help to reduce the computational workload and overload of the network

Next Challenges

The efficient use of energy requires the development of more efficient (and faster) algorithms and Trust mechanisms that allow to consume less energy (this would generally require avoiding the use of cryptographic schemes).

This aspect is very important in the presence of small devices and it has not yet been deepened by the research

Next Challenges

Performance improvement (in general) remains a major challenge in the area of Secure Multi-party Computation (SMC) and homomorphic encryption (homomorphic encryption that allows the manipulation of encrypted data).

In this context, efficient key management and distribution is important from a privacy point of view

On the other hand, the introduction of complex algorithms and cryptographic techniques increases energy consumption due to the need to perform more complex calculations.

Next Challenges

An autonomous Trust management system is difficult to implement at device level, especially in the presence of poorly equipped devices.

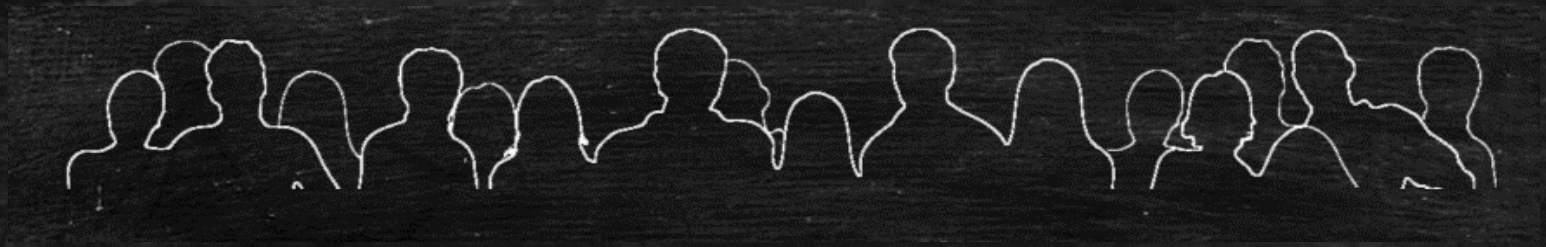
Similarly if the IoT is supported by the Cloud because the devices in this case are mobile and it is often complex to take this fact into account and, of course, even more so if the devices have limited resources (e.g. computational)

A reliable **Data Fusion** is not easy to achieve. Transmitting raw device data is too expensive so **Data Fusion** techniques are essential to reduce these costs.

However, obtaining reliable, efficient, accurate, and confidential data fusion and mining processes is not easy.

Next Challenges

As already mentioned, the integration and cooperation at all levels of all Trust management mechanisms to realize a holistic vision of the Trust in the IoT is a great work that has not yet been done and which represents the biggest challenge



Social Trust

Introduction to Social Networks

- **Social Networks** (SN) are social structure connecting social actors among which occur some social relationships
  - The social actors can be: Individuals, Groups, Organizations, Governments, …
  - The nature of the social relationships can be: Interactions, Trades, Values, …
- Given the relevance of SNs, they are widely investigated.
- The SN study is known as "Social Network Analysis", which provides methods for analyzing SNs in order to identify local and global patterns, locate influential entities, and examine the SN dynamics.

Introduction to Social Networks

- The Social Network Analysis has an intrinsically interdisciplinary nature involving several scientific sciences, among which psychology, sociology, statistics, graph theory, economy, computer science and so on.

- Internet modified traditional SN on three main dimensions:
  - Space: social interactions are not more connected to physical places
  - Time: social interactions can be carried out in a "time independent" fashion
  - Size: the potential individual social sphere is incomparably greater

- SNs make easier to have an overwhelming number of social interactions with (almost) everyone in everyplace and in every time than traditional SNs.

Social Trust

- The Social Trust is closely linked to the Social Relations and therefore Social Network (SN) concept to which, for convenience, we will refer to
- SNs have been defined by J.A. Barnes (1954) as connected graphs where the nodes represent the social actors and the arcs identify the social interactions between the nodes and this kind of representation is suitable to underline the intrinsic complexity of SNs.
- In other words, in the IoT we can consider that each node represents an IoT device and the outgoing/entering arcs its relations with the other nodes
- Several studies have been devoted to SNs and their characteristic properties and how external factors can influence them.
- In addition, the features of the devices characterize these SNs in terms of Space, Time and Size

Introduction to Social Networks

- In the early utopian view, Internet allows an enhancement in existing social connections and encourages new connections. Conversely, in the dystopian view, Internet alienates people in a virtual world were social connections are not real [Wellman, 2004].
- These sentences are paradoxes because Internet has only changed modes (i.e., the medium), given new opportunities to carry out social communications and provided traditional SNs of the basis to realize specialized and multiplexed OSNs
- Internet does not negatively affected social communications but strong traditional connections usually remained strong also into OSNs, weak existing connections are often become strong, as well as virtual connections sometime are become real connections.

Property: Density/Spread

- One can distinguish social relationships in:
  - Dense where the members of the network are closely connected to each other and have frequent mutual relationships
  - Scattered where network members are poorly connected to each other and only a few of them have frequent mutual relationships

- A further main characteristic is that the network of social relations in the IoT requires both the sharing of objectives and that at least some characteristics of the nodes are shared (similarity).

Property: Similarity

- In the IoT domain similarity can be understood as the tendency of nodes to associate or relate to other similar nodes for one or more aspects

- Two types of similarity can be distinguished (the terms reflect those used in SN):
  - By state: in this case we refer to the nature of the node (for example, homogeneous nodes by nature)
  - By value: in this case we refer to the end of the relationship (for example homogeneous nodes per objective)

Property: Small World

- A "small-world" network is a mathematical graph theory that states that in complex networks most nodes are not close to any other node of the network, but from each node can be reached with a sufficiently limited number (depending on the size of the network) of "hops" every other node of the network

- A small-world network is defined as a network where the typical distance **L**, that is the number of connections (steps or hops) between two randomly chosen nodes, grows proportionally to the logarithm of the number of nodes **N** that belong to the network

- *Note: This theory suggests that human society is a small-world characterized by relatively short paths between its components*.

Property: FOAF

- The concept of "Friend of a Friend" (FOAF) is common in many social networks [Mika,2007] and implies that "Friendship" is a transitive property, but remember that the Trust is not transitive but propagative.
- A FOAF approach can be risky when the size of the network grows and can lead to a loss of Trust in the system.
- In real social networks, Granovetter [1973] introduced the concept of "tie strength" (with reference to time, emotions, intensity, intimacy and reciprocity) that characterizes the surroundings of people with whom an individual has relationships (note that similarities and trusts are also keys to forming stable groups [Rosaci et al., 2016]).
- In IoT these concepts must be mediated and in practice they usually have a more limited dimensional scale.

Property: Social Capital

- A shared definition of Social Capital (SC), from the point of view of SNs, can be understood as the collective value associated with an SN

- How is the SC measured? The measurement cannot be unambiguous, but it depends on the purpose of the network and different can be measured, which ones:
  - Structural — characteristics of the social interactions between the actors
  - Relational — frequency of relationships between actors
  - Cognitive — knowledge sharing
  - Altruism — benevolence towards other actors
  - Speculative — activation of connections between the actors that increase the value of SC
  - Collective — cooperation at community/group level
  - Attitudinal — presence of factors that influence the attitude to relationships between the actors

Property: Social Capital

- Undoubtedly, the basic elements of *Social Capital* are:
  - The number of interactions
  - The nature of interactions

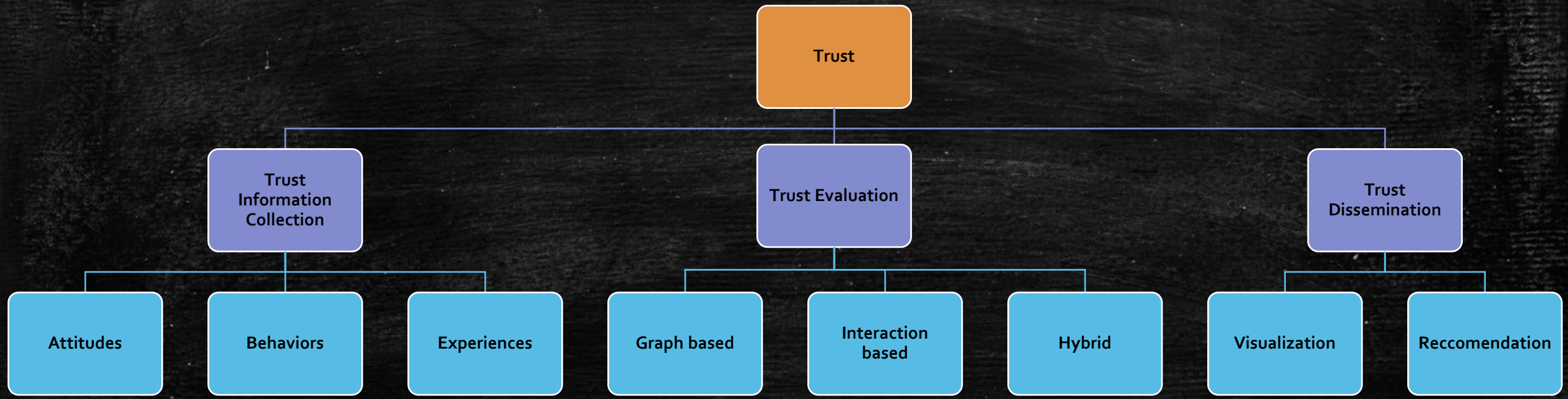
- As a result, high *Social Capital* requires a high number of interactions that provide members of a community/group with (possibly significant) benefits.

Social Capital vs Social Trust

- The *Social Trust* is the skeleton of *Social Capital* and this is an important detail
- High social capital requires good behaviors that are only possible in the presence of Trust among the members [Putnam, 2000] and *Social Capital* increases when the Trust increases [Huang, 2007].
- Some attitudes (i.e., to cooperate, etc.) help the growth of the *Social Trust* [Huang, 2007].
- Some studies have shown that the *Social Trust* depends primarily on the entities involved, while the social structure and social norms are less decisive for its growth.

From Social Capital to Social Trust

- Calculating the Social Trust can be seen as a three-step process:
  - Collection of information
  - Evaluation
  - Diffusion.



Trust Information Collection

- Trust information can have three main origins:
  - Attitude
  - Behaviors
  - Experiences

Trust Information Collection: Attitude

- The Attitude should be understood as the set of natural attitudes in terms of "*positive*" or "*negative*" with respect to something, but in IoT can refer to only two out of the three models classified in the literature
  - ~~Affect: emotional response in terms of preference for a target~~
  - Behavior: typical trustee behavior
  - Cognition: evaluation of the trustee with respect to the objective
- Most of the '*Attitudes*" are influenced by the environment and experience and, therefore, are subject to changes over time.

Trust Information Collection: Experiences

- *Experience* is the perception that you have on the basis of implicit (indirect) or explicit (direct) knowledge gained from past experiences with the target
  - Explicit: experience derived from first-hand interactions
  - Implicit: experience derived from third party feedback
- *Experience* can influence *Attitude* and *Behavior*
- Positive experiences encourage future interactions and vice versa for negative experiences
- A large number of Trust models are based on experiential information (explicit, implicit or both) and all reputation models are based on *Implicit Experience*

Trust Information Collection: Behavior

- The observed *Behavior* of the devices reflects much more their nature than the knowledge from past direct experience.
- For example, rapid "behavioral" changes can be detected and their effects on the Trust properly considered
- The Behavioural aspect is generally considered separately from the other two (i.e., *Attitude* and *Experience*).
- However, the trend in Trust research is to integrate all three aspects into Trust models, which is a complex challenge, particularly in IoT because this would require the adoption of complex (usually resource-intensive) models.

Trust Models

- To represent the Trust in a synthetic way a numerical value is generally used which is calculated on the basis of some Trust model

- Within the Social Trust there are three main families of *Trust models*:
  - Network-based
  - Interaction-based
  - Hybrid

Trust Models: Network-based

- The structure of the network influences the degree of Trust
- The increase in connections in and out of a member, from a theoretical point of view, produces a potential improvement in their knowledge of the other members of the community/group and, therefore, increases the degree of awareness of their level of Trust towards their trustees.
- Buskens [1998] verified that:
  - Members with a high level of outgoing connections have a high level of Trust
  - The Trust level increases if outgoing connections are directed to members who have many outgoing connections
  - Members with a central role in the network increase their Trust level

Trust Models: Network-based

- Models that exploit the structure of the network adopt the concept of "*Web of Trust*" or *FOAF*
- A sub-network Trust (ego-network) generally exists for each member and its nodes represent members and arcs of trust relationships.
- An SN can be seen as the union of its members' ego-netoworks
- There are different approaches to cross a network and determine the Trust between two nodes
- These approaches take advantage of the propagative nature of the trust and calculate it based on how the members are related to each other and how the Trust flows through the network; however, these techniques fail to capture the intensity, frequency and nature of the interactions that are important indicators of Trust

Trust Models: Interaction-based

- These models consider only the interactions that happen between the members ignoring completely the structure of the network
- Usually these models provide to classify the type of interaction in order to weigh it opportunely on the base of its typology
- Example of a possible classification is:
  - the type of interaction
  - the nature of the entities involved

Trust Models: Hybrid

- Hybrid models simply combine the two previous models
- The development of the Hybrid models is much more complex than the other two considered individually, also due to the difficulty of keeping together in a unitary model, and in an appropriate way, information such as social ties, frequency and nature of interactions.
- Given their intrinsic complexity these models are not very widespread and, moreover, there is a tendency to simplify them in order to make them more manageable, but this makes them lose many of the advantages given by using a unitary model

Spreading: Recommendation

- *Trust-based Recommendation* models built from a Trust network, where nodes are entities and arcs (directional) represent Trust relationships, are able to generate very accurate and highly personalized suggestions by aggregating the opinions of the other members of the Trust network.

- Although these models are intended to disseminate Trust information on the network, it is verified that this information is not always accessible to everyone and at all times especially for mobile devices or operating periodically (asynchronism of information).

Spreading: Visualization

- The Visualization of the Trust is a way for its dissemination (and awareness) within a community
- Through a graph, the number of connections between two nodes (i.e. members of a community/group) represents the strength of their relationships and the level of their involvement.
- SocNetV, NetVis and Graphviz are freely usable tools to display Trust information on a graph
- The frequency of relationships can be combined with temporal information to represent the evolution of relationships over time (diachronic graphs).
- The main advantages of the visualization models is in the ease with which you can identify the most relevant nodes and relationships, on the other hand, with large graphs there is the risk of not immediately grasping relevant information

Further Comments

- In computer science the focus is on how to calculate Trust values, in sociology and psychology on how Trust influences decision-making processes. We need a more unified vision.

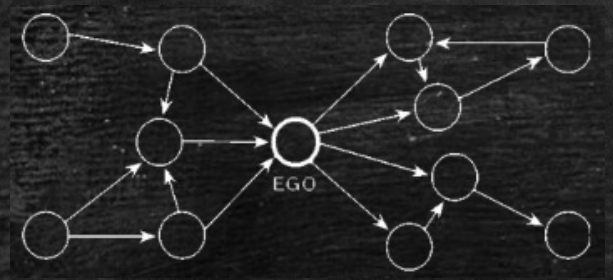
- Neglected aspects in the Social Trust are:
  - When the Trust can be transferred from one context to another
  - The initial Trust value to assign to a new member so as not to penalize them too much
  - The weight to assign to active interactions compared to passive interactions
  - The level of Trust needed for there to be a Trust community and not just a community
  - The impact of strong and weak links between members in relation to Trust models (the former identify the sphere of already established Trust relationships with other community members and provide highly reliable information; the latter are important sources of new information [Granovetter,1973], although the reliability of this information may be low).

Further Comments

- Malicious and aggressive behavior should encourage us to implement robust Trust models capable of countering different types of attacks aimed at altering the representation of the Trust within the community [Josang et al, 2007].

- Privacy management can hide a correct representation of the Trust, on the contrary the absence of Privacy increases the opportunities to perpetrate attacks and consequently exposes the Trust (individual and community) to risks.

- Centralized or distributed approaches to collecting, evaluating and disseminating information about the Trust have a strong impact on the *Social Trust* because they change the individual representation (e.g., overview or bird's eye) of the Trust within the community.



Example 1: An Agent-based Sensor Grid to Monitor Urban Traffic

Premise

- Vehicle traffic is constantly increasing
- Authorities prefer to manage existing rather than create new infrastructure, even when this is possible.
  - *Increased environmental awareness*
  - *Reduced availability of financial resources*

Premise

- Intelligent Transport Systems (ITS) systems help to improve the performance of the transport network and enable the infrastructure manager, for example:

  - *Assist users in the choice of routes using information captured in real time and sent to users' personal devices*
  - *Adopt effective strategies for traffic control starting also from the analysis of users' habits*

Proposal

- Monitoring of urban traffic through a grid of acoustic sensors (Neural Network based) capable of recognizing the passage of vehicles and classify them into three categories, based on the noise emitted and a heuristic trust-based algorithm to mitigate any inaccuracies
- The proposed system is able to work in real time, is economical and easily installed
- Measured performance is more than satisfactory
- Each sensor is associated with a software agent that coordinates its activities and collaborates with other sensor-agents.

Traffic sensors typology

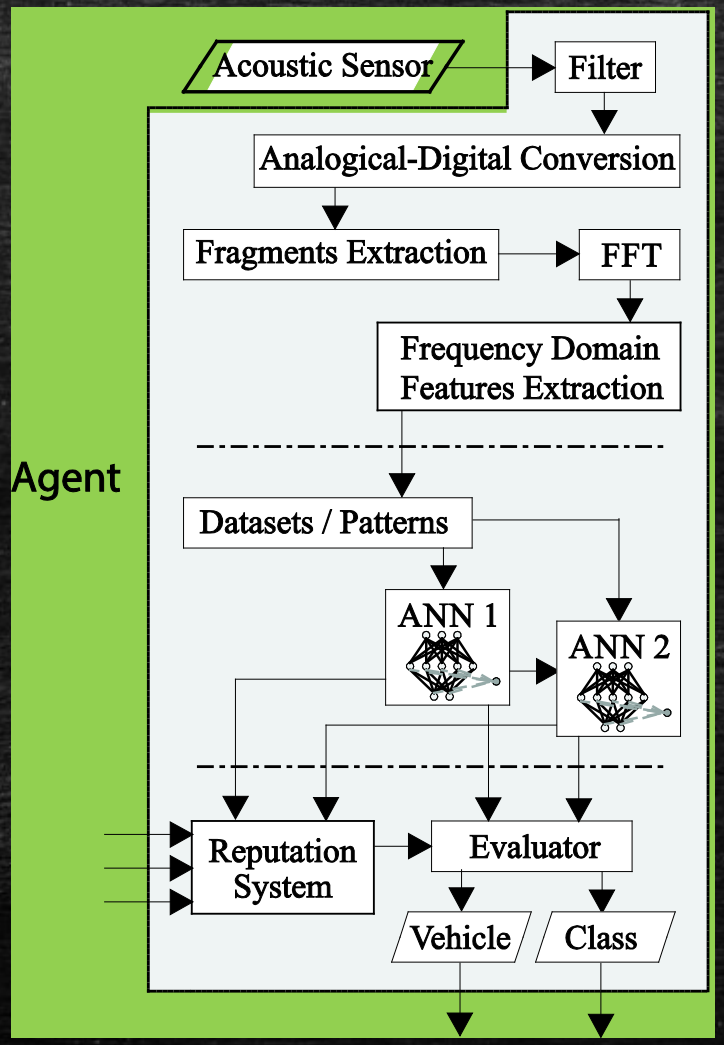
- Signal Detection
  - on board
    - GPS-based
- in/over roadway
  - intrusive
    - magnetic coils
    - piezometric plates
    - pneumatic tubes
- in/over roadway
  - not intrusive
    - video
    - audio
    - microwave
    - infrared

The Acoustic Signal

- The acoustic signal produced by a vehicle is generated by:
  - Pneumatic-road interaction
  - Internal sources (i.e. engine, various rotisms)

- Can return information about:
  - Speed
  - Distance between vehicles
  - Class of vehicles
  - Acoustic and non-acoustic pollution (by extrapolation)

The Sensor Agent



Criticality of the Acoustic Signal

- Very complex information extraction
- Doppler Effect
- Climatic conditions
- Rapid changes that occur in time and frequency domains
- Composition and entity of the traffic flow
- Stationary or very slow vehicles
- Obstacles that reflect the signal
- High background noise levels
- Incorrect sensor positioning

Signal Processing

- The acoustic signal of the vehicles is full of information (not all necessary). 90% of the useful part is in the range 0.1-5 KHz, therefore:
  - Filtering Phase: All low intensity signals and all information above 5KHz are eliminated from the analog signal.
  - A/D conversion by applying a Pulse Code Modulation (PCM) transformation with:
    - Sampling at 10 KHz
    - 16-bit Quantization
    - Grey encoding

Signal Processing

- From the signal between two gaps of 1 sec. is extracted an interval (F) of 1.5 sec. (centered on the peak value of the signal), in turn divided into 3 slices of 0.5 sec. (to exploit the Doppler effect) and converted in the frequency domain by a Fast Fourier Transformation (FFT).

- Each slice is then divided into 10 frequency bands, each one represented by the average value of the emitted power.

Signal Processing

- 2 NN multilayer (input + hidden + output) Back-Propagation in cascade with hyperbolic and sigmoid activation functions for the hidden layers and output, respectively
- The two NNs both receive 30 inputs
- The first NN discriminates the passage of a vehicle from a noise, the second classifies each recognized vehicle in three classes (car, heavy vehicle, motorcycle)

Signal Processing

- 3 classes of vehicles and 6 classes of noise (rain, wind, strong wind, various noises, white noise, background noise) were used for training.
- Each sensor-agent makes a correction of the output returned by its two neural networks using that of the agents directly connected to it.
- The correction is based on the trust values that each agent assigns to its neighbors and that are used to weigh the traffic values that they return

Trust Reliability Reputation Model

- A distributed version of TRR is adopted in this work.

- In TRR each agent has its own perception of trust (τ) than any other agent based on reliability (ρ) and reputation (π) measures.

Trust Reliability Reputation Model

- RELIABILITY
  - Each agent autonomously models his or her perception of reliability based on the iterations that occurred, therefore $\rho_{ab} = f(i_{ab})$
- REPUTATION
  - Agent a calculates b's reputation (i.e. π_{ab}) by asking other agents for their opinion. At TRR we assume that the opinion of b is the measure of trust an agent has of b. Agent a weighs the opinion of c using the measure of trust that a has of c (i.e. τ_{ab})

$$\pi_{ab} = \frac{\sum_{c \in C - \{a,b\}} \tau_{cb} \cdot \tau_{ac}}{\sum_{c \in C - \{a,b\}} \cdot \tau_{ac}}$$

Trust Reliability Reputation Model

- TRUST
  - The trust measure combines those of reliability and reputation

$$\tau_{ab} = \alpha_{ab} \cdot \rho_{ab} + (1 - \alpha_{ab}) \cdot \pi_{ab} \qquad \text{with} \quad \alpha_{ab} \in [0,1]$$

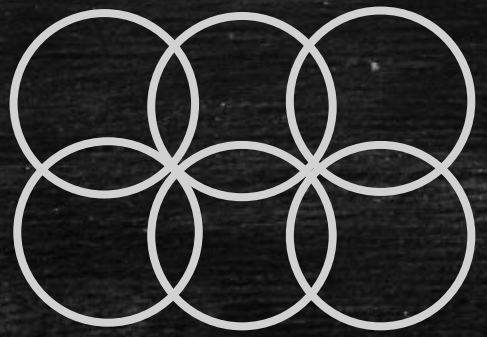
$$\alpha_{ab} = \begin{cases} \dfrac{i_{ab}}{N} & \text{if } i_{ab} < N \\ 1 & \text{otherwise} \end{cases}$$

  - All agents require n linear systems in n-1 equations

$$\tau_{ab} = \alpha_{ab} \cdot \rho_{ab} + (1 - \alpha_{ab}) \cdot \frac{\sum_{c \in C - \{a,b\}} \tau_{cb} \cdot \tau_{ac}}{\sum_{c \in C - \{a,b\}} \cdot \tau_{ac}}$$

Distributed Trust Reliability Reputation Model

- Distributed TRR
  - In this work, each agent uses TRR only with reference to the agents directly related to him in the road graph.

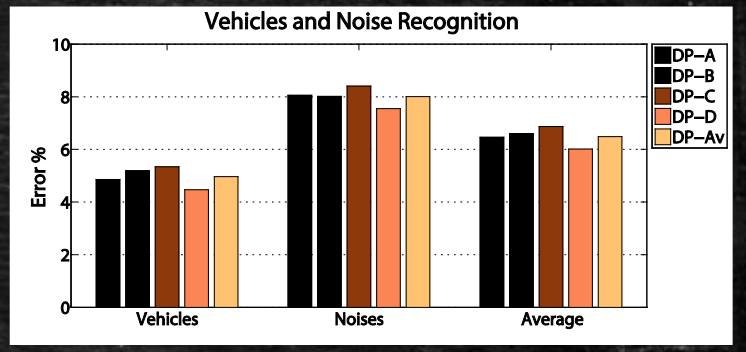


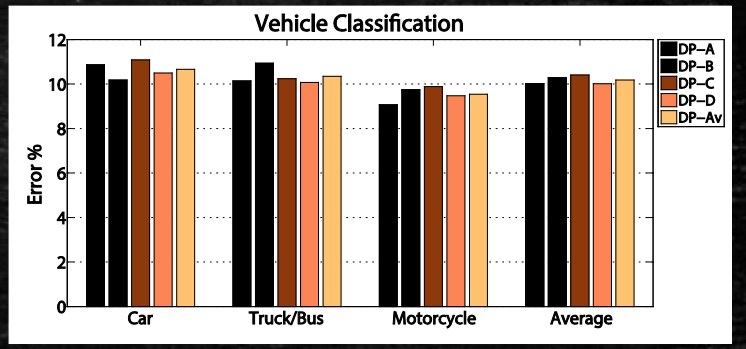
Experiment 1

- 4 detection points for 4 working days for 3 sessions per day (8-9, 13-14, 18-20) on one-way and one-way roads in different traffic and climate conditions
- Detection station: Beringer C1 microphone, stand, notebook, video camera
- Topology NN1: 30+55+1 neurons on three layers
- Topology NN2: 30+25+3 neurons on three layers
- Output NN1: vehicle/noise
- Output NN2: car/heavy vehicle/motorcycle
- Training datasets: 2500 normalized patterns (50% vehicles, 50% noise on 6 noise classes)
- *Note that adopting only one NN is not possible to have an acceptable accuracy*

Experiment 1

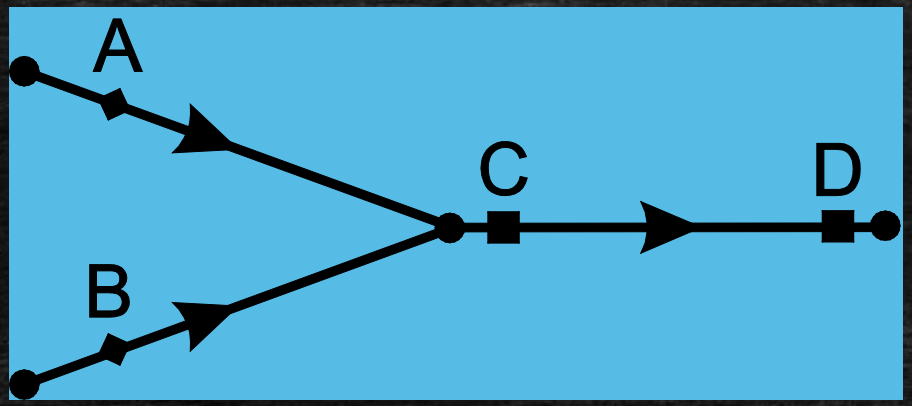
- To verify the identification capability, the NNs test pattern is composed of a continuous stream of acoustic signals
- Positive results: 93:41% for NN1 and 88:46% for NN2
- The results are interesting, the errors are mainly due to:
  - Acoustic signatures of vehicles similar to those of other categories
  - The errors of the first NN affect the performance of the 2nd NN.
  - Cars stopped or very slow and bad weather conditions
- In general, tests have shown a slight tendency to overestimate vehicles. Most of the errors are noises recognized as vehicles.





Experiment 2

- The 2nd experiment verifies the ability to operate in grid, mitigate anomalies in traffic measurements and the tendency to overestimate the sensors.
- The test was done on a small grid of 4 sensors



- Each sensor agent (i.e. x associated to the detecting point x) sends to its neighbors the traffic measurements (i.e. F_x) and the trust values of its neighbors that it has calculated with reference to the time interval Δt

Experiment 2

- Let $F'_x = F_x \cdot \tau_x^x$ the weighted value of the traffic measurement of x
- If F_x (i.e. F'_x) is greater than F_x^{max} then $F_x = F_x^{max}$ $(F'_x = F_x^{max})$
- Be FI_x and FO_x (i.e. FI'_x and FO'_x) the sum of the incoming and outgoing flows from x (i.e. incoming and outgoing flows weighted by the trust value that agent x has on another agent's ability to provide correct traffic measures FI'_x and FO'_x and both $\overline{F_x} = \frac{FI'_x + FO'_x}{2}$

Experiment 2

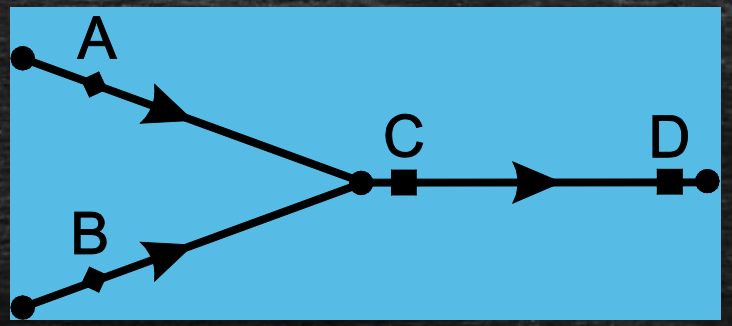
- No correction of traffic measurements is made if:
  1. F_x, FI'_x and FO'_x differ by more than 20% from the estimated values in the previous range
  2. F_x, $FI'_{x,i}$ and $FO'_{x,i}$ are respectively equal to FI_x^{max}, $FI_{x,i}^{max}$ and $FO_{x,i}^{max}$ otherwise

$$F_x'' = \begin{cases} F_x' - \delta \cdot \dfrac{\bar{F}_x - F_x'}{2} & \text{if } F_x'' \leq F_x^{max} \\ F_x^{max} & \text{otherwise} \end{cases}$$

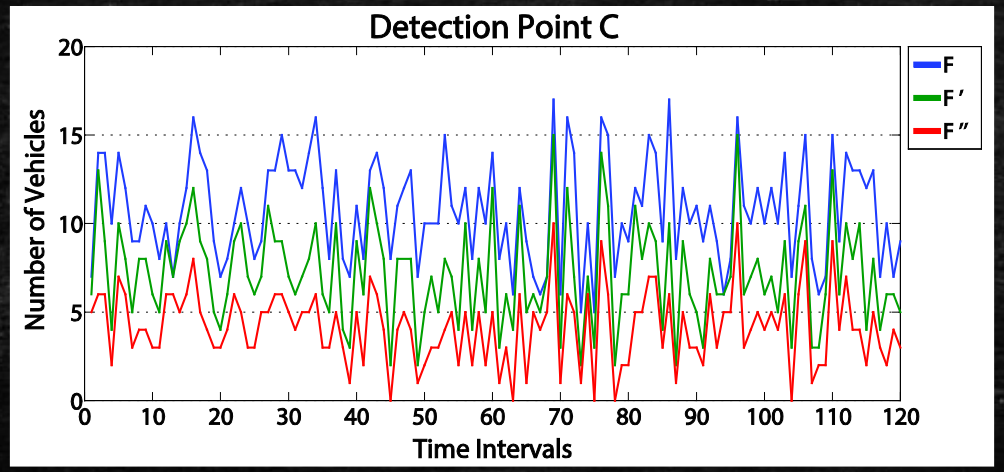
$$\rho^x = \begin{cases} \rho^x + \varphi \cdot \dfrac{F_x'' - F_x'}{F_x} & if\ \rho^x \leq 1 \\ 1 & otherwise \end{cases}$$

Experiment 2

- With reference to node C of the sensor grid

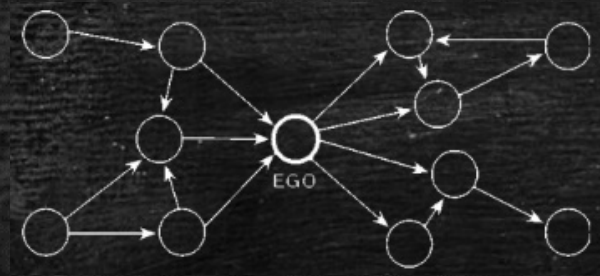


- and adopting Δt=2 minutes δ=0.5 φ=0.75



Conclusions

- It was presented an agent-based sensor that exploits the sound emitted by a vehicle to identify its passage and typology by exploiting the capabilities of artificial neural networks.
- These sensors can work in a grid and cooperate with their neighbors to refine their traffic measurements using a trust system and heuristic algorithm.
- Two experiments on real data were carried out considering the sensor both in stand-alone and grid configuration



Ego-Networks

Ego-networks

- An important issue in OSNs is the capability to generate useful suggestions for users, as users, resources and so on
- This implies the necessity of evaluating the trustworthiness a user should assign to other members of his/her online community
- A number of models that rely on "global" reputation: they are based on the evaluation of the behaviors of the users, that is spread across the entire community
- These models show an evident limitation due to the difficulty of taking into account the effects of malicious or fraudulent behaviors
- Other approaches, that consider also a local perspective of the trust, are limited by the fact they are supervised, i.e. they need a training phase in generating recommendations

Ego-networks

- A novel approach extends global reputation models by integrating a local reputation, computed on the ego-network of the user, by means of an unsupervised approach
- It is a sub-network of the entire OSN, containing all the information coming from direct experiences with other members and those coming from the direct experiences of these later
- A novel model has been designed to consider three main parameters, namely:
  - the relevance given to local reputation with respect to global reputation;
  - the threshold of reputation under which a user can be considered as unreliable;
  - the size of the ego-network associated with a target user, i.e., the subgraph of the social graph containing all the OSN members connected to the user via a path of trust links.

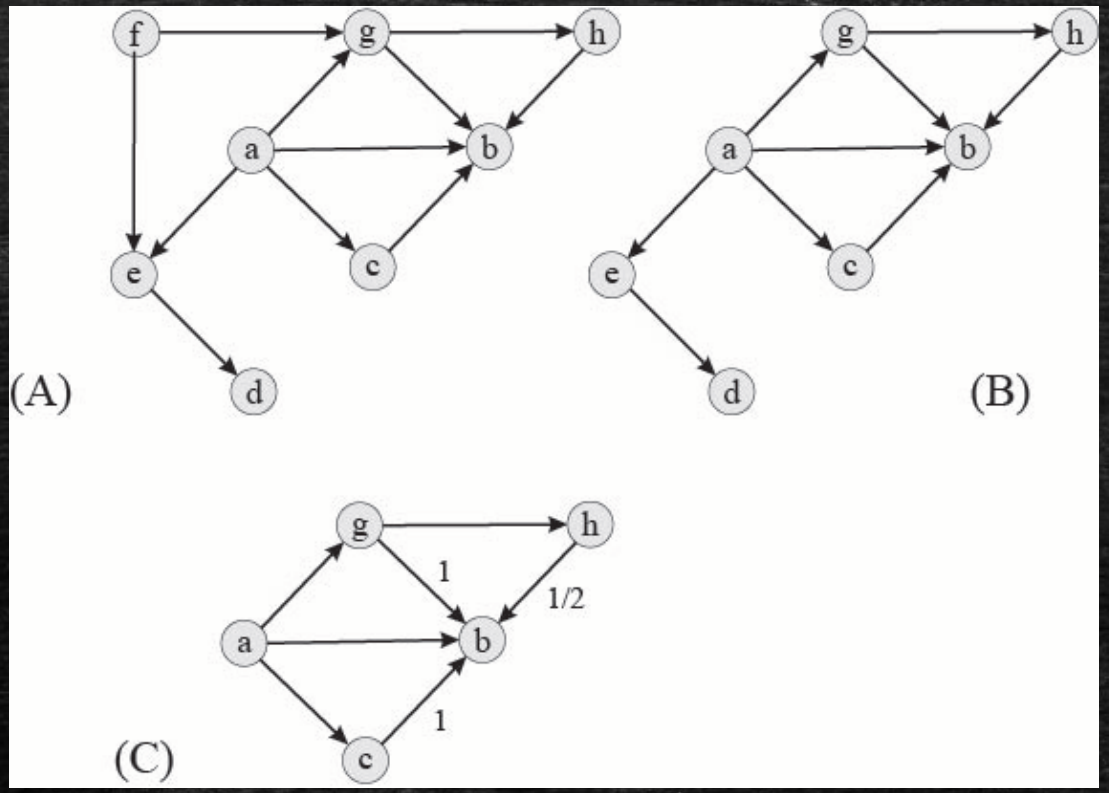
Ego-networks

- This proposal is aimed to recommend users as possible interlocutors without the need of a training phase, since the predictions are based on the data available at that time.

- Results referred to the different role played by local and global reputation highlight that:
  - global reputation is significant only for users having an ego-network small enough.
  - the use of the sole local reputation is the best choice to minimize the average error in predicting the users' interests for the items

- These two results are important to design a recommender system for OSNs having large ego-networks for most of the users because the use of a global reputation mechanism, as in many real OSNs, gives negligible benefits, in terms of quality, if compared with the usage of recommendations only coming from the user ego-network.

Example of ego-networks



(A) An example of OSN; (B) The ego-network of node a; (C) The nodes involving in the computation of $\lambda(a, b)$ - links label the contributions -

Trust Ego-networks

- Let $G_S = \langle N, A \rangle$ be a trust network and let $u \in U$ be a node. The trust ego-network associated with u is a sub-graph $G_u = \langle N_u, A_u \rangle$ defined as follows:
  - a node $v \in N$ belongs to $u \in U$ if and only if ∃ (at least) one path in G_S going from u to v.
  - an edge $\langle u, v \rangle$ belongs to A_u if and only if the two following conditions hold true:
    1) both nodes v and w belong to N_u
    2) there is an arc $\langle u, v \rangle$ in A_s

- A trust ego-networks consists of a node, called ego, and the nodes, called alters, to which the ego is connected to. Ties among the alters will be included in the trust ego-network

- The identification of the trust ego-network with ego u requires to traverse the trust network G_S starting from u. Such a procedure can be implemented through a Depth First Search or a Breadth First Search procedure to linearly scales in the number of arcs in G_S

Local Trust Ego-networks

- Let $G_S = \langle N, A \rangle$ be a trust network, $u \in N$ be a user and $G_u = \langle N_u, A_u \rangle$ be the trust ego-network for u.
- For a target user v, the local trust network $L(u, v)$ associated with u and v is defined as the set of members of G_u who have also trusted v as $z: z \in v \; \exists \langle z, v \rangle \in A_u$
- $L(u, v)$ consists of the users who belong to the trust network of u (G_u) but, in their turn, they have also provided an opinion about v.
- Opinions provided by members of $L(u, v)$ are relevant to estimate how the community centered on u perceives v as trustworthy.

Local Trust Ego-networks

- We aggregate opinions of members of $L(u,v)$ about v to get a trust estimation $\lambda(u,v)$ as:
  - for each node $k \in L(u,v)$ let us compute the shortest path going from u to k and the weight of the the contribution provided by k to the trust computation is $w_k = \frac{1}{2^{l_{u,k}-1}}$
  - Let us sum all weight contributions w_k and normalize them in such as way as to ensure that $\lambda(u,v)$ ranges in a fixed interval (e.g.,$[0,1]$). The formula adopted for computing the (normalized) local reputation $\lambda(u,v)$ is: $\lambda(u,v) = \frac{\sum_{k \in L(u,v), \forall k \neq u,v} \frac{1}{2^{l_{u,k}-1}}}{max_{z \in U, \forall z \neq u,v}\left(\sum_{h \in L(u,v), \forall h \neq u,v} \frac{1}{2^{l_{u,h}-1}}\right)}$

Local Trust Ego-networks

- In real OSNs, the trustworthiness of a user u on the user v decreases with the distance, in terms of direct knowledge, between u and v in the trust network. In this way, "far" user are less important than those provided by nearby fellows

- More alternatives could be proposed, to model how trust decreases with social distance but our experiments based on the exponential decrement represented in Equation in real OSN context produced the best results with respect to other evaluated alternatives

- The normalization at the denominator of the $\lambda(u, v)$ equation allows to represent the importance that v has for u for all the other nodes of the network, rather than in an absolute way. In other words, if the denominator of the $\lambda(u, v)$ equation has, for instance, a relatively small value for the node v, but it has also more small values for the other network nodes, the importance of v will be relatively high with respect to the other nodes.

Local Trust Ego-networks

- An important characteristic of $\lambda(u, v)$ is the number of nodes that contribute to its computation, denoted by $|L(u, v)|$
- If $|L(u, v)|$ is very small, it obviously means that u will not have sufficient information about v from his/her ego-network.
- More structured formulations might be adopted in order to compute $\lambda(u, v)$ and the current choice strictly depends on the necessity of achieving a good trade-off between obtaining accurate results and the need of producing computational models easy-to-understand.

Ego-networks: Experiments

- The proposed approach has been tested on CIAO data set (>35000 records)

- First, we examined how the global error $\varepsilon = \varepsilon(\omega, \tau)$ depends on the mean $|L(u,v)|$ of the user u, i.e. on the following parameter, denoted by ρ_u and computed as $\rho_u = \sum_{v \in U} \frac{\lambda(u,v)}{|L(u,v)|}$

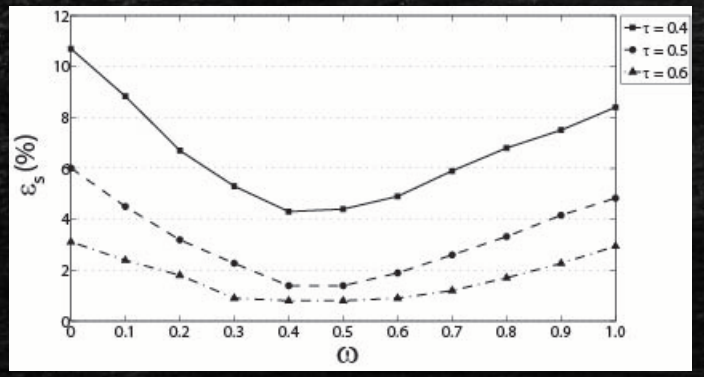
- The relationship $\varepsilon = \varepsilon(\omega, \tau)$ is reported for different values of ω and τ, and for 4 users having different values of ρ. For all the users the error generally decreases with the parameter τ, obtaining the best results in correspondence of τ =0,6. For values of τ higher than 0.6, the error increases, thus we have only represented the curves corresponding to some values of τ >0,6

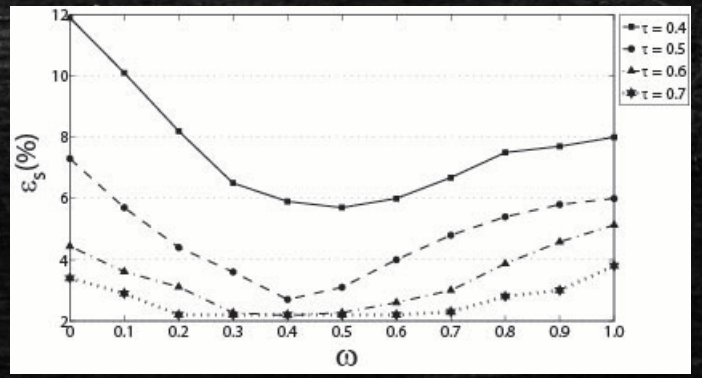
Ego-networks

- Statistics of the dataset Ciao
- Attribute Value
- Number of Users 7252
- Number of Items 21880
- Number of Ratings 183749
- Rating Density 0.0012
- Number of Social Relations 110536
- Social Relation Density 0.0021

Ego-networks: Experiments

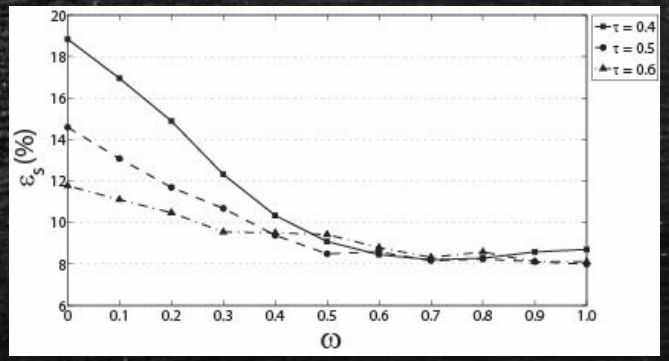
- The user 101 has a small value of ρ_u =733, and the error has a minimum for (ω,τ)=(0,5,0,6). For this type of user, it is important to merge both local and global reputation in equal measure, due to the fact that the local ego-network is not sufficiently large to suggest a correct trust without the help of the global reputation.
- User 1, having ρ_u=2560, again presents a minimum error in (0,2,0,7), but the difference from using high values of ω is less important (if we would set ω=1 to exploit only the local reputation).

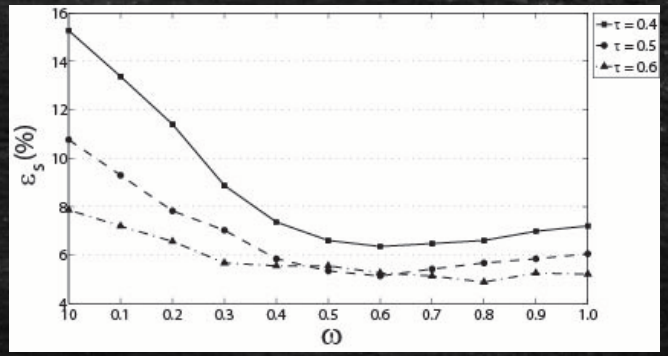




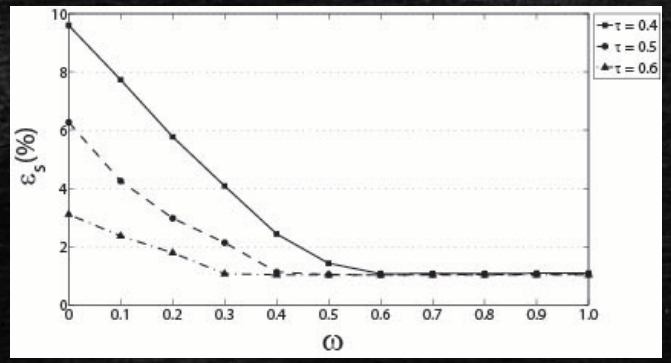
Ego-networks: Experiment

- For users having very high values of ρ_u the influence of the global reputation is negligible.





- The global error averaged on all the 2600 CIAO users shows that it is possible to obtain the minimum average error for the entire community, also with $\omega = 1$, avoiding to consider the global trust.



Summary IoT Security: Trust and Blockchain module

- Trust
- Blockchain

Modulo 2: Blockchain

- Blockchain technology
- The components of a blockchain
- Smart contracts
- Blockchain digital transformation
- Impact of blockchain on the world

The Blockchain technology

- The Blockchain is a digital technology that has emerged in recent years, whose potential is still largely unexplored, which allows you to create and manage a distributed database for managing transactions shared between nodes of a network.
- The blockchain is a technology that falls under the so-called trusted computing
- The basic philosophical idea was to find a way in which decisions could be distributed and power could no longer be concentrated in the hands of a decision maker or a few economic, institutional and non-institutional actors.

The Blockchain technology

- The blockchain is a decentralized database that allows you to create unique digital assets on a Peer-to-Peer network. Duplicating an asset (e.g. a digital coin) resets its value to zero.
- The blockchain technology allows you to create and manage a large distributed database (ledger), structured in blocks containing multiple transactions that can be shared between multiple nodes of a network that once validated by the network itself, through the examination of each block, become unmodifiable (if not through the re-proposal and a new validation by all nodes)

nb: *the transaction is not deleted, but canceled and re-proposed a new version subject to a new validation process*

The Blockchain technology

- The nodes of the blockchain form a network where in each node know, control and approve all transactions and store all the blocks with all transactions allowing their traceability
- Data related to transactions in the blocks are managed by encryption
- It is considered the basis of the so-called Internet of transactions

Why it is a relevant technology?

Time saving

Transactions take place almost instantaneously rather than in days

Cost reduction

With regard to general and intermediation expenses

Risk reduction

Combats tampering, fraud and cybercrime

Increase the Trust

Through processes of sharing and record keeping

What are the advantages?

Shared Ledger

It's a distributed system of shared record only

Privacy

Ensures adequate visibility; secure, authenticated and verifiable transactions

consensus

All parties accept a transaction verified on the network

Smart contract

The contractual terms are incorporated into the transaction database and executed with the transactions

The Blockchain technology

- In the early 1990s, the scientific community tried to solve the problem of having an encrypted and irreversible data structure for the (secure) distribution of documents.

- You can see this as a Trust problem, because the use of a "secure" mechanism in the management of data generates the trust of users

- The Blockchain is similar to the registers managed by public or private central authorities in which we trust (registry, banks, etc..) with the difference that it is a distributed system without a central

The Blockchain technology

- At the end of the 2000's, an attempt is made to find a way to create a digital currency that is disconnected from a central authority and that is able to resist fraud, attack, anonymous, non-duplicable (double-spending problem), fair and safe.
- In 2008 Satoshi Nakamoto creates the Bitcoin crypto-currency
- *The circulation of money (the recognition of its nominal value) depends only on the trust that who receives in payment a certain amount of money has to be able to transfer this money to other subjects in exchange for other goods and services. This "trust mechanism" guarantees that the nominal value is also the real value of money* [Wikipedia].

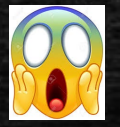
The Blockchain technology

- In 2008 Satoshi Nakamoto *(pseudonym of the creator of the Bitcoin cryptocurrency -BTC or XBT-*) published the "Bitcoin" protocol, an open source software designed to implement the communication protocol and peer-to-peer network behind the cryptocurrency.
- In 2009 he distributed the first version of the client software and contributed to the development of the project (*always anonymously*) until 2010. Since 2011 we lost the (virtual) traces of S. Nakamoto
  - The 03/01/2009 saw the light the first block (genesis block) of Bitcoin
  - On 12/10/2009 5,050 BTC were purchased for 5.02 US$
  - On 22/05/2010 were purchased 2 pizzas (~25 US$)
  - On 17/12/2017 a Bitcoin was worth 19,800 US$
  - On 02/12/2020 a Bitcoin was worth 15,659 US$

The Blockchain technology

- In 2013 is proposed by Vitalik Buterin the creation of the blockchain Ethereum whose development was partly funded by the market (the DAO), in 2014 was distributed the so-called Yellow paper, on 30 July 2015 was created the genesis block of Ethereum and the platform came out of beta on 14 March 2016
- Ethereum is an open-source project and allows the creation and Peer-to-Peer publication of smart contracts created in a proprietary programming language that is Turing-complete.
- Ethereum is equipped with its own virtual coin the Ether

The Blockchain technology

- Bitcoin and Ethereum are the two best known Blockchain
- There is currently a plethora of Blockchain exclusive for both cryptocurrency and smart contracts
- In recent years Blockchain have spread in many realities and many studies exist to expand their scope (data storage, voting, business, transport, etc.).
- Blockchain have a cost and applications must take this into account
- However, it is clear that the assumption Blockchain= ₿ bitcoin IS FALSE, even if at 02/12/2020 7532 cryptocurrency (+2000  in 18 months are active, https://coin.market/coins/info)

The Foundations

- The mathematical bases are:
  - Cryptography (asymmetric)
  - The digital digest (hashing)

- The computer bases are:
  - Internet
  - Peer-to-Peer architecture
  - computational capabilities

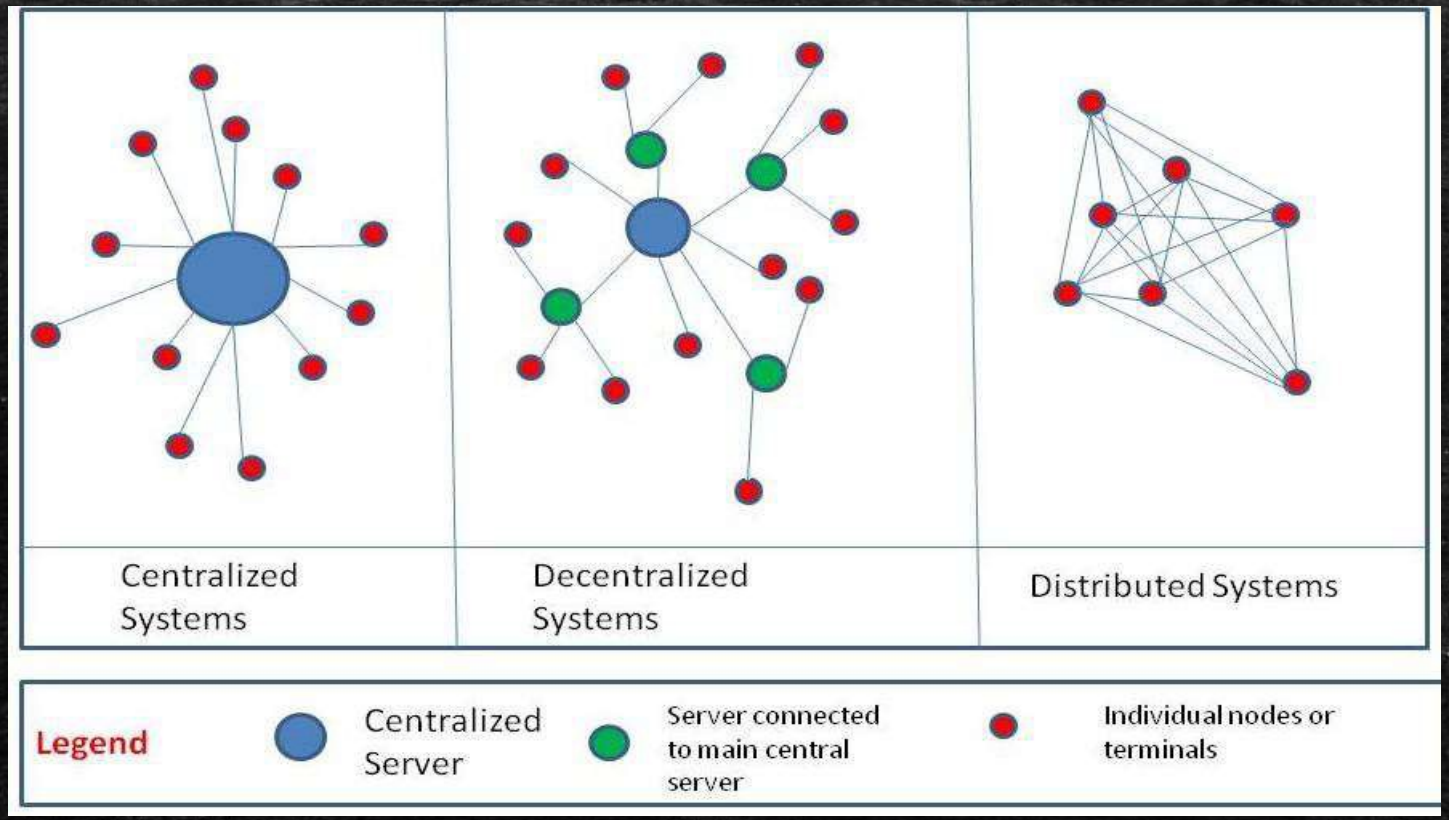
Cryptography and Hashing

- It uses asymmetric key encryption based on two complementary encryption keys (one private known only to the sender and one public known to the rest of the world).
- What is encoded with one key can be decoded only with the other key and vice versa and the exclusive possession of the private key guarantees the functioning of the mechanism.
- The Message Digest (H) is a relatively small numerical summary of the message obtained through Hash functions (e.g., MD5 and SHA 256).
- It is impossible to go back from the Message Digest to the message in clear (for this reason the algorithm applied is also called Hashing Monodirectional)

The Architecture

- In centralized systems there is a central authority (or server) and all other nodes behave as clients who accept messages and execute orders.
- In decentralized systems there are several servers that receive messages from one central server. The individual nodes are connected to the secondary servers.
- In distributed systems there is no central authority or hierarchy. All nodes have the same authority (but not necessarily the same features such as storage, computational capacity, etc.).

The Architecture



The Ledger

- Initially the Ledgers adopted the centralized logic as in the paper version. Someone was in charge of the data entry phase, others managed the systems and, finally, there were those who managed the data extraction and/or processing with a centralized approach
- The Distributed Ledger is the fulcrum of the blockchain. That is, the total absence of intermediary entities, i.e. there is no longer a governance role, replaced by trust between all actors. No one will prevail in decision-making during a transaction
- The blockchain is an inherently decentralized system, composed of several actors acting according to certain personal incentives and based on the information available to them

The Central Ledger

- A Central Ledger operates on the basis of the trust that everyone has in its manager (e.g., public administrations, banks, etc.)
- This trust enables transactions between people mutually unknown, and in the absence of mutual trust, because a trusted third party (the manager of the Central Ledger) guarantees the parties
- The Central Ledger manages the grants for access to information and all the governance rules of the Central Ledger
- Initially, the digitization has made the Ledger faster, user-friendly, performing and with new features, but without changing the centralized logic even when passing on the Internet the relationships became virtual

The Distributed Ledger

- The Blockchain provides the same functionality as the Central Ledger, but without a central entity verification, control and authorization
- The first difference is that the Ledgers are multiple and accessible to everyone
- The second is that everyone can execute a transaction, or change an existing one, but only if all (or most) of the nodes agree to implement it because they agree that it is legitimate (i.e. the author is authorized)

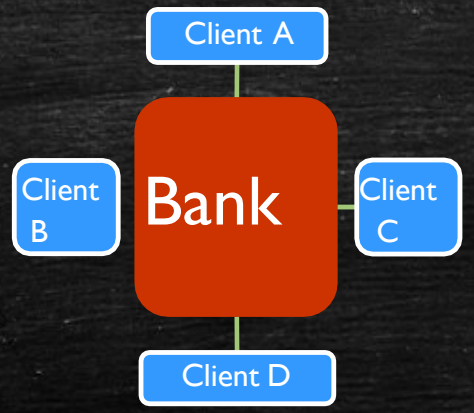
The Distributed Ledger

- All participants can verify the requirements of who is going to make the transaction. The checks are performed reliably and automatically by each actor

- Each operation contributes to create a fast and safe Ledger system both because it is distributed to all nodes (which have a copy of each operation) and because it can withstand manipulation

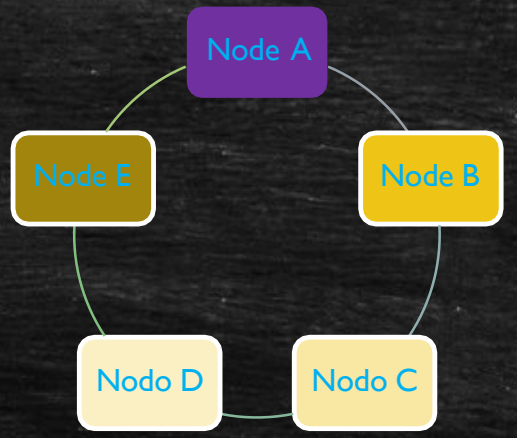
The Distributed Ledger

Central Ledger



- There are more ledgers, but the Bank holds the "golden share".
- The Client must reconcile his ledger with that of the Bank and convince it of the "real state" of his ledger in case of discrepancies.

Distributed Ledger



- There is a ledger and all nodes have some level of access to the ledger
- All nodes accept a protocol that determines the "real state" of the ledger at any time (this protocol is also called "reaching consensus").

The Problem of Byzantium

- The problem of the two generals (1975) refers to a scenario in which two generals have to attack a common enemy. General 1 is the leader of the operation, General 2 is an executor ("the follower")
- To defeat the enemy army the two generals will have to cooperate by attacking the enemy simultaneously from two different fronts and at the same time because their individual armies are not alone adequate to defeat the enemy
- There is no way to guarantee that every general can be certain that the other general has accepted the plan of attack
- Both generals will have the doubt that their last messenger was caught crossing enemy lines
- The problem of the two generals has been defined as unsolvable

Blockchain components

- **Node**: are the participants in the blockchain (participants server)
- **Transaction**: it is made up of data representing an asset (transaction) that must be verified, approved and filed
- **Block**: a set of merged transactions to be verified, approved and archived by the nodes of the blockchain
- **Ledger**: is the public register in which all transactions are stored in a sequential and immutable way. The Ledger consists of blocks chained together by encryption and hashing
- **Hash**: operation (not invertible) that maps a string of variable length into another (unique) string of determined length from which it is not possible to trace the starting string. The Hash identifies in a unique and secure way each block

Blockchain Characteristics

- A Blockchain has the following features:
  - Master books (Ledger) on multiple copies (Distributed data)
  - Accessible
  - Irreversible (history of transactions/smart contracts)
  - Decentralized
  - Encrypted (each actor -ledger, user- is provided with a pair of asymmetric keys)
  - Distributed
  - Equipped with consensus (common knowledge of network management and control processes)

Transaction

- Any transaction (i.e. the data that represent it) is subject to a double asymmetric key signature process that, although without certificates issued by third parties (accredited certifiers) because the blockchain provides for their passing, works with a mechanism similar to that of digital signature

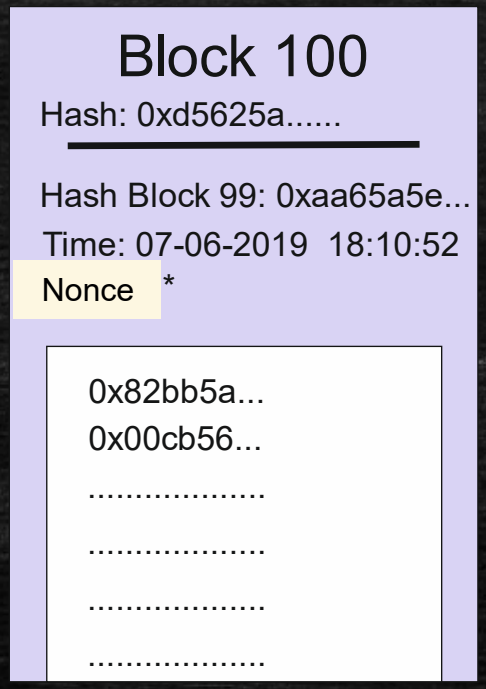
- Distributed Ledger Technologies provide for the use of cryptographic algorithms that enable the user to use the system by providing a public and a private key to use to subscribe to transactions or to activate smart contracts or other services related to the blockchain

Block

- A block is a list of data validated by a consensus process
  - one block contains:
  - list of transactions/smart contracts
  - block generation timestamp
  - message digest (*Hash*) of the previous block
  - nonce (for *Proof of Work* and similar)
  - message digest (*Hash*) of all information listed above
- Obviously, the *genesis block* has no reference to an earlier block
- The fingerprint of the block is obtained through the hashing algorithm
- Once you get the hash, the block is valid, but if the calculated hash is not valid, you have to recalculate a new hash (e.g. using a new nonce)

Block

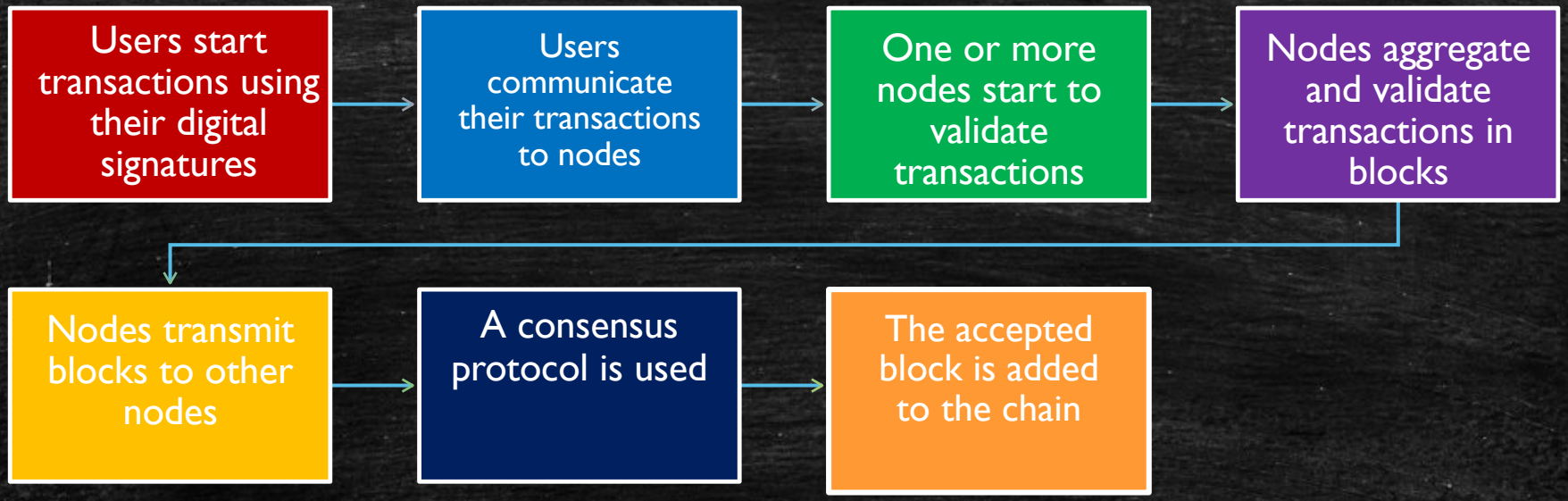
- In the Bitcoin Blockchain Bitcoin about every 10 minutes a new block is added
- A "*complete*" Ethereum node grows about 1 GB per month



The Base Process

- Operation (Transaction/Smart Contract)
  - Sending the data
- Block creation
  - Collection and validation of operations by the miner
  - Transmission to the other nodes of the block
- Checking the block
  - Competition between miners to close the block
- Lock closure
  - The first miner to solve the problem communicates it to others for block validation and its addition in the blockchain
- Adding the block to the chain
  - To the ledger of each node of the network is added the new block

The Base Process



Example

The transaction receives the Digital Signature and the Public Key of the two actors

The transaction is inserted in a Transaction Block

- The transaction is inserted in a new Block with all the data related to the transaction between Luca and Sofia and the data related to the car and the economic capacity of Sofia
- The block, which also includes other transactions, when it is ready is prepared to be verified and approved by the nodes of the blockchain

The Block, with the Transaction, must be verified by the Network

- The block with the transaction is sent to the Network for verification by the nodes of the blockchain

Example

Once verified, the Block is added to the chain

- The new Block is added to the chain of blocks that forms the blockchain, is accessible to all participants and is in the archive of all participants
- The reference of that specific transaction becomes permanent and unchangeable

The transaction is completed and stored in all nodes of the blockchain

- If the information is considered correct, the transaction is authorized, validated and carried out

Miner

- To add a new transaction block (active) it must be checked, validated and encrypted
- This step requires that for each new block is solved ("*Mining*") by the "*Miner*" a complex mathematical problem that requires a great computational effort
- The work of the "*Miner*" is basic in the management of blockchains
- Anyone can become a "*Miner*" and compete to be the first to solve the mathematical problem to create a new block of transactions valid and encrypted to add to the blockchain

Miner

- The miner needs to be remunerated and incentivized. In "*Private*" or "*Permissioned*" blockchain this depends on the authority that has activated the blockchain
- In the blockchain "Public" or "*Permissionless*", this role can be played by each node of the blockchain and the miner is incentivized with forms of remuneration that depend on the type of rules and / or governance active in the blockchain
- Typically, the miner who creates a valid block and adds it to the chain is remunerated by commissions for his transactions that:
  - can refer to unit values per transaction (if these are thousands, the fee can be large)
  - can receive new currency created and put into circulation as a mechanism of inflation (e.g., Bitcoin)

Miner

- Adding a new block to the chain updates the Ledger of all nodes in the blockchain, which will accept a new block when, solved the mathematical problem, both verified the validity of all its transactions.

- If the verification process returns an error, an anomaly, a discrepancy, the block will be rejected and everyone will have visibility that the transaction has not been authorized. Otherwise, the block is created and added to the blockchain as a permanent and unchangeable public record; no participant in the blockchain will no longer be able to change or remove it

Miner

- Blockchain permissionless
  - Everyone can participate
    - The block validation methodology depends on the degree of trust between the nodes. Where there is no basis for trust, consensus can be reached, for example, through the Proof of Work, which requires the algorithmic resolution of a Cryptographic Hash (240 kWh per Bitcoin in 2014)

- Blockchain permissioned
  - The actors are known and trusted
    - An industry, company or group of companies
    - Many of the mechanisms mentioned above are not necessary - or rather, they are replaced by private contracts or writings.
    - The validation mechanism of the blocks depends on the degree of trust between the nodes. Alternatively, where there is no centralized authority, the consensus can be algorithmically determined

Blockchain Permissioned

Permissioned Network

Known actors and access rights within the "*operating domain*".

Confidential Transactions

Transactions are visible to selected actors

No cryptocurrency

Does not require mining and expensive calculations to record transactions

Programmable

Can exploit the logic of smart contracts

Blockchain Permissioned or not

	Bitcoin	Ethereum	Hyperledger
Cryptocurrency required	Bitcoin	Ether, user-created cryptocurrency	None
Network	Public	Public or permissioned	Permissioned
Transactions	Anonymous	Anonymous or private	Public or confidential
Consensus	Proof of work	Proof of work	PBFT
Smart contracts (business logic)	None	Yes (Solidity, Serpent, LLL)	Yes (chaincode)
Language	C++	Golang, C++, Python	Golang, Java

Immutability

- To alter a Central Ledger simply violate the central authority that manages it, in the blockchain you must simultaneously violate all copies of the Ledger owned by all nodes
- An operation (node function) almost impossible
- There can not even be a "*fake*" Ledger because all nodes have only one authentic version that can be used for comparison and verification
- Trust and control of transactions reside in nodes and transactions are decentralized, transparent and open to everyone
- This is a blockchain Permissionless, i.e. "*without permission*" and there is no authority to deny permission to participate in the control and addition of transactions

Immutability

- A blockchain Permissioned requires permissions that depend on governance that, based on its policies, can assign a group of actors management and authority in issuing grants for access, controls, permissions and adding transactions

- The blockchain Permissioned are still transparent, unchangeable and safe as the blockchain Permissionless. The Permissioned could relax the consensus protocol and are managed by public administrations, banks and, in general, by the same type of entities that managed the Central Ledgers

Consensus

- In Distributed Ledger each participant manages one node of the network and is authorized to update the Ledger independently (but controlled) from the other nodes
- Updates (records) are independently created and loaded by each node that processes and controls each transaction (verified, voted and approved by the majority of nodes on the network)
- The autonomy of each node is subject to the achievement of a *consensus* on the operations carried out, only with the *consensus* the transactions are then authorized (and in the case of smart contracts, activated)
- The Ledgers are updated with the latest version of each operation if they have the *consensus* (which becomes permanent and immutable in each node) of each participant who will have a copy - immutable - of each operation

Consensus

- In Distributed Ledger each participant manages one node of the network and is authorized to update the Ledger independently (but controlled) from the other nodes
- Updates (records) are independently created and loaded by each node that processes and controls each transaction (verified, voted and approved by the majority of nodes on the network)
- The autonomy of each node is subject to the achievement of a *consensus* on the operations carried out, only with the *consensus* the transactions are then authorized (and in the case of smart contracts, activated)
- The Ledgers are updated with the latest version of each operation if they have the *consensus* (which becomes permanent and immutable in each node) of each participant who will have a copy - immutable - of each operation

Consensus

- The DLT (or also Shared Ledger) requires a P2P network and algorithms to collect the consensus and the possible confirmation of the operations
- Consensus management models are one of the possible differences between the blockchain, but obviously not all applications of consensus protocol are blockchain

Timestamp

- The Timestamp blocks the alteration or annulment of an operation once it has been performed
- The Timestamp associates a certain and legally valid date and time to a computer document (provides a time validation that can be opposed to third parties)
- The Timestamp is a sequence of characters (in shared and comparable format) that once applied (Timestamping) uniquely, indelibly and immutably identifies a date and/or time of a certain event. This process is one of the basics of how the blockchain works

Distributed Consensus

- To avoid fraud it is necessary to complicate the validation process
- The best-known *Distributed Consent* is perhaps the *Proof of Work* that requires verification and approval based on calculation resources of the blockchain nodes (competing to solve a cryptographic puzzle)
- Whoever finds the solution has the right to validate the block by submitting the *Proof of Work* (proof of the solution of the puzzle) and will then be paid (the how depends on the type of blockchain)
- N.B. The *Proof of Work* allows you to create between "public" nodes, that do not know each other, a relationship of "trust" based on "collaboration" in solving the Proof of Work

Double Spending

- Conceptually, the copy of a digital asset is not distinguishable from the original. If the copied asset is a coin you would have the possibility to spend it infinitely (but its value would be reset to zero)
- In conventional digital transactions it is the banks that guarantee that you cannot spend more money than you have, this is achieved by harmonizing the withdrawal and deposit accounts of the operation with structured processes
- In purely digital transactions the solutions were not fully satisfactory
- The solution presented in *Satoshi Nakamoto's white* paper is to give an identity to the coin (with the support of cryptographic techniques)

Double Spending

- Cryptography allows you to manage the identity of the cryptocurrency, each with its own specific code (ID) and its history
- At each transaction (i.e., passage of the currency from one Wallet to another) the information on the passage of the currency is filed and all nodes of the blockchain will be informed of this transaction
- The currency itself will archive this passage in its history, with all the information necessary to identify the actors.
- This process is incremental with the life of the currency that therefore once spent will no longer be in the availability of the old owner, who will not even have a copy of that "currency"

Double Spending

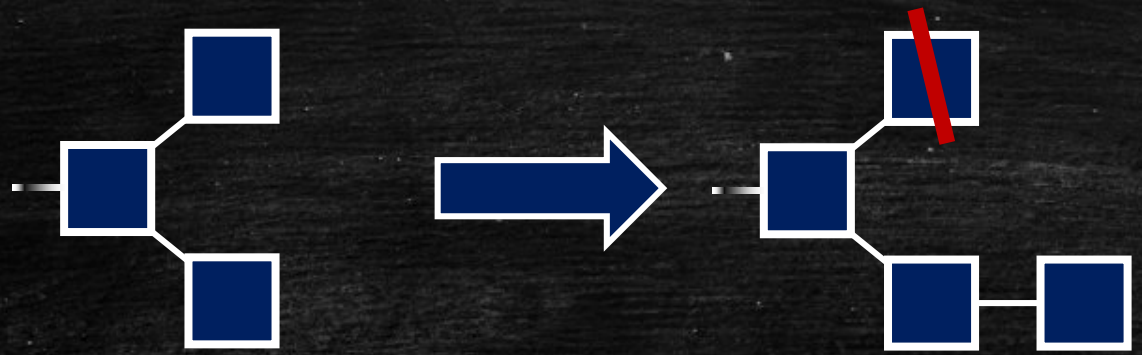
- Each node has the entire transaction history (like the other nodes), but could it store false information to duplicate assets?
- If it could alter the history of the transactions (theoretically possible, but improbable) inserting a false transaction to alter the passage of the ownership of an asset, it would have a situation of potential "Double Spending"
- Going to use this irregular asset you would notice the problem

Double Spending

- The exchange in units of Cryptocurrency requires traceability, i.e. the control of the transaction history from the creation until the moment of the new transaction (e.g. a purchase)
- The proof of the transition to the new property is given by a digital signature of the transaction by the last person who exchanged it and is fixed by a Timestamp
- The block is added and published with its Hash. The previous Hash is also included in the generation of the new one, thus forming a public and shared chain
- With these conditions, it is extremely difficult to achieve a Double Spending situation

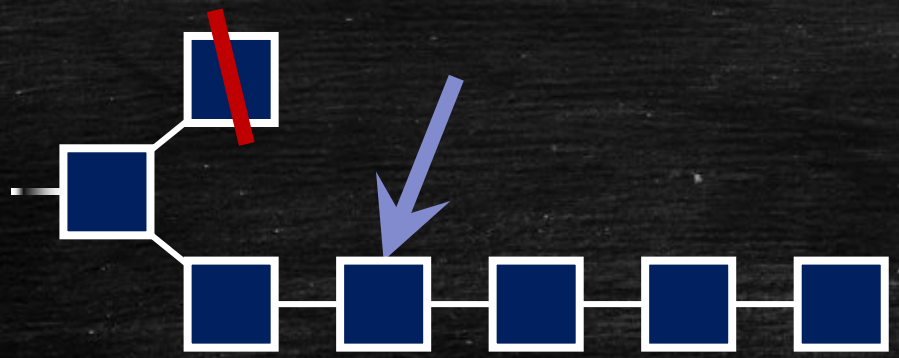
Fork Chain

- Longest chain wins
- Transactions are cancelled
- Double-spending is a danger



Fork Chain

- A transaction is confirmed when "its" chain is long enough



Token

- A token is a digital asset (*a set of digital information that attests a right of ownership to a subject on the same information stored in a blockchain and that can be transferred via a special protocol*) based on the blockchain that can be exchanged between two parties without intermediaries

- A token could also incorporate additional rights as for smart contracts

- The Bitcoin is an example of token, others also use the code such as, for example, the blockchain Ethereum whose tokens allow the management of smart contracts in which transactions give rise to a series of obligations between the parties already programmed in the token

Class 1 Token

- The class 1 token is similar to a currency, has no counterparty and is transferred by means of unchangeable transactions on blockchain
- It has the function to register a right of ownership of the token itself or the existence of a certain subject/object
- The owner has only the right to the ownership of the token itself
- They are class 1 tokens those of cryptocurrency as:
  - Bitcoin
  - Bitcoin Cash
  - Litecoin
  - etc

Class 2 Token

- The class 2 token gives to its owners rights exercisable against the person who generated the token or possibly against third parties
- It is a token that allows to exercise rights towards counterparties
- The class 2 tokens are similar to titles that give the holder the "*right to the benefit indicated in it towards presentation of the title*"
- Examples:
  - bonds or loan securities
  - the equity securities
  - securities representing goods and legitimation documents

Class 2 Token

- **Smart contract token for the management of future payments -** gives a right to have future payments according to certain contractual conditions that the token automatically manages
- **Token as an asset** - the token represents a kind of right of ownership of a certain asset (tangible or intangible). E.g., shares of the issuing legal entity or of third parties
- **Token for "standardized" payments** - gives the right to receive a payment for a well-defined amount
- **Token for the management of service provision** - gives the right to a specific service or good from the emitter or a third party who has signed a commercial agreement. E.g., access to computer infrastructure, provision of services, etc.

Class 3 Token

- The class 3 token has a mixed function
- It can represent:
  - rights of co-ownership
  - right to vote
  - economic rights for legal representatives
  - etc.
- This type of token does not give the holder a right in rem towards the issuer of the security or third parties

Token +

- There are also Labelled Tokens or Token+ or Labelled Tokens (LB)
- They are tokens that have associated a series of metadata (labels) for which the exchange is conducted on a secondary market, through Smart Contracts on blockchain Ethereum Mainnet, and are adopted in many international projects.
- The Token+ has five great advantages
  - is uniquely labeled and associated with metadata;
  - it is not divisible;
  - lives' digitally on the blockchain;
  - can be traced in its history of passages on the blockchain
  - differently manageable for single label according to meaning/value

Token +

- Token+ can be used as a financing instrument and in particular in four legally legal areas:
  - Business Projects
    - innovative based on blockchain
    - innovative NOT based on blockchain
    - NOT innovative
  - Tangible assets (real estate, works of art, etc.) or intangible assets and services (e.g. Software) of any nature whatsoever

- As for the "traditional" Coin/Tokens in the various countries there is discussion about their compliance with the rules on privacy, investor or consumer protection, anti-money laundering, identity

Let's summarize

- The digital signature ensures that the actors of any message (e.g., a payment) are identified in a certain way
- A node randomly chosen by a mathematical model can validate that a set of messages with the timestamp and communicate it and write it in the register of the other nodes of the network irreversibly
- All operations are quickly confirmed by the distributed consent process called "Mining“
- The correctness of the block of operations entered into the network is verified by the nodes with the updated version of the blockchain
- The first node that gets green light communicates it to all others, which validate the block and update the blockchain preserving the chronological order of operations and net neutrality

Permissionless o Permissioned Ledger

- Permissionless Ledgers (e.g. Bitcoin) are open, without central ownership or governance, and are designed to be uncontrolled
- Allow everyone to contribute to the Ledger update and everyone has, as participants, immutable and identical copies of all transactions due to consensus
- Any form of censorship is prevented, no one can prevent a transaction from taking place and being added to the Ledger once it has won the necessary consensus among all nodes (participants) in the blockchain
- They can be used as a global database is immutable over time

Permissionless o Permissioned Ledger

- Permissioned Loggers can be controlled, have ownership, governance and behavior rules for data access and visibility
- The addition of a new record does not require the approval of the majority of participants, but only those assumed Trusted and therefore Permissionless Ledgers are more performant and faster than Permissionless Ledgers
- They respond to the need for widespread updating across multiple actors who can operate independently, but with control limited to those who are authorized
- Compared to Smart Contracts, Permissionless Ledger will introduce changes in some professions

Permissioned Ledger

- Permissioned ledgers compete:
  - Infrastructure
  - Ecosystem
  - Applications
  - Governance
- Permissioned blockchain infrastructure requires reliable and extensively tested private or closed networks
- The security of these solutions is directly related to the ability to ensure the impenetrability of the network from unauthorized parties
- The infrastructure consists of networks and nodes

Permissioned Ledger

- Permissioned blockchains are populated by actors that as users or service providers to the blockchain, share its purpose
- Application and service providers to the blockchain are required to work in the form of close and controlled partnership with infrastructure providers
- The application component in private blockchains is closely linked to the technological and governance logic defined by the companies active on the infrastructure

Permissioned Ledger

- The Permissioned blockchain is based on a set of rules shared by all actors since the conception and design phase from infrastructure to applications
- Governance is an integral part of the design process and compliance with the rules to ensure the security of the blockchain and the achievement of business objectives

Fork

- Forks are tools used to improve blockchain performance and management and are divided into:
  - Soft Fork: reversible update of the blockchain protocol compatible with previous releases that does not prevent "outdated" nodes from participating
  - Hard Fork: irreversible change that requires participants to upgrade and can be Planned or Contentious:
    - Planned: protocol change is planned and approved by community participants, does not split the blockchain, and rules are updated continuously
    - Contentious: the change is not unanimously approved and with the Hard Fork a contextual splitting of the blockchain is generated leading to a splitting of the old blockchain

Fork

- Pros and cons of a Hard Fork:
  - Weakening of the Trust
    - Trust in blockchain is closely related to the number of participants, a Hard Fork produces a split between nodes
  - Democratic management of the blockchain
    - The probability of winning a Proof of Work is proportional to the computational power which, in turn, is proportional to the availability of financial resources.
  - enhance performance and scalability
    - E.g. increase the number of transactions per second
  - Improving Network Governance
    - To drive the system to new solutions, preserve its value, create new value, …

Smart Contract

- A Smart Contract [Nick Szabo] (born in the 70s for the activation-deactivation of software) is the transposition in code of a contract that automatically, on the basis of the clauses agreed by the parties, verifies the occurrence of certain events and autonomously performs some actions (or provision for the execution of actions) when the conditions established by the parties are reached and verified.

- In the absence of human interpretive input, the Smart Contract must be precise in all its terms, including the data sources that act as triggers for the execution of the contract terms

- Obviously data and information must
  - Have a provenance that is certain and identified
  - Interpreted according to identified and precise rules

Smart Contract

- The Smart Contract is in fact a program that processes data and information in a deterministic way
- The deterministic execution of the code guarantees the reproducibility of the output with the same input excluding any form of interpretation
- Contractors must define conditions, clauses, modalities and rules of control and action, if the contract is accepted (i.e. is in the form of code and therefore a Smart Contract) the effects are independent of their will

Smart Contract

- The current Smart Contracts require blockhain to guarantee the "trust" that used to be assured by a centralized authority.
- In parallel to Smart Contracts, the study of a specific semantics and meta-learning solutions has also developed.
- It goes from intelligent automation to semantic contract to learn and adapt its behavior according to the acquired notions
- With the use of semantic techniques and AI, errors will be reduced (perhaps) and content knowledge will increase, influencing the contract drafting phase
- It is always imperative to ensure the determinism of a Smart Contract

Bitcoin

- Bitcoin is one (of many, but remains the first) cryptocurrency
- To operate with Bitcoins you must install a Bitcoin wallet application on your device
- Installed the wallet, will be generated a first Bitcoin address (shareable and theoretically disposable) that will be used to receive currency from the person with whom the address was shared
- All confirmed transactions are saved in the Blockchain, and from your wallet you can check the amount of your Bitcoin deposit, i.e. the Bitcoins you can spend
- The whole system is protected from cyber attacks by encryption

Bitcoin

- Money is transferred between two Bitcoin wallets with a transaction protected by a private key that ensures that the Bitcoins really belong to the person who is making the transaction, and that no one changes this transaction (secure transaction)
- Bitcoins have a value exactly like traditional currencies
- It went from barter to primitive forms of coins (e.g., shells, worked stones), to precious metal coins (e.g., gold circulation), to paper money (first fully and then partially convertible to gold), to the abolition in 1971 of gold convertibility (Richard Nixon, 1971)

Bitcoin

- The assumption is that current currencies do not have a physical equivalent (e.g. gold) but are used because we are sure that if we accept them, then we can use them in exchange for some good because others accept them as we do.
- In the case of state currencies there is the trust that since it is a state currency, in the end the state will guarantee its value.
- In the case of Bitcoin the trust is technologically distributed through the blockchain and the solution to the problem of double spending

Bitcoin

- Bitcoin does not respond to a central entity, but is governed by a distributed database, where each node in the network keeps track of each transaction
- Informatically, a Bitcoin is a file saved locally (on the wallet)
- Each "bitcoin address" in the wallet can be tied to a variable number of Bitcoins
- To facilitate transactions (and promote anonymity) a new address is generated for each of them
- Each address (public key), is associated with the equivalent of a digital signature (private key), so that only the owner of an address can associate it to a transaction

Bitcoin

- Transaction:
  - user A downloads X Bitcoin from his wallet
  - user A uploads X Bitcoin to user B's wallet
  - each node updates the register and transfers the information to the next node
- Operationally:
  - A activates an address (as a public key) where to report the transaction
  - B identifies one of its addresses (referring to a certain number of Bitcoins) and starts the transaction
  - B adds the public key activated by A to its address, joins the private key in the role of digital signature to verify the transaction requirements
  - No node will keep track of the balance of A and B
- The ownership of a Bitcoin is attested by past transactions
- Operationally, a transaction consists of a series of inputs (each referring to a Bitcoin address) connected to previous transactions

Bitcoin

- To transfer X Bitcoin from A to B

- With 1 input specifying a number Y greater than Bitcoin
  - first transaction - A moves Y Bitcoin to B
  - second transaction - B returns Y-X Bitcoin to A
- With 2 or more inputs specifying ad, example K and W Bitcoin with X=K+W
  - first transaction - A moves K Bitcoin to B
  - second transaction - A moves W Bitcoin to B
- The network nodes check the inputs to validate the sum, tracing back to the first Bitcoin transaction involved
- For greater efficiency, by installing the wallet you also download the history of all operations performed, immediately processed to attest its authenticity
- This operation can take up to several hours (depending on the number of Bitcoins).

Bitcoin

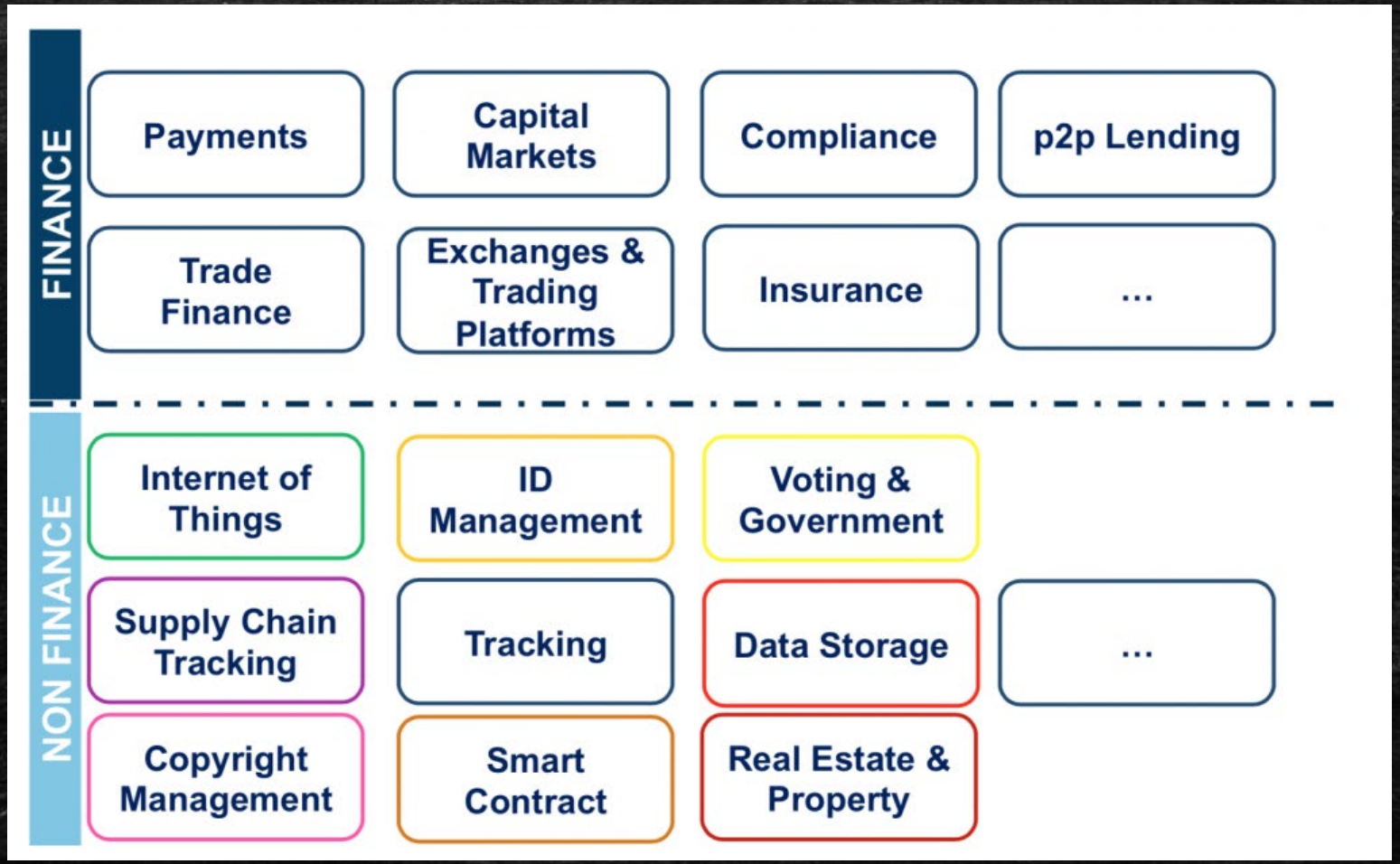
- Bitcoin provides security through:
  - The private key: which ensures the real owner of X Bitcoin to carry out a transaction for those X Bitcoins
  - The control of the previous inputs: which ensures the actual possession of the Bitcoins necessary for the transaction.
- There is a risk related to the order of transactions
- Launching two transactions - potentially conflicting since they are tied to the same input - does not guarantee that they will be transferred in the correct order (the order will be random)
- A node may receive the first transaction after the second (this problem is not solvable)

Side Chain

- SideChains mediate between the advantages/disadvantages of Permissionless and Permissioned blockchain, allowing to generate transactions or exchange of assets on one blockchain, manage them on another and return, if needed, to the one that generated them

- Sidechains allow to manage transactions on the public blockchain (main chain) ensuring the interchangeability of assets, with management and registration on external environments (private blockchain) with the advantages of permissionless (efficiency and access control) integrated with the control of permissioned used for some needs

Bitcoin Technology Application



Ethereum

- Ethereum is a computational platform that makes available an immutable, shared repository of all tasks and is designed so that it cannot be stopped, blocked or censored
- Tasks are remunerated in Ether (ETC), a token treated as cryptocurrency, which represents both the processing power required to produce contracts and to pay for their realization
- Ethereum is a Programmable blockchain that allows users to create their own "operations", i.e. decentralized (moving from Distributed Database to Distributed Computing) and customized blockchain applications.
- The Internal Transaction Pricing Mechanism (GAS) of Ethereum allows to optimize network resources, to prevent spam and to allocate resources function of requests

Ethereum Virtual Machine EVM

- Ethereum is "Turing complete" and allows to create applications for the Ethereum Virtual Machine (EVM) with programming languages related to JavaScript and Python.
- EVM represents the runtime environment for the development and management of Smart Contracts in Ethereum
- EVM operates separately from the Network (protected) and the code it manages does not have access to the Network and each Smart Contract is separate from other Smart Contracts
- In 2016 Ethereum there was a hard fork (following a breach) and Ethereum gave rise to two different blockchains, Ethereum Foundation and Ethereum Classic

Ethereum Foundation e Classic

- Ethereum Foundation (non-profit) is the (Swiss) organization that manages the Ethereum platform (2014 - Vitalik Buterin with Anthony Di Iorio, Mihai Alisie and Charles Hoskinson)
- The latest ongoing project, Serenity is expected to innovate the algorithm that handles Ethereum's consensus
- Ethereum Classic was born from the fork of Ethereum Foundation and is managed by a different team than Ethereum Foundation. It is fully compatible with Ethereum technology and maintains its history. Ethereum Fondation is, in essence, a new blockchain.

Ethereum

- Ethereum Foundation is the "official" version managed and updated by the creators that was born as a result of a hack of an Ethereum funding activity (The DAO)
- Ethereum Foundation is based on the fact that if the majority of the community can decide on possible changes to the blockchain
- Ethereum Classic is based on the principle that the blockchain cannot be modified and therefore they continue to operate on the old version
- In practice, the fork has created two parallel Ethereum blockchains

NEM

- New Economy Movement (NEM) Foundation encourages the development and extension of an ecosystem of NEM users at all levels:
  - NEM Foundation (the New Economy Movement Foundation)
  - NEM blockchain (on which the services are based)
  - XEM the NEM cryptocurrency

- For many, NEM is a cryptocurrency (called XEM based on a token) and a set of solutions for managing blockchain-based services that go well beyond payments

- Specifically, NEM is an interpretation of the blockchain that is based on the Proof of Importance (POI) process, a reputation control trust system (EigenTrust) on multi-signature accounts and crypto messaging

Proof of Importance

- It is based on an algorithm used in NEM transactions and which establishes the importance of a user based on how many XEMs he has and according to the number of transactions he makes with his wallet
-
- The importance of a transaction is based on the currency involved and the nature of the transactions
- The POI uses a ranking system called NCDawareRank that measures and monitors the consensus process
- The POI evaluates NEM transactions primarily using volume and Trust criteria of each transaction

Hyperledger

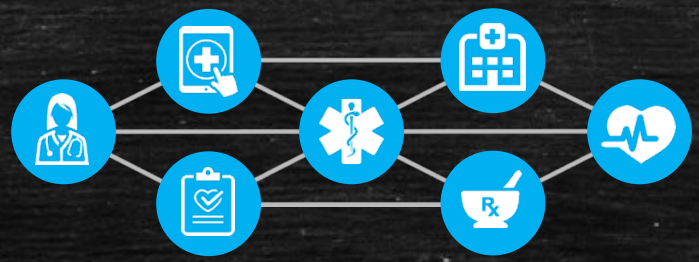
- In December 2015, the Linux Foundation created the Hyperledger Project. The founding members of the project were announced in February .
- The objective of the project is to advance cross-industry collaboration by developing blockchains and distributed ledgers, with a particular focus on improving the performance and reliability of these systems to support business transactions
- The project will integrate independent open protocols and standards by means of a framework for use-specific modules, including blockchains with their own consensus and storage routines, as well as services for identity, access control and smart contracts
- In early 2016, the project began accepting proposals for incubation of codebases and other technologies as core elements. One of the first proposals was for a codebase combining previous work by Digital Asset, Blockstream's libconsensus and IBM's OpenBlockchain.[6] This was later named Fabric

Shared Ledger Database

Blockchain allows multiple different parties to securely interact with the same universal source of truth

Finance

Streamlined settlement, improved liquidity, increased transparency and new products/markets



Healthcare

Unite disparate processes, increase data flow and liquidity, reduce costs and improve patient experience and outcomes

Supply Chain

Track parts and service provenance, ensure authenticity of goods, block counterfeits, reduce conflicts

Hyperledger Goals

Where open source teams build diverse approaches for business blockchain technology systems

Create enterprise grade, open source, distributed ledger frameworks & code bases

to support business transactions

Provide neutral, open, & community-driven infrastructures

supported by technical and business governance

Build technical communities

to develop blockchain and shared ledger POCs, use cases, field trials and deployments

Educate the public

about the market opportunity for blockchain technology

Promote our community of communities

taking a toolkit approach with many platforms and frameworks

Hyperledger Modular Umbrella Approach

Infrastructure
Technical, Legal, Marketing, Organizational

Ecosystems that accelerate open development and commercial adoption

THE LINUX FOUNDATION

Cloud Foundry | Node.js | **Hyperledger** | Open Container Initiative

Frameworks
Meaningfully differentiated approaches to business blockchain frameworks developed by a growing community of communities

Hyperledger **Indy** | Hyperledger **Fabric** | Hyperledger **Iroha** | Hyperledger **Sawtooth** | Hyperledger **Burrow**

Tools
Typically built for one framework, and through common license and community of communities approach, ported to other frameworks

Hyperledger **Quilt** | Hyperledger **Composer** | Hyperledger **Explorer** | Hyperledger **Cello**

Hyperledger Business Blockchain Frameworks

- **Hyperledger Fabric**: To develop applications or solutions with a modular architecture. It includes components, such as consensus and membership services, to be plug-and-play.
- **Hyperledger Iroha**: A business blockchain framework designed to be simple and easy to incorporate into infrastructural projects requiring distributed ledger technology.
- **Hyperledger Sawtooth**: A modular platform for building, deploying, and running distributed ledgers adopting the Proof of Elapsed Time (PoET), which targets large distributed validator populations with minimal resource consumption.
- **Hyperledger Burrow**: A permissionable smart contract machine which provides a permissioned smart contract interpreter built in part on the Ethereum Virtual Machine
- **Hyperledger Indy**: Tools, libraries, and reusable components for providing digital identities rooted on blockchains or other distributed ledgers so that they are interoperable across administrative domains, applications, and any other silo.

Hyperledger Technical Scope

	Custom Applications	App Layer
Out of Scope		
In Scope	API libraries and GUIs Specialized consensus algos Membership policies Gateway Operations dashboard	**Value Added Systems**
	Core APIs	**Core APIs**
	Code execution environment Ledger data structures Modular consensus framework Modular identity services Network peers	**Shared Ledger**

Hyperledger Fabric

- The Hyperledger project encourages a collaborative approach to develop blockchain technologies via a community process, with intellectual property rights that encourage open development and the adoption of key standards over time.

- Hyperledger Fabric is one of the blockchain projects within Hyperledger. Like other blockchain technologies, it has a ledger, uses smart contracts, and is a system by which participants manage their transactions.

- Where Hyperledger Fabric breaks from some other blockchain systems is that it is private and permissioned. Rather than an open permissionless system that allows unknown identities to participate in the network (requiring protocols like “proof of work” to validate transactions and secure the network), the members of a Hyperledger Fabric network enroll through a trusted Membership Service Provider (MSP)

Hyperledger Fabric

- Hyperledger Fabric also offers several pluggable options. Ledger data can be stored in multiple formats, consensus mechanisms can be swapped in and out, and different Membership Service Providers are supported.

- Hyperledger Fabric also offers the ability to create channels, allowing a group of participants to create a separate ledger of transactions.

- This is an especially important option for networks where some participants might be competitors and not want every transaction they make — a special price they're offering to some participants and not others, for example — known to every participant. If two participants form a channel, then those participants — and no others — have copies of the ledger for that channel.

Shared ledger

- Hyperledger Fabric has a ledger subsystem comprising two components: the world state and the transaction log. Each participant has a copy of the ledger to every Hyperledger Fabric network they belong to.

- The world state component describes the state of the ledger at a given point in time. It's the database of the ledger. The transaction log component records all transactions which have resulted in the current value of the world state; it's the update history for the world state. The ledger, then, is a combination of the world state database and the transaction log history.

- The ledger has a replaceable data store for the world state. It simply records the before and after values of the ledger database being used by the blockchain network.

Smart Contracts

- Hyperledger Fabric smart contracts are written in chaincode and are invoked by an application external to the blockchain when that application needs to interact with the ledger. In most cases, chaincode interacts only with the database component of the ledger, the world state (querying it, for example), and not the transaction log.

- Chaincode can be implemented in several programming languages. Currently, Go and Node are supported.

Privacy

- Depending on the needs of a network, participants in a Business-to-Business (B2B) network might be extremely sensitive about how much information they share. For other networks, privacy will not be a top concern.

- Hyperledger Fabric supports networks where privacy (using channels) is a key operational requirement as well as networks that are comparatively open.

Consensus

- Transactions must be written to the ledger in the order in which they occur, even though they might be between different sets of participants within the network. For this to happen, the order of transactions must be established and a method for rejecting bad transactions that have been inserted into the ledger in error must be put into place.
- There are many ways to achieve it, each with different trade-offs. For example, PBFT (Practical Byzantine Fault Tolerance) can provide a mechanism for file replicas to communicate with each other to keep each copy consistent, even in the event of corruption. Alternatively, a mining process to solve a cryptographic puzzle can be defined
- Hyperledger Fabric has been designed to allow network starters to choose a consensus mechanism that best represents the relationships that exist between participants. As with privacy, there is a spectrum of needs; from networks that are highly structured in their relationships to those that are more peer-to-peer.

Things to address for enterprise use ts

1. Ledger should not be shared with everyone
   - *I don't want to share all of my data with every participant.*
2. Users should not be anonymous
   - *I want to know who my users are. Anonymity does not benefit me.*
3. Users should not have full transparency
   - *I want to control which users can see which parts of my data.*
4. Group Consensus should be replaced by Participant Consensus
   - *I don't need the entire network to validate transactions, I can simply have the participants validate their own transactions*

Hyperledger Fabric Key Concepts

- Problem: Ledger should not be shared with everyone
  - *I don't want to share all of my data with every participant.*
- Solution: Channels
  - Channels offer a way to create multiple ledgers, each of which can have a unique set of participants and permissions.

An example:

- Alice manufactures consumer electronic devices.
  - Alice's products are sold in many retail stores.
- Bob owns a chain of retail electronic stores in Europe and North America.
  - Bob's stores sell many electronic devices.
- Alice does business with many stores, Bob is only one. Bob does business with many manufacturers, Alice is only one.
  - Bob has negotiated a special low price with Alice.
  - Alice and Bob wish to keep their negotiated price confidential.
- If there was only one ledger…
  - Every retail store would know the price Alice was charging every other retail store
  - Every manufacturer would know the price Bob was paying every other manufacturer for their products

An example:

- What if?
  - Alice could have many ledgers, and only one of them was shared with Bob?
  - Bob could have many ledgers, and only one was shared with Alice?
- Alice and Bob can share a channel together
  - A channel provides:
    - A ledger
    - A collection of Smart Contracts
    - A set of permissions

An example:

- In order for Alice's products to get to Bob's stores, we need some help…
  - **Charlie** runs the shipping company that ships products all around the world.
    - Charlie only needs to know what's being shipped and where it's going.
  - **Diane** is the banker that financed Bob's purchase of Alice's products.
    - Diane only needs to know the price Bob paid to Alice as well as Bob's sales figures.

An example:

- In order for Alice's products to get to Bob's stores, we need some help…
  - **Evelyn** works for the customs agency in the receiving market.
    - Evelyn only needs to know what's in those shipping containers.
  - **Frank** runs an advertising agency that helps Bob sell more items.
    - Frank only needs to know when the shipment has arrived in the local market and is headed for the retail stores

An example:

- Channel 1 – Alice and Bob
  - Contains all the details of Alice's and Bob's transactions together.
  - This channel feeds data to channels 2-5
- Channel 2 – Charlie
  - Contains only information about shipments leaving Alice's factory for Bob's stores.
  - What is being shipped?
  - Where is it going?
- Channel 3 – Diane
  - Contains Bob's purchase info from Alice
  - Contains Bob's store sales records
- Channel 4 – Evelyn
  - Contains Bills of Lading and Shipping Manifests
- Channel 5 – Frank
  - Contains arrived status (yes/no)

Problem 1

- Problem: Users should not be anonymous
  - *I want to know who my users are. Anonymity does not benefit me.*
- Solution: Membership Service Provider
  - A pluggable component to a Hyperledger Fabric Network
  - One per organization
  - Contains list of all known human and system identities
  - Gives all participants on a Fabric network an identity
  - LDAP, Active Directory, oAuth are common

Problem 2

- Problem: Users should not have full transparency
  - *I want to control which users can see which parts of my data.*
- Solution: Channel Permissions and ACLs
  - ACL == Access Control Listing
  - Permissions made possible via identity (MSP)
  - Permissions can be applied at the channel level

Problem 3

- Problem: Group Consensus should be replaced by Participant Consensus
  - *I don't need the entire network to validate transactions, I can simply have the participants validate their own transactions.*
- Solution: Endorsement Policies
  - Once identity is known, group consensus can be replaced by participant consensus.
  - If you and I have a transactions, and we both agree on the outcome of that transaction why do we need anybody to help validate it?

Functionalities

- Each blockchain node performs three functions:
  1. Keep a redundant copy of the ledger (or ledgers)
  2. Execute any requested Smart Contract code
  3. Keep all copies of the ledger in-sync
     - Same data
     - Same order!
- What if each function has its own node type?
  1. *Committing Node* - Keep a redundant copy of the ledger (or ledgers)
  2. *Endorsing Node* - Execute any requested Smart Contract code
  3. *Ordering Node* - Keep all copies of the ledger in-sync
     - Same data, in the same order!

Transaction Flow

1. An end user initiates a transaction.
2. The network verifies the identity of the initiator using the appropriate Membership Service Provider.
    - Are you a valid user?
3. The network verifies the identity of the initiator using public/private key cryptography.
    - Are you who you claim to be?
4. The network verifies the user has permissions to perform the transaction.
5. The transaction is broadcast to all Endorsing nodes on the channel.
6. Each Endorsing node executes Smart Contract code and returns their result to the client application.
7. The client application checks to if consensus was reached by examining the returned results.
8. The client application informs the Ordering Nodes that a new Transaction is to be recorded on the ledger.
9. The Committing nodes (and the Endorsing nodes) record the transaction on their copy of the ledger.

References

- https://en.bitcoin.it/wiki/Main_Page
- https://www.blockchain4innovation.it
- https://www.blockchain.com/it/btc/unconfirmed-transactions

A Reputation Framework to Share Resources into IoT-based Environments

Introduction

- A large amount of data could potentially be exchanged between IoT devices and this exposes them to significant security and privacy risks both for themselves and their users
- Solving these security and privacy issues in the IoT environment is not a trivial challenge, especially when there are a large number of interconnected objects.
- However, a solution should ensure:
  - Autonomy in recognizing threats
  - Activate protections without penalizing system efficiency

Introduction

- Open and dynamic environments encourage anomalous behavior and introduce significant risk to IoT devices in providing/accessing resources, especially if they are not free.
- Identification-only approaches may be ineffective in creating an effective "trustworthiness atmosphere" where IoT devices can have reasonable confidence in their counterparts.
- Even in IoT scenarios a solution worthy of attention is represented by trust- and reputation-based systems but, obviously, the specificity of the IoT context requires the development of new trust and reputation mechanisms.

Proposal

- The proposal consists of a distributed Reputation Framework (RF).
- Each IoT device is provided with a Trust and Reputation Layer (TRL) where a tamper-proof Reputation Agent (RA) operates.
- The reputation of IoT devices is propagated in the RF when RAs mutually interact. For this purpose it is assumed that one or more secure communication channels exist.
- A specially designed reputation model implements some countermeasures to detect malicious and cheating IoT devices

The Reputation Framework

- The reputation framework consists of:
  - Reputation Agents (RA), tamper-proof components hosted by each IoT device and operating at the Trust and Reputation Layer (TRL) of the device
  - Echo Agents (EA), passive elements that propagate reputation information
  - Framework Agent (FA), manages the RF and supports RAs and EAs
- There are multiple RFs, each referring to an open environment where multiple autonomous IoT devices operate, free to enter or leave at any time
- Each component operating in an RF is able to encrypt any sensitive information exchanged between IoT devices for the security and privacy of data and communications within the RF.
- We will refer to IoT devices that provide resources as producers (p), and IoT devices that require resources and services as consumers (c). Each IoT device could play the two roles (i.e., prosumers) depending on the convenience

The Reputation Framework

- The RA manages all instances regarding the management of reputation information and the following activities:
  - Affiliation: Registration and authentication phase performed by an RA with the FA of each domain where it wishes to operate;
  - Hosted: Activities of interaction with other RAs and consisting of:
    - Meta-data synchronization
    - Submission and authentication
    - Resources request
    - Offering of resources
    - Resource provisioning

The Reputation Model

- The reputation model is associated with the following definition of reputation ``an expectation about the user's behavior based on information about the observations of her/his past behavior''
- Based on this, the reputation model considers the entire history of the IoT device (i.e., the behavior, in terms of the feedback the IoT device has received)
- It is assumed that, after the provision of a resource has been performed, the producer's and consumer's RAs calculate and exchange their feedback (which may differ between the producer's and consumer's IoT devices) to update their respective reputation scores

The Reputation Model

- Operationally, a producer and a consumer manage reputation data in the same way.
- Therefore, both ai and aj are the RAs associated with the IoT devices of i and j, manage feedback and reputation data in a secure manner
- Now, let Ø_(i,j)^D∈[0;1]⊂R be the feedback given by a_j to user i for resource D (where o represents the minimum appreciation for D, while 1 identifies the maximum appreciation)
- This feedback is sent by aj to a_j to compute the new reputation value of i (R_i^{new})

The Reputation Model

$$R_i^{new} = \begin{cases} \varphi\, \phi_{j,i}^{*D} + (1-\varphi) R_i^{old} & \phi_{j,i}^{*D} > 0 \bigvee R_i^{old} \geq 0{,}5 \\ R_i^{old} & \text{otherwise} \end{cases}$$

where

$$\phi_{j,i}^{*} = \frac{1}{2}\left(\sigma^{D} + \eta_{j,i}\right)\phi_{j,i}^{D}\, \xi_j$$

- $\sigma_{j,i}^{D}$ it depends on the cost of the resource
- $\eta_{j,i}$ it depends on the interactions passed between i and j
- ξ_j it depends on the ability of j to give correct feedback (in the domain $[0;1] \subset R$) executed autonomously by its RA that is tamper-proof

The Reputation Model

$$R_i^{new} = \begin{cases} \varphi\, \phi_{j,i}^{*D} + (1-\varphi) R_i^{old} & \phi_{j,i}^{*D} > 0 \bigvee R_i^{old} \geq 0{,}5 \\ R_i^{old} & \text{otherwise} \end{cases}$$

where

$$\phi_{j,i}^{*} = \frac{1}{2}\left(\sigma^{D} + \eta_{j,i}\right)\phi_{j,i}^{D}\, \xi_j$$

- $\sigma_{j,i}^{D}$ it depends on the cost of the resource
- $\eta_{j,i}$ it depends on the interactions passed between i and j
- ξ_j it depends on the ability of j to give correct feedback (in the domain $[0;1] \subset R$) executed autonomously by its RA that is tamper-proof

Computation of σ

$$\sigma^{D} = \begin{cases} 1 & C = 0 \\ \dfrac{C(D)}{C_{max}(D)} & otherwise \end{cases}$$

Computation of η

$$\eta_{j,i}(t) = \begin{cases} 1 & \phi_{j,i}^{D} < 0 \\ \dfrac{1}{\theta_{j,i}(t)} & \phi_{j,i}^{D} \geq 0 \end{cases}$$

where $\theta_{j,i}(t)$ is given by:

$$\theta_{j,i}(t) = \begin{cases} 1 & t = 0 \\ \theta(t_l) + 1 & t - t_l < \Delta_t \\ \max\left(1, \theta(t_l) - \left[\dfrac{t - t_l}{\Delta_t}\right]\right) & t - t_l \geq \Delta_t \end{cases}$$

The Reputation Model

- $\sigma_{j,i}^{D}$ it depends on the cost of the resource
- $\eta_{j,i}$ it depends on the interactions passed between i and j
- ξ_j it depends on the ability of j to give correct feedback (in the domain $[0;1] \subset R$) executed autonomously by its RA that is tamper-proof

The Reputation Model

- To overcome the problem of centralized reputation systems, in this case the reputation score is self-stored by RAs that are assumed to be tamper-proof components

- When an RA detects a communication breakdown due to a failure or deliberate intervention to avoid receiving negative feedback, then the RA decreases the reputation value of its device as $R^{new} = \tau\, R^{old}$ where:

  - τ is randomly chosen in $\vdash]0.5;1] \in R$ with probability 0.5 for communications outages due to failure
  - τ "is randomly chosen in " $[0,0;5[\in R$ for voluntary communication outages

The Experiment

- The proposed reputation model was tested by simulating an urban mobility scenario on a small transportation network (our RF).

- Specifically, two scenarios were simulated, each consisting of a single FA, 1000 RAs associated with IoT devices (i.e., vehicles), and 4 EAs (i.e., associated with traffic lights) located on the transportation network
  - Scenario A includes some unreliable RAs that release random feedback tending to 0 at "reliable Ras" and tending to 1 at "unreliable Ras" and "communication failure" with probability pc
  - The scenario B comprises RA " unreliable " that they act in order to obtain one reputation ``positive'' on low cost services in order then to spend it in order to cheat on services to high cost (considered in the measure of 25% of the total)
-

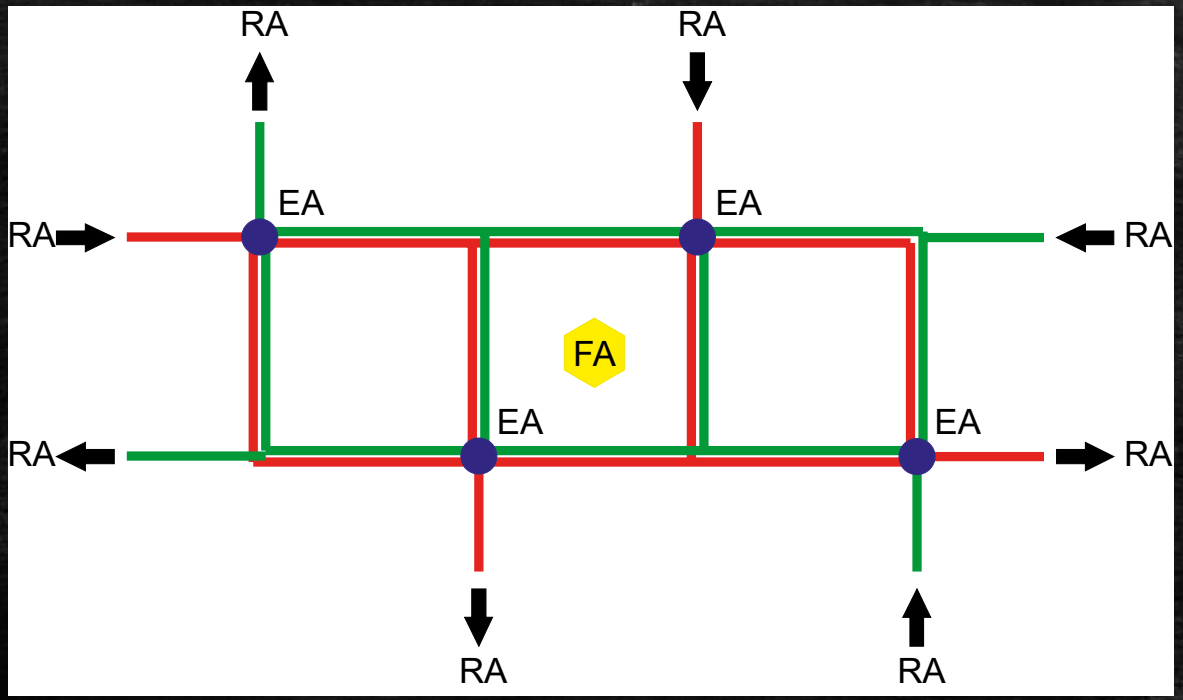
The Experiment

- Each experiment is formed by 50 epochs, for each epoch 25% of RAs, randomly chosen, are active on the network
- The initial reputation was set equal to 0.5, which is also the threshold adopted to separate reliable RA from unreliable ones
- RAs having an unreliable behavior get a feedback <0.5 with probability pf (see Table in the next slide)
- The speed of the vehicles was randomly chosen in the range [25,50] Km/h
- Finally, IoT interactions are assumed to be realized when vehicles are stopped at traffic lights without it conditioning the results

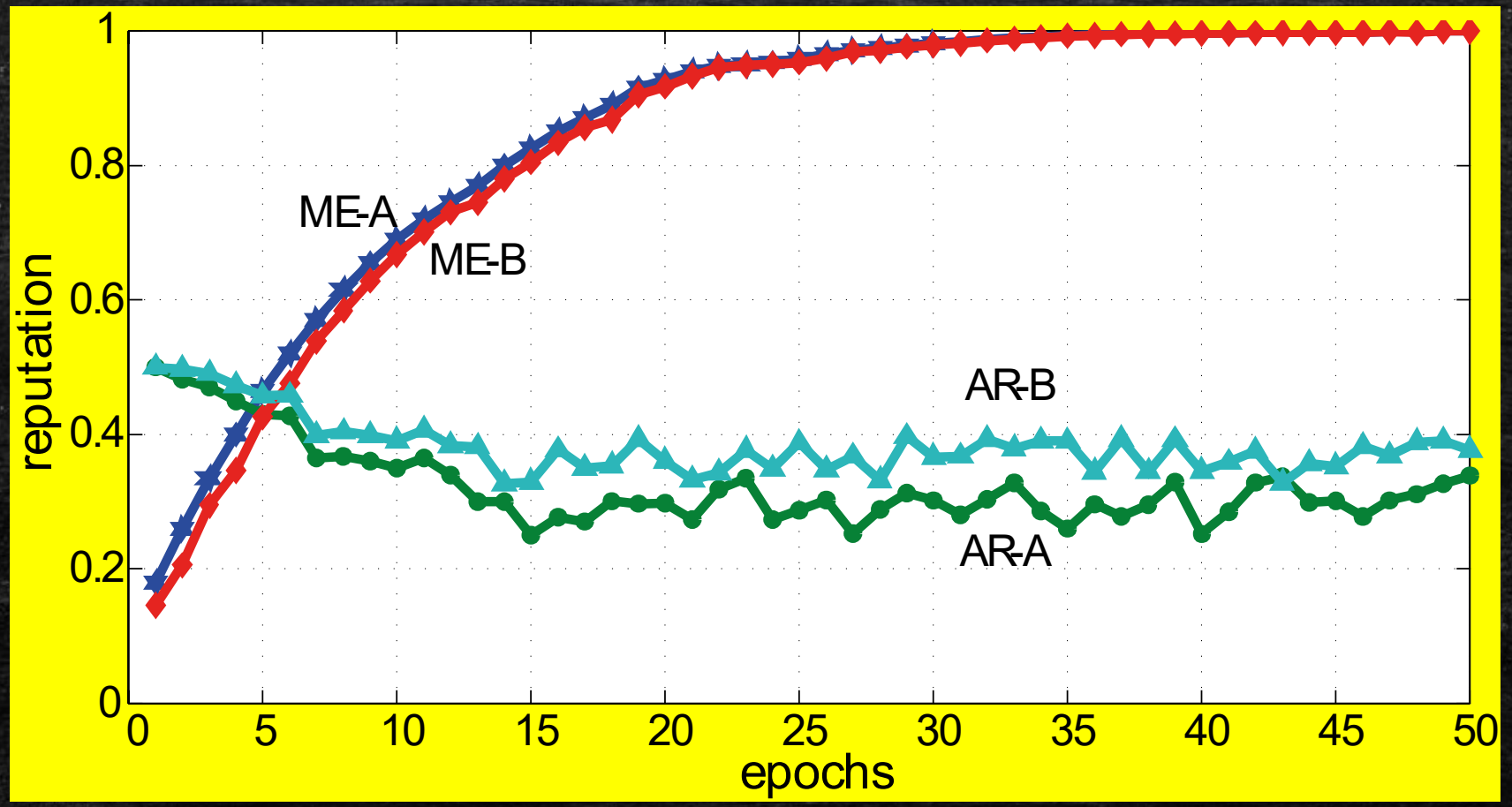
The Experiment

Scenario	Untrustworthiness IoT devices	Unreliable IoT devices Behavior
A	10%	Low performance. Unreliable IoT devices will cause a communication interruption with probability pf
B	10%	Building positive reputations on low cost services for cheating on high cost services. Absence of interrupted communications

$\Delta t = 5, \quad \eta = 0.5, \quad \alpha = 0.5, \quad p_c = 0.5, \quad p_f = 0.2, \quad c_g/c_v = 1 \quad (i.e., 60/60 sec.)$



The Experiment



Conclusions

- We examined a reputation-based framework to support IoT devices in sharing resources in "smart environments".
- To this end, a distributed reputation model has been proposed, capable of rendering ineffective, or at least minimizing, the effects of both collusive and fraudulent activities.
- Some experiments referred to a simulated scenario of mobility have attested the effectiveness of the proposed approach

- A Reputation Capital and Blockchain-based Model to Support Group Formation Processes in the Internet of Things

Introduction

- The effectiveness of a group (real or virtual) is related to the number of interactions that occur between the members of that group or, in other words, by its social capital
- In this scenario, an interesting problem is represented by "trustworthiness" in a network composed of multiple federated domains and to which a large number of agents are affiliated
- In such a scenario, a common way for an agent to select a partner is based on the (global) reputation that a device (i.e., agent) has in its community since its personal experience is not suitable to directly make a good choice
- However, in a distributed context, making a global reputation measurement is not trivial given the absence of a centralized repository

Introduction

- I device IoT sono equipaggiati con un crescente numero di sensori e con crescenti capacità computazionali per realizzare ambienti pervasivi, contestuali e smart
- La cooperazione tra device IoT e un potente mezzo per incrementare le loro prestazioni
- In un ampia rete di distribuiti e federati domini IoT, è possibile supportare la cooperazione tra smart device IoT associando un agente software ad ognuno di essi
- Ogni device può muoversi tra i differenti domini che formano la rete, dove egli dovrà selezionare alcuni partners affidabili anche se avrà un'esperienza insufficiente per una buona scelta
- Una soluzione promettente per questo problema è rappresentata dalla formazione di strutture sociali, p.es. gruppi di agenti, per supportare idoneamente i device in ognuno degli ambienti federati

Proposal

- To this end, we model an agent's reputation by:
  - A reputation measure called Reputation Capital (RC).
  - A blockchain protocol to maintain and certify each agent's Reputation Capital across federated domains
- In addition, we implement a competitive IoT scenario where:
  - Groups of agents can be formed within each federated domain
  - Cooperation for services is offered for free within the same group (otherwise it is provided only for a fee).
- By combining IoT, reputation systems, blockchain and group formation, IoT devices moving across federated domains of the IoT network can certify their Reputation Capital to be active groups in their current domain.
- A simulation was conducted considering honest and dishonest agents. The results obtained showed that almost all rogue agents were identified and that honest agents pay significantly less than rogue agents

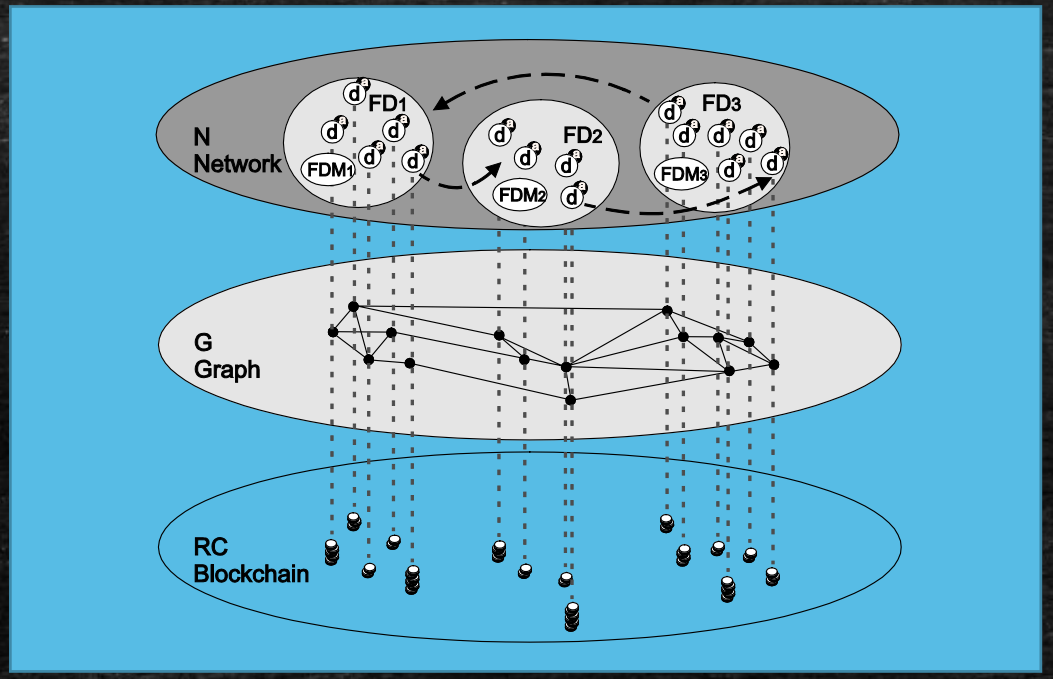
IoT Framework

- Let N be the IoT Network formed by n (with n > 1) Federated Domains (FD)s and populated by a large number of heterogeneous and smart IoT devices
- Assuming that:
  - a blockchain is associated with N
  - SD is the set of devices
  - SA is the set of software agents, each associated with a unique personal IoT device
- Each federated domain FD ∈ N is administered by a trusted and equipped agent called Federated Domain Manager (FDM) that provides all devices, temporarily affiliated with its administered domain, with some basic services.
- In particular, an FDM provides to:
  - Associate an identifier (Id), unique in N, to each agent the first time it is active in N
  - Manage and update a registry of all agents currently hosted in its administered FD

IoT Framework

- The set of SA agents affiliated with N and the relationships between them will be represented by a graph G = <NG;LG>, where:
  - NG represents the set of nodes of G and each node of NG is associated with a unique agent $a \in SA$ (i.e., device $d \in SD$)
  - LG represents the set of oriented arcs where each arc of LG is associated with a relation taking place between two agents of N (middle layer in Figure)
- Groups can be formed within each FD and each agent can request affiliation to one or more active groups in its current FD based on its Reputation Capital certified by the blockchain (note that agents can freely move from one domain to another)
- Let $g_k^{FD_m}$ be the k-th group formed within the m-th domain $FD_m \in N$ and, similarly to each active group in FD, it will be managed by the respective FDM

IoT Framework



Reputation Capital

- In N each IoT device is a prosumer that consumes and produces services (offered for free only to agents belonging to its own group of producers)
- When an IoT device operates as:
  - Consumer, the blockchain certifies its reliability
  - Provider, its Reputation Capital score certifies the quality of its services
- The Reputation Capital (RC) is represented by a numerical score that takes into account the past "behaviors" of the device (i.e., agent) through its past **qualified interactions** with other devices belonging to N when it acted as a provider.
- The first time an agent is active in N, it receives an initial RC that should not penalize the new entrant too much but should discourage whitewashing strategies of malicious agents that in the presence of a compromised RC could leave the framework and then return to it to receive a new and more advantageous initial RC

Feedback and Interaction Relevance

- We define interaction as when a consumer agent a_j obtains a service s from a provider agent a_i
- After s is consumed, a_j issues a feedback $\phi_{ji} \in [0; 1] \in R$ representing its appreciation for s and if the interaction is qualified (see later) this feedback will be used to update the RC_i of the agent a_i
- The Relevance *R* of an interaction is calculated as:

$$R = \begin{cases} \frac{c_s}{C} & se \;\; c_s < C \\ 1 & otherwise \end{cases}$$

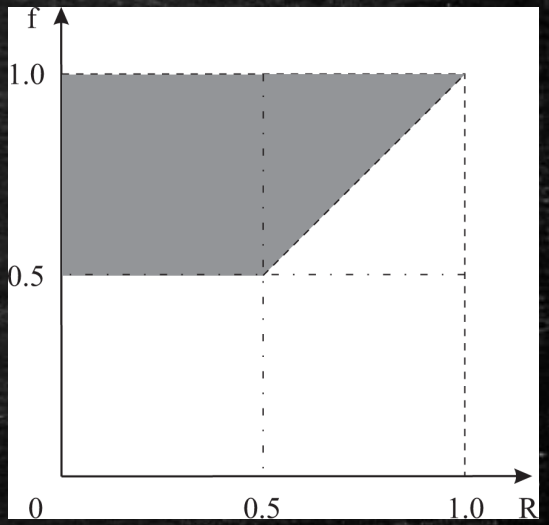
where:

- c_s is the cost of s
- *C* is the maximum cost, for higher costs it is assumed that the relevance is "saturated"

Feedback and Interaction Relevance

- We say that an interaction is qualified when the feedback ∅, with respect to R, is:

$$\begin{cases} f \leq 0{,}5 \quad \forall\, R \\ R \geq f \quad per \quad f > 0{,}5 \;\; e \;\; R \geq 0{,}5 \end{cases}$$



- Based on the last h qualified interactions of provider agent a_i occurred with different agents, its RC_i will be updated as follows:

$$RC_i = \sum_{n=1}^{h} \frac{\omega_n \delta_{j,n} \emptyset_{ji,n}}{R_{ji,n}}$$

where:

- ω weights the h qualified interactions giving more importance to the most recent among them
- δ mitigates the impact of "complaining" agents giving negative feedback (e.g., < 0:5) systematically, δ is calculated as:

$$\delta = 1 - \frac{number\ of\ negative\ feedback}{number\ of\ feedback}$$

Reputation Capital

- When a service is provided, a smart contract is initiated on the blockchain platform (e.g., Ethereum) to realize all contractual obligations, including payments (e.g., using Ether).
- To realize a competitive framework, groups are formed assuming that interactions taking place between members of the same group are free of charge.
- A group formation algorithm is run periodically in each federated domain (FD) by its manager (FDM) to group agents (devices) based on their RC scores
- For each FD, its FDM decides the number of groups allowed and the RC score (increasing between groups) required by each group to join an agent
- Each agent can apply to his FDM for affiliation with a group active in the FD, the FDM will assign this agent to the best possible group based on RC score
- Periodically, the FDM checks if the RC of each agent is still adequate for his group, otherwise the agent will be moved to another group based on his RC score or removed from each group if his RC score is too low for each active FD group.

Algorithm executed by each FDM

This pseudocode is executed by the FDM of each FD

(symbols are listed in the table)

Table of the main symbols

Symbol	Description	Symbol	Description
N	Network	τ_m	Time Threshold
FD	Federated Domain	FDM	Federated Domain Manager agent
DG	Dataset of Groups	DA	Dataset of Agents
MG	Max. number of Groups	AG	Set of Active Groups

- Input: DA_m; DG_m; τ_m; MG_m;
- 1: for all $a_i \in FD_m$ do
- 2: if (*time_last RC_check* $\geq \tau_m$) then
- 3: ask to the blockchain for the updated *RC* of a_i and then update DA_m
- 4: end if
- 5: end for
- 6: for *all groups* $\in AG_m$ do
- 7: if (*time_last_affiliation_check* $\geq \tau_m$) then
- 8: for all $a_i \in g_k$ do
- 9: if ($(RC_i < \Gamma_k)$ || $(RC_i \geq \Gamma_{k+1})$) then
- 10: *Assign* (a_i; AG_m; DA_m; DG_m)
- 11: end if
- 12: end for
- 13: end if
- 14: end for
- 15: for all ($a_i \in FD_m$) requiring to be affiliated with a group do
- 16: if $RC_i \geq \Gamma_1$ then *Assign* (a_i; AG_m; DA_m; DG_m)
- 17: else reject the request of a_i and sends to the agent a message
- 18: end if
- 19: end for

Function Assign ( )

- Assign( ) has as input:
  - an agent
  - The maximum number of groups in FD_m
  - the dataset DA_m
  - the dataset DG_m
- For each agent in FD_m, Assign( ) checks their RC with the affiliation threshold Γ
- if $RC_i < \Gamma_1$ of the last group (with the smallest Γ in AG_m) then the Remove( ) function is called to remove a_i from each group in AG_m until its RC_i is adjusted to a new group affiliation
- Otherwise a_i to the group that best fits his RC_i

- Input: a_i; DA_m; DG_m; $MaxG_m$;
- 1: for all $a_i \in FD_m$ do
- 2: if ($RC_i < \Gamma_1$) then *Remove* (a_i; DA_m; DG_m)
- 3: else
- 4: for all $g_k \in G_m$ do
- 5: if ($RC_i \geq \Gamma_{k+1}$) then
- 6: assign a_i to the group g_k
- 6: end if
- 7: end for
- 8: end if
- 9: end for

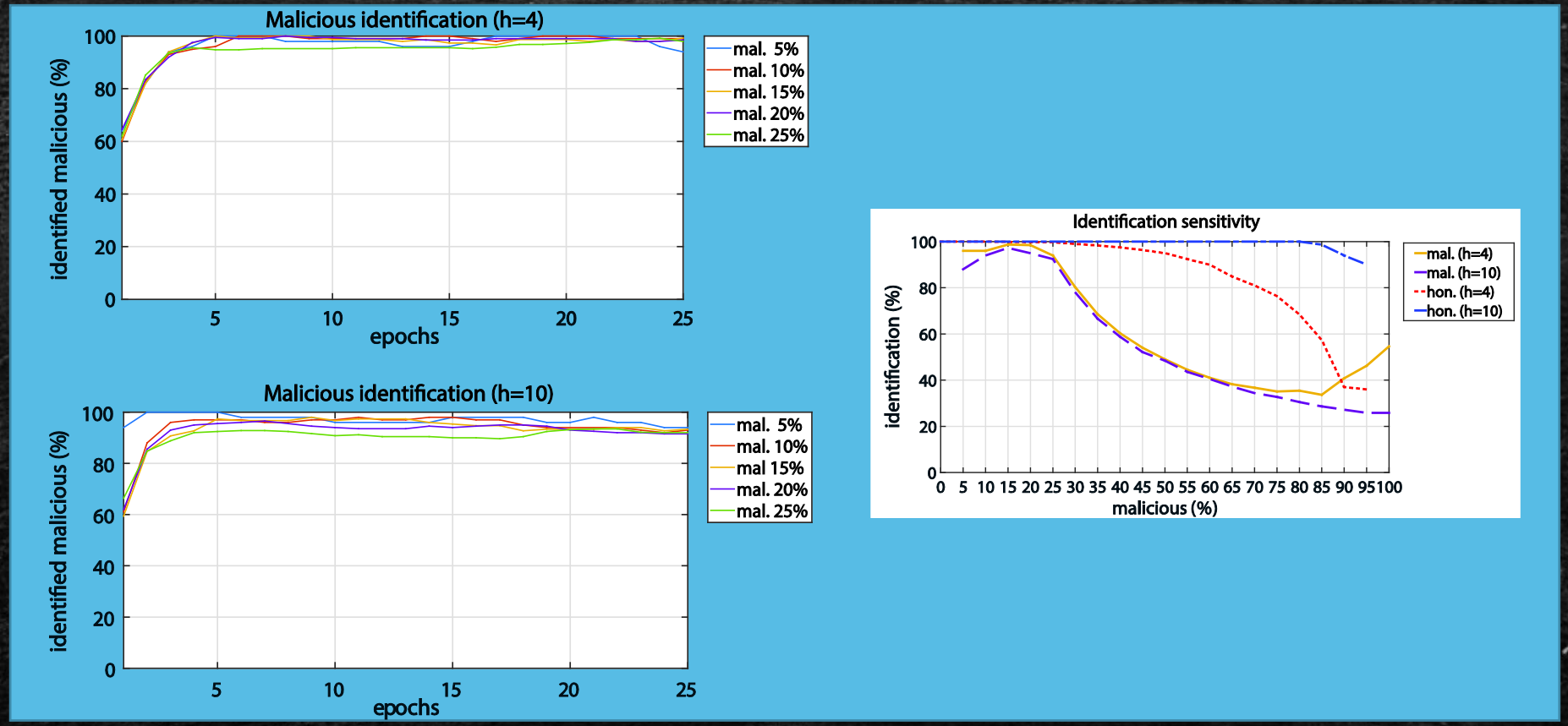
Experiments

- A number of simulations were conducted to verify the proposed framework by modifying both the horizon (h) and the percentage of malicious actors
- In particular, the simulations were aimed at verifying:
  - The ability in identifying malicious actors in the presence of different and concurrent types of attacks carried also with different strategies
  - The distribution of devices (agents operating as prosumers) among the groups
  - The growth of the RC with the number of executed interactions
  - The costs sustained by the devices to buy services from the providers
- Note that the RC of a device is related only to its activity as a provider because when it acts as a consumer it is trusted by the blockchain
- The simulations involved only one federated domain

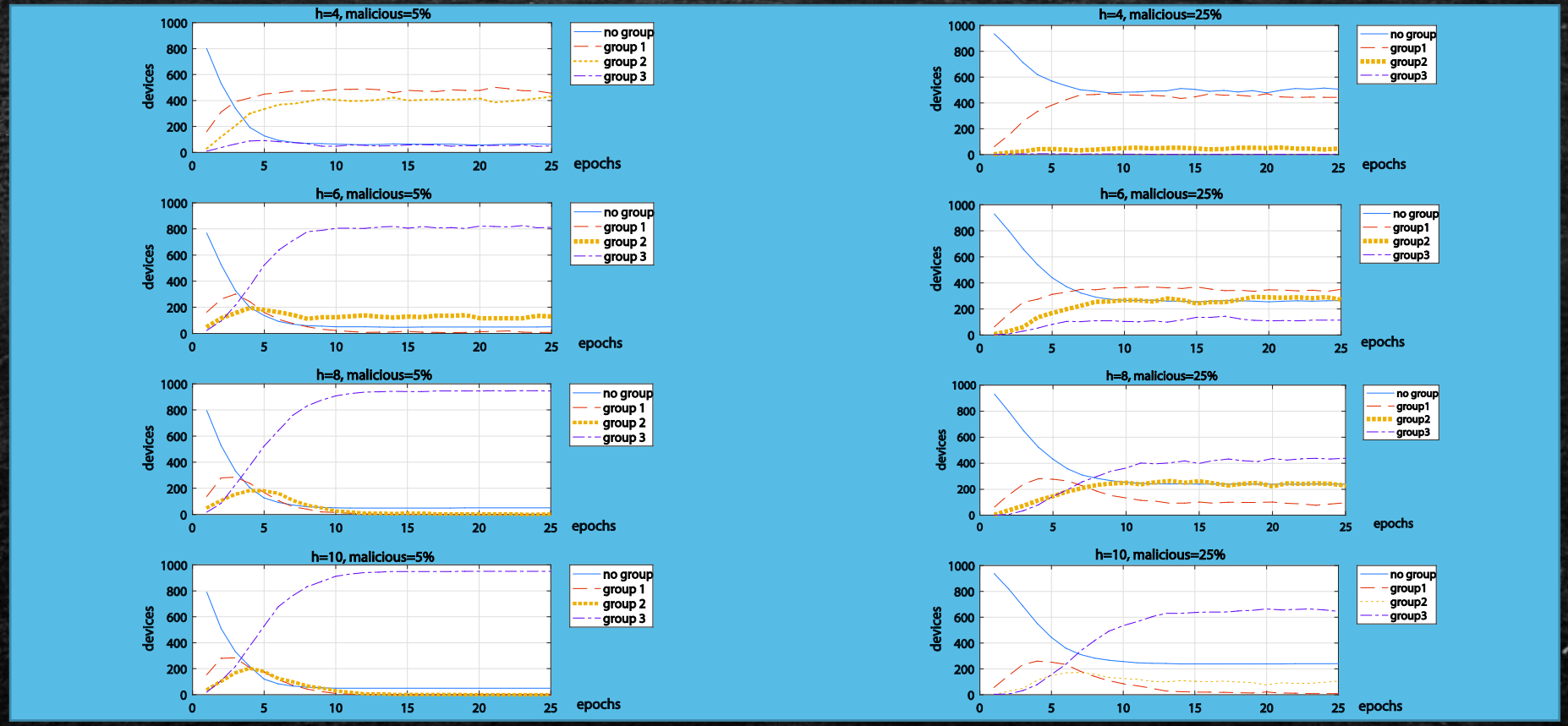
Paramater Setting

- The adopted setting of the main parameters of the simulations was:
  - A population of 1000 IoT devices/agents
  - Interactions organized by epochs, each consisting of 1000 interactions
  - Simulations 1000 epochs long (N.B., the results are stable already in about 10 epochs)
  - The initial RC score assigned to each device was 1.0
  - The cost of a service s has been assigned randomly in the domain 0.01÷1.5$ with a threshold value P = 1$
  - The horizon (h) adopted for the simulations has been made to vary from h = 4 to h = 10 with step 2
  - Based on the different strategies adopted by honest and malicious devices, the frequency of the qualified interactions has varied with ratios of 1:h, 1:h/2, 1:1 and in random way
  - The percentage of cheaters has varied from 5% to 25% of the population with 5% steps
  - Four types of malicious behavior were considered:
    - alternate
    - collusive
    - malicious
    - disturbing
  - The number of groups was set at 3 with RC thresholds for affiliation equal to 2.5, 4.5, and 6.0

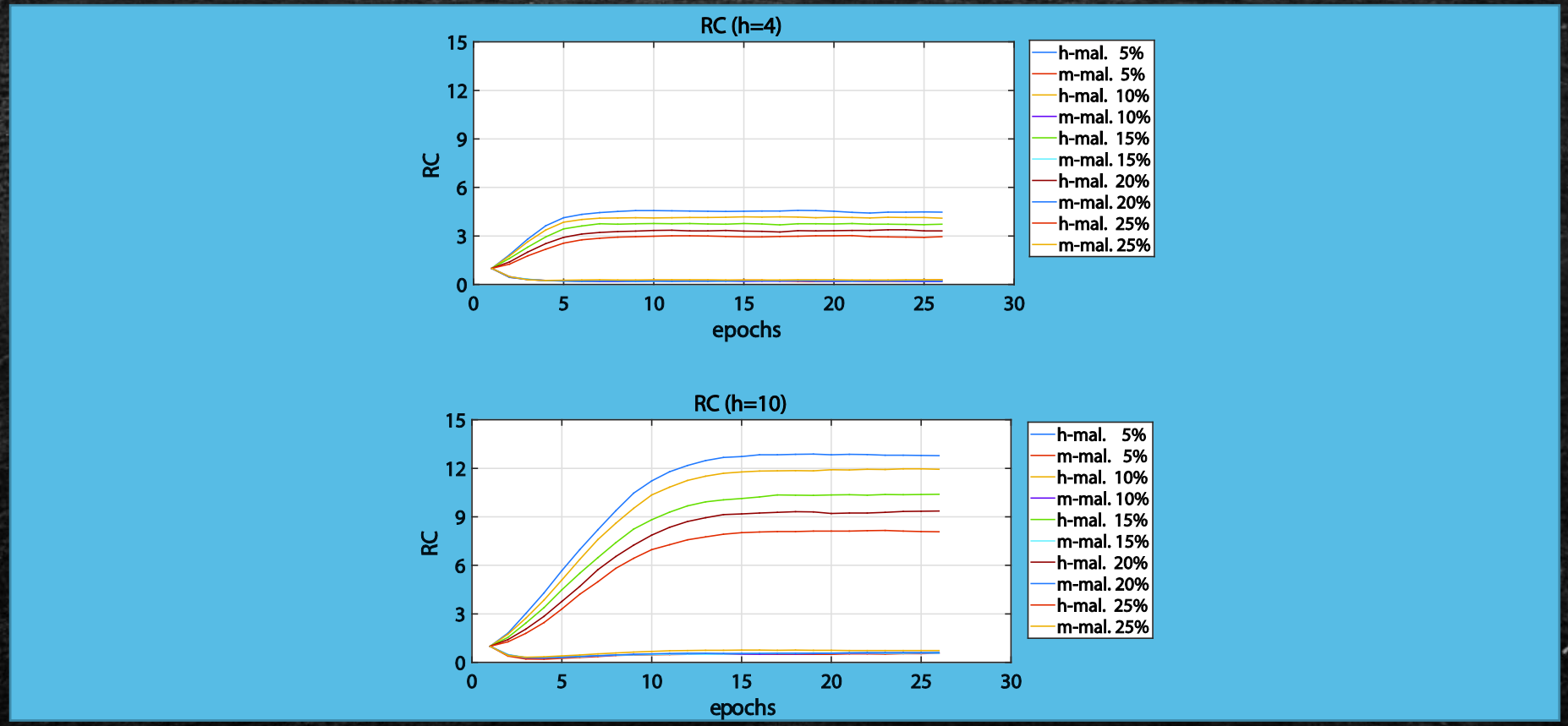
Malicious Identification



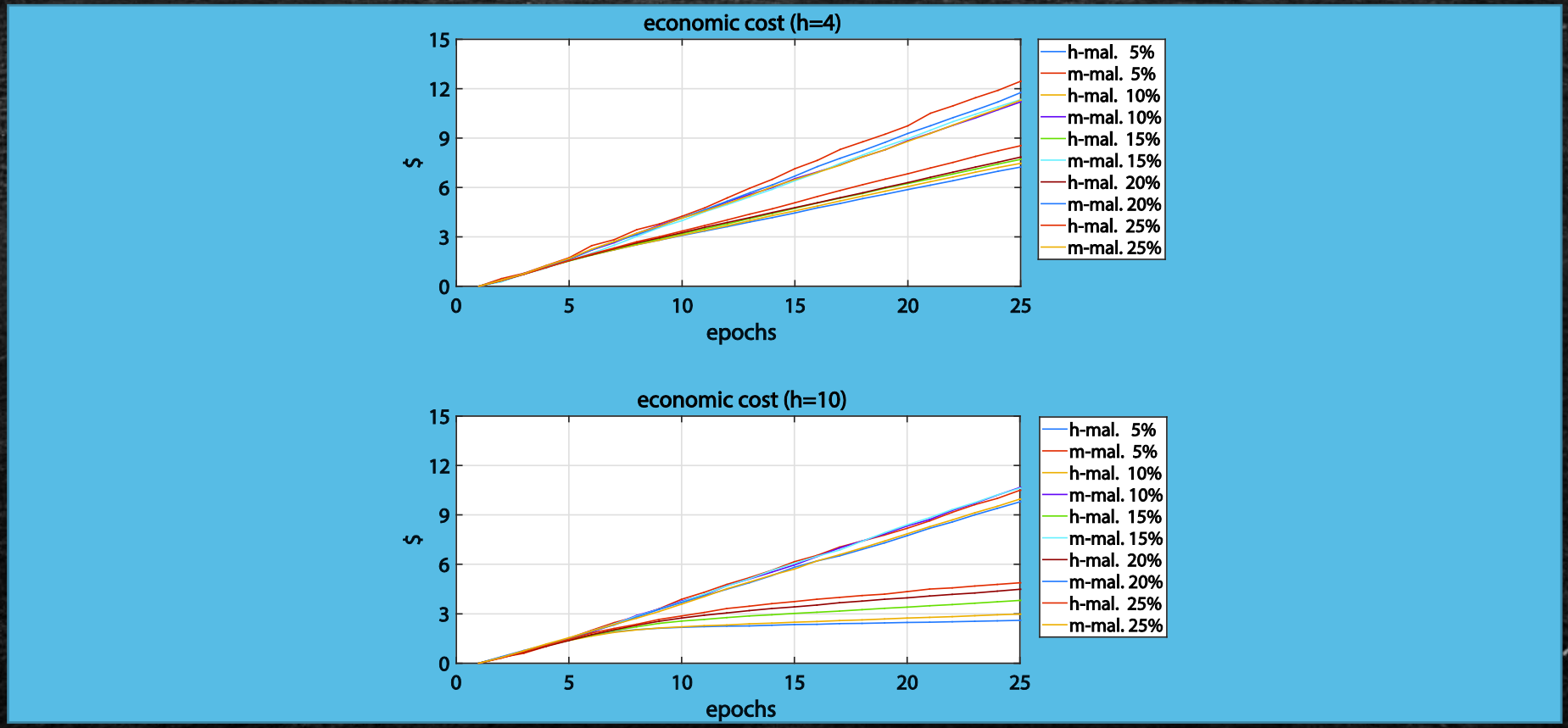
Group Affiliation



Reputation Capital



Economic Cost



Conclusions

- We considered a wide IoT network formed by federated domains where heterogeneous, smart and cooperative IoT devices (associated to software agents) are free to move between domains and form groups.
- To support the cooperation among devices in choosing reliable partners, the Reputation Capital model was proposed, built on the basis of the feedbacks released by devices for each interaction and disseminated by means of a blockchain (to avoid any centralized component).
- Experiments have verified that the proposed framework is resilient to malicious agents in terms of their identification, group formation, Reputation Capital and from an economic point of view
- Synergies from the Reputation Capital model, group formation algorithm, and blockchain confirmed expectations allowing the framework to operate properly

References

- G. FORTINO, F. MESSINA, D. ROSACI and G.M.L. SARNÈ (2019). *Using Blockchain in a Reputation-Based Model for Grouping Agents in the Internet of Things*. IEEE Transactions on Engineering Management (TEM), pp. 1-12, pub. IEEE, eISSN: 1558-0040, doi: 10.1109/TEM.2019.2918162

- G. FORTINO, F. MESSINA, D. ROSACI and G.M.L. SARNÈ (2018). *Using Trust and Local Reputation for Group Formation in the Cloud of Things*. Future Generation Computer Systems (FCGS), vol.89, December, pp. 804-815, pub. Elsevier, ISSN: 0167-739X, doi: 10.1016/j.future.2018.07.021

- P. DE MEO, F. MESSINA, M.N. POSTORINO, D. ROSACI e G.M.L. SARNÈ (2017). *A Reputation Framework to Share Resources into IoT-based Environments*. In Proceedings of the "14th IEEE International Conference on Networking, Sensing and Control" (ICNSC 2017), a cura G, Fortino, M. Zhou, Z. Lukszo, A.V. Vasilakos, F. Basile, C.E. Palau, A. Liotta, M.P. Fanti, A. Guerrieri, A. Vinci, pp. 513-518, Art. N.8000145, IEEE, ISBN: 978-150904428-3, Falerna (Catanzaro), Italia, 16-18 Maggio 2017. doi: 10.1109/ICNSC.2017.8000145

- M.N. POSTORINO e G.M.L. SARNÈ (2014). *An Agent-based Sensor Grid to Monitor Urban Traffic*. In Proceedings of XV Workshop from "Objects to Agents" (WOA-2014), a cura di C. Santoro e F. Bergenti, vol. 1260, pp. 1-6, CEUR Workshop Proceeding, ISSN:1613-0073, Catania, Italia, 25-26 Settembre 2014.