ARTIFICIAL INTELLIGENCE

Laboratory @ Dept. Digital Systems, University of Piraeus

Trust Semantics in IoT Entities' Deployment

KONSTANTINOS I. KOTIS AND GEORGE A. VOUROS

1st AI-IoT Workshop, SETN 2016

University of Piraeus
Department of Digital Systems

DATA STORIES
The University of Piraeus 's
Data Science Lab

(I) Problem Area

Setting

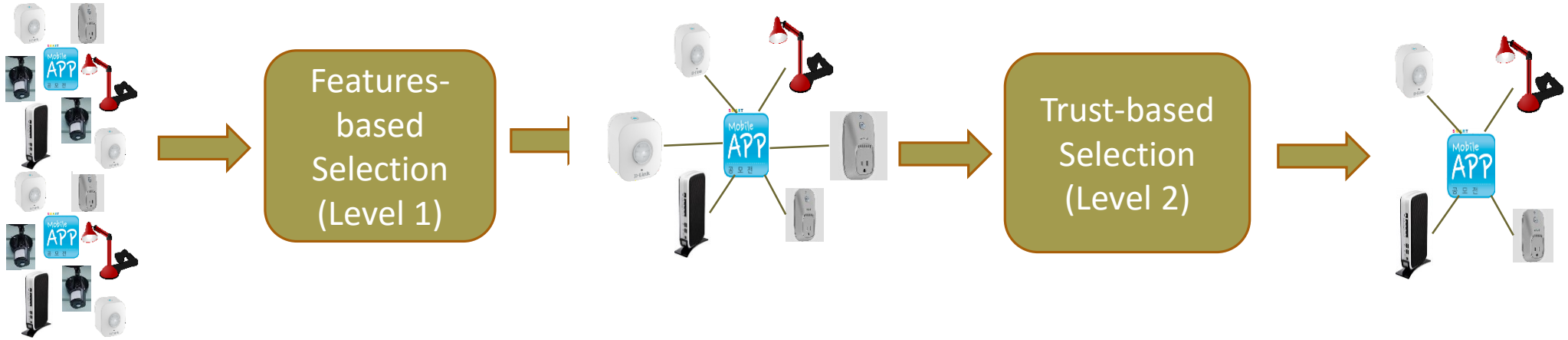
- Automated deployment of IoT entities in open and heterogeneous IoT environments
- IoT entities matchmaking: 3rd party smart app selects the suitable devices for deployment, based on i/o specifications matching
  - App functional services' descriptions (input/output) match to functionality descriptions of devices (sensor output, actuator input, app data/commands messaging)
- More than one device of the same type may be discovered in the deployment setting, suitable for selection for the deployment of the app
  - E.g. home security app matched with 2 motion detection sensors that are present in the deployment environment (both fulfill the i/o requirements of app's functional input service)
- Ontology-based IoT gateway solutions (e.g. the SSGF) provide such matchmaking functionality (features-based selection of IoT entities)

(I) Problem Area

The (selection) problem

IoT entity (app) need to

- ‘decide’ on trustworthiness between the features-based matched devices (who to trust)
- Select (from the features-based matched ones) the most trustworthy entities for its deployment



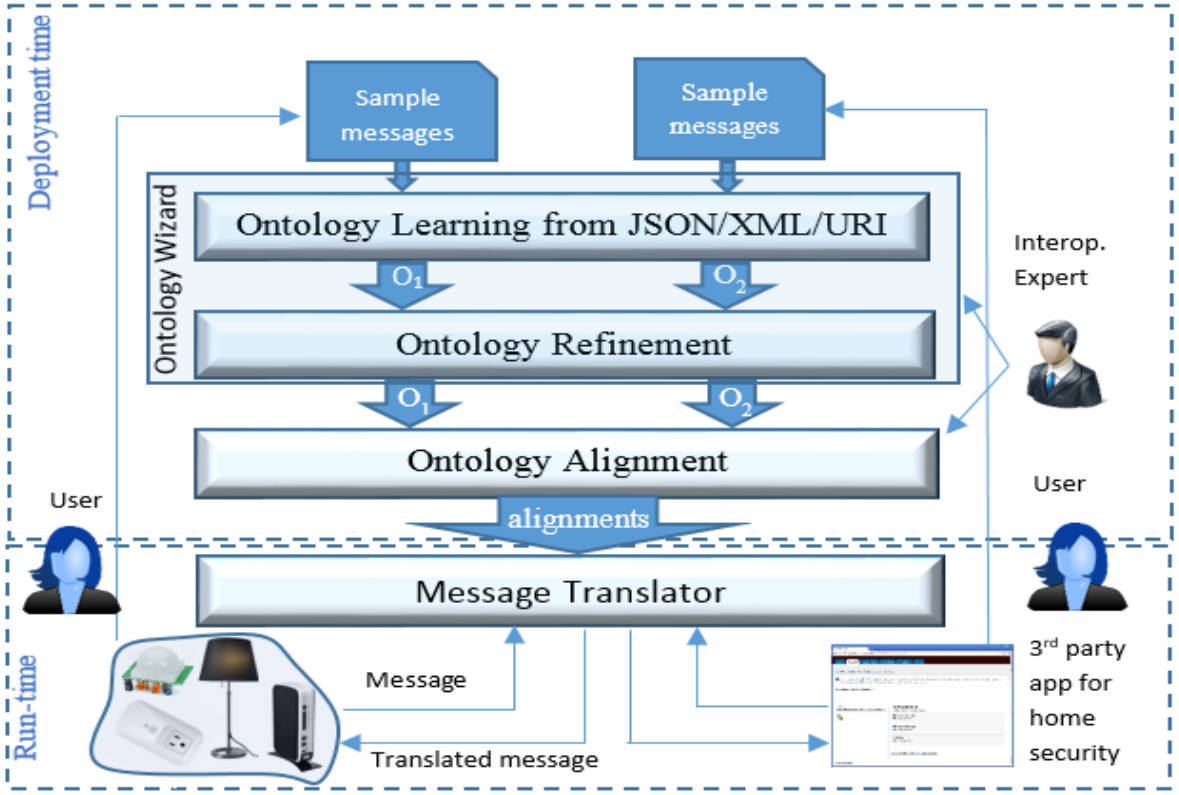
Techniques, methods, resources, tools

- IoT ontologies as a Semantic Registry for IoT entities (synthesis of upper and domain-specific ontologies)
- Tools for learning ontologies, extracting semantics from exchanged messages between IoT entities (data, commands, info) and transforming into a common syntax and semantics (e.g. OWL ontology for a home security app)
- Ontology alignment methods for the matchmaking of semantically annotated i/o specifications of IoT entities (e.g. between an home security app and a motion detection sensor)
- Commonly agreed and widely-used ontological resources (e.g. SSN ontology, DUL ontology, SWEET ontologies)
- **Trust models (e.g. O'Hara's trust model of Trustworthiness: Tw <Y, Z, R(A), C>)**
- **Trust semantics using ontological representations**
- **Fuzzy semantics using standard ontology language (OWL 2 and fuzzyOWL2)**
- **Computational models for dynamic computation of trust (Social IoT: modeling relations between owners of IoT entities)**

The SSGF

…for IoT entities that are 'foreign' to each other (different or no semantics), SSGF* provides a way to:

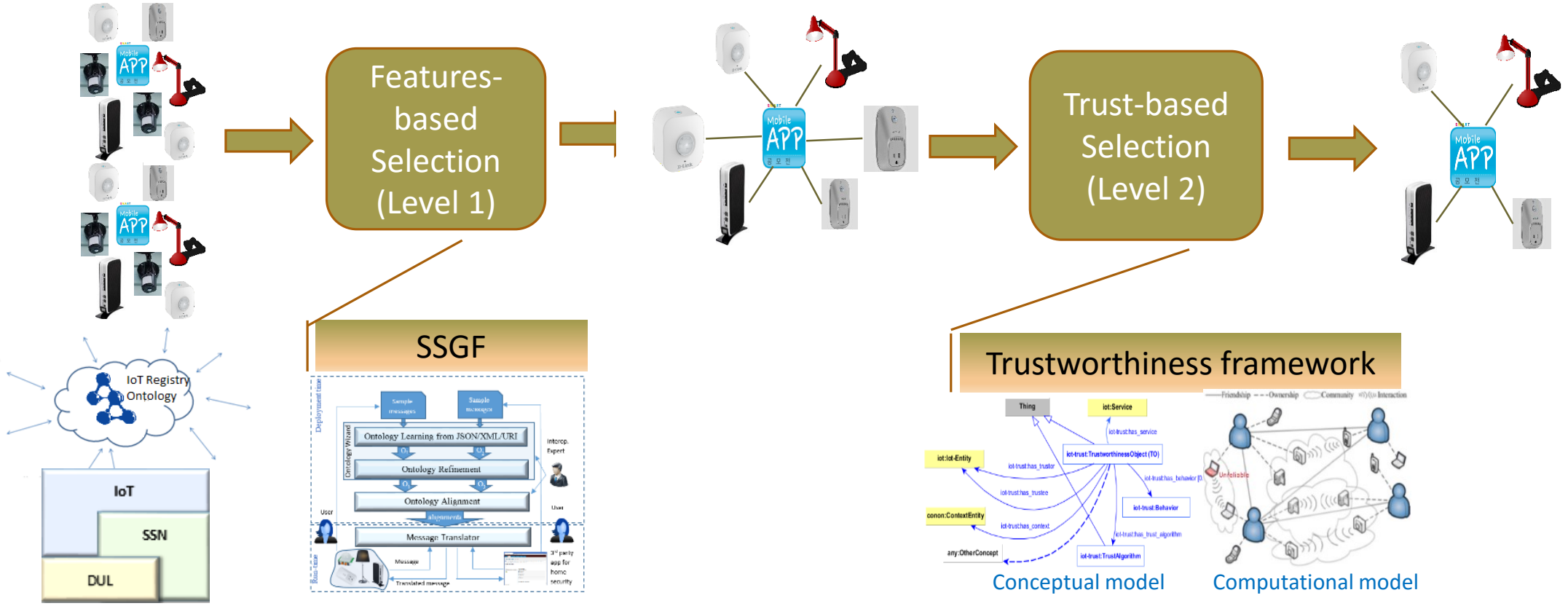
- (semi)automatically deploy IoT entities produced by different vendors
- develop 3rd party *generic* applications (general purpose) to run on different IoT devices (different vendor but same purpose)



K. Kotis, and A. Katasonov. Semantic Interoperability on the Internet of Things: The Semantic Smart Gateway Framework, International Journal of Distributed Systems and Technologies (IJDST), vol. 4, issue 3, pp. 47-69, 07/2013

(III) Use of AI technology to solve the problem

Overall Deployment Approach



(III) Use of AI technology to solve the problem

Focus of current work

- Propose a framework for trust-based selection of IoT entities (level 2) as an extension of any IoT ontologies, introducing simple and extensible semantics

Trustworthiness Framework

- **Conceptual model (representing trustworthiness of IoT entities)**
  - Reuse trust semantics of existing trust models/ontologies (e.g. based on O'Hara definition of trust)
  - Integrate fuzzy semantics reusing the framework of FuzzyOwl2, a fuzzy extension of OWL 2
- Computational model (computing trust values for IoT entities)
  - Reuse and extend a state-of-the-art well-defined computation model of Bao & Chen* on dynamic trust management for community-based social IoT environment

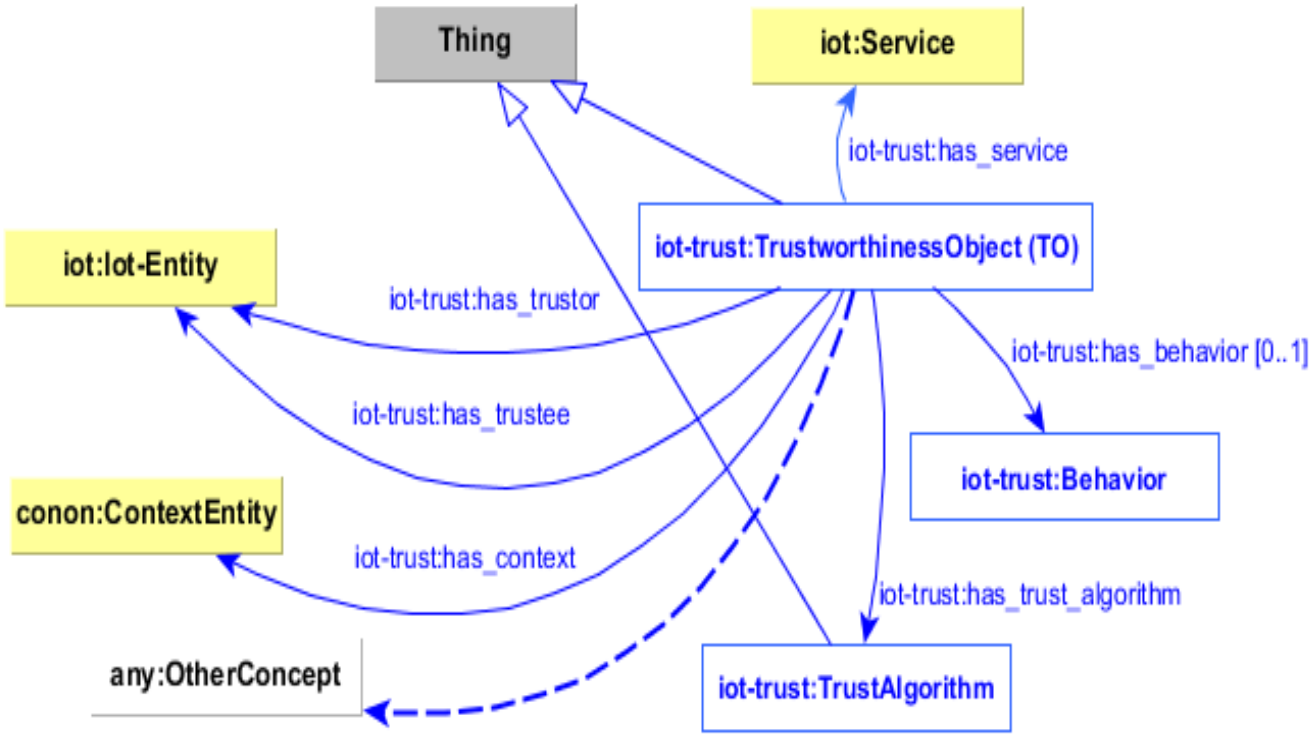
F. Bao and I. Chen. Dynamic trust management for internet of things applications. In *Proceedings of the 2012 international workshop on Self-aware internet of things* (Self-IoT '12). ACM, New York, NY, USA, 1-6. DOI=http://dx.doi.org/10.1145/2378023.237.

Conceptual model requirements

- **Reuse** existing ontologies
  - IoT-ontology, context ontologies, trust ontologies
- Easily **pluggable** in (reused by) any IoT ontology
  - introduce an ontology pattern that relates a Trustworthiness Object (TO) to any IoT-entity, assigning the role of a trustor or of a trustee
- **Extensible:** easily add new trust-related properties
  - The proposed TO class definition pattern allows an easy and straightforward addition of new properties (object or datatype)
- **Simple** (minimum required semantics)
- **Represent vague (fuzzy) information** using standard language such as OWL 2 (use OWL 2 annotation properties to encode fuzzy ontologies: FuzzyOWL2*)

F. Bobillo, U. Straccia, Fuzzy ontology representation using OWL 2, International Journal of Approximate Reasoning, 52 (7), October 2011, Pages 1073–1094.

The Trust model



φ ≥α* or *φ ≤β

φ is a fuzzy proposition and *α, β* ∈ [0, 1]

"the degree of truth of *φ* is *at least α* (resp. *at most β*)"

E.g. ***'x is a reliable temperature sensor ≥ 0.9'*** (the degree of truth of *x* being a reliable temperature sensor is at least 0.9)

R: X × Y → [0, 1]

A (binary) fuzzy relation R over two countable classical sets X and Y is a function

Under revision (ver 0.5) & development: http://ai-group.ds.unipi.gr/kotis/ontologies/IoT-trust-ontology

Querying the model

"For a ***room*** context, for a ***smart room*** **app**lication and for a ***detection service***, get the most ***reliable*** *entities* for its deployment (given a trustworthiness threshold of 0.7)

// assertions
(related TO_2 reliable has_behavior 0.7)
(related TO_1 reliable has_behavior 0.5)
//queries
(min-related? TO_1 reliable has_behavior)
(min-related? TO_2 reliable has_behavior)
//reasoner translation and answer
Is TO_1 related to reliable through has_behavior ? >= 0.5
Is TO_2 related to reliable through has_behavior ? >= 0.7

Encoded in FuzzyOWL2 using the binary fuzzy relation R 'has_behavior' over two sets: Behavior and TO

Not (?) a very useful way for representing fuzziness in order to select the most trustful instances (IoT entities)

Querying the model

"For a ***room*** context, for a ***smart room*** **app**lication and for a ***detection service***, get the most ***reliable*** *entities* for its deployment (given a trustworthiness threshold of 0.7)

```
SELECT * WHERE {
?trustObject a iot-trust:TrustworthinessObject.
?trustObject iot-trust:has_context conon:room.
?trustObject iot-trust:has_trustor iot-app:smartRoomApp.
?trustObject iot-trust:has_behavior iot-trust:reliable.
?trustObject iot-trust:has_service iot:motionDetectionService.
?trustObject iot-trust:hasTrustValue ?value.
FILTER (?value >=0.7)}
```

Encoded in SPARQL, without utilizing fuzzy semantics

Querying the model

Engineering of the ontology using fuzzy semantics can enrich the definitions in our model

e.g. W*hich **lamp** is the most trustworthy instance of the class **SmartLamp**, given its specific characteristics*.

//assertions
(instance mySmartLamp SmartLamp 0.7)
(instance herSmartLamp SmartLamp 0.3)
//queries
(min-instance? herSmartLamp SmartLamp)
(min-instance? mySmartLamp SmartLamp)
//reasoner translation and answer
Is herSmartLamp instance of SmartLamp ? >= 0.3
Is mySmartLamp instance of SmartLamp ? >= 0.7

realizing entities as fuzzy members of specific classes

A better definition:

(instance mySmartLamp TrustworthySmartLamp 0.7)
(instance herSmartLamp TrustworthySmartLamp 0.3)

Proposing a computational model

Reuse and extend a state-of-the-art well-defined computation model Bao & Chen 2012*

- on dynamic trust management for community-based social IoT environment

Compute trust values between IoT entities (composing *honesty*, *cooperativeness*, and *community-interest*) using:

- social relationships such as *ownership*, *friendship, community (for entities' owners)*
- introduce a context-depended property, we call *capacity*, as the ability of an IoT entity to function within specific context requirements (e.g. environmental properties such as light, noise, temperature)

Context requirements are specified in the IoT ontology (semantic registry) at the context level definition, and matched against devices' and applications' specs (also specified in the IoT ontology after their registration in the semantic registry)

This matching task results to a capacity signature *cap* of an IoT entity *E* for a specific context *C*, i.e. to a capacity value for each device per context

Cap is taken into consideration for the computation of trust value between two IoT entities

Issues such as the propagation (transivity) and aggregation of trust (composition of *honesty*, *cooperativeness*, and *community-interest metrics*) i.e. how to disseminate and combine trust information are treated by the computational model (the social part)

F. Bao and I. Chen. Dynamic trust management for internet of things applications. In *Proceedings of the 2012 international workshop on Self-aware internet of things* (Self-IoT '12). ACM, New York, NY, USA, 1-6. DOI=http://dx.doi.org/10.1145/2378023.237.

Further Work

- Further refine and engineer the conceptual model
- Complete the implementation of the extended (with *Cap*) computational model and evaluate the computation of context-based trust with no centralized trust authority (based on a prototype NS-3-based simulation system provided to us by Bao & Chen)
- Use case implementation and evaluation of the overall approach in real IoT setting
  - video conferencing broadcasting app deployment in camera/mic-enabled mobile phones of socially-networked attendants in outdoor and indoor social meetings
- Investigate the distribution of IoT-entities' information (context, app and devices properties, trustworthiness), in the absence of a central IoT registry or trustworthiness authority, utilizing social-networking infrastructure (e.g. Facebook API)

Thank You!

kotis_konstantinos

(+30) 6974822712

http://ai-group.ds.unipi.gr/kotis/

kotis@aegean.gr

Trust

'*Trust is an attitude that one takes to the trustworthiness of another; in turn, the other's trustworthiness is a property that they have*' (O'Hara 2012)*

Trustworthiness can be expressed as a quadruple:

Tw <Y, Z, R(A), C>

Y and Z are entities, R is a representation of behavior aimed at an audience A, and C is a context.

This states that Y is trustworthy, assuming that there is some context for Y's trustworthiness. The context C is some type of relevant restriction of the circumstances in which Y is claimed to be willing, able and motivated to conform to R. In our current work, R represents the behavior of 'being reliable' in a specified context and task. Furthermore, if Y is trustworthy in all (or most) specific contexts where she has a duty, or is claimed, to be trustworthy, then it is generally trustworthy.

K. O'Hara. A General Definition of Trust. Technical report, University of Southampton, 2012

Social IoT

A social Internet of Things (IoT) system can be viewed as a mix of traditional peer-to-peer networks and social networks, where “things” autonomously establish social relationships according to the owners’ social networks, and seek trusted “things” that can provide services needed when they come into contact with each other opportunistically.

Ing-Ray Chen, Fenye Bao ; Jia Guo. **Trust-based Service Management for Social Internet of Things Systems**. IEEE Transactions on Dependable and Secure Computing (Volume:PP , Issue: 99)

Smart City example

A smart city IoT application running on Alice's smartphone for air pollution detection. Alice tries to avoid stepping into high air pollution areas (in terms of the levels of carbon dioxide, PM10, etc.) for health reasons. **Alice's smartphone is a member of the air pollution awareness social network**. She decides to invoke her smartphone to connect to sensor devices in an area she is about to step (or drive) into. **Alice knows that many IoT devices will respond, so she needs to make a decision on which sensing results to take**. She instructs her smartphone to accept results only from *n*=5 most "trustworthy" sensors and she will follow a trust-weighted majority voting result. That is, each yes or no recommendation is counted as 1 weighted by Alice's trust toward the recommender. If the total trust-weighted "yes" score is higher than the total trust-weighted "no" score, Alice will step into the area; otherwise, she will make a detour to avoid the area.

Ing-Ray Chen, Fenye Bao ; Jia Guo. **Trust-based Service Management for Social Internet of Things Systems**. IEEE Transactions on Dependable and Secure Computing (Volume:PP , Issue: 99)