ΕΛΛΗΝΙΚΗ ΔΗΜΟΚΡΑΤΙΑ
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗΣ ΚΑΙ ΘΡΗΣΚΕΥΜΑΤΩΝ
ΓΕΝΙΚΗ ΓΡΑΜΜΑΤΕΙΑ ΕΡΕΥΝΑΣ ΚΑΙ ΤΕΧΝΟΛΟΓΙΑΣ

ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
«ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ»

ΕΘΝΙΚΟ ΣΤΡΑΤΗΓΙΚΟ ΠΛΑΙΣΙΟ ΑΝΑΦΟΡΑΣ
ΕΣΠΑ 2007-2013

ΔΡΑΣΗ «ΑΡΙΣΤΕΙΑ»

Deliverable D3.1 Market service and incentive mechanisms' specification

Παραδοτέο 3.1 Προδιαγραφή υπηρεσίας αγοράς και μηχανισμού κινήτρων

ΑΚΡΩΝΥΜΙΟ/ΚΩΔΙΚΟΣ:	INCEPTION/940
ΚΥΡΙΟΣ ΕΡΕΥΝΗΤΗΣ:	ΓΕΩΡΓΙΟΣ Δ. ΣΤΑΜΟΥΛΗΣ
ΦΟΡΕΑΣ ΥΠΟΔΟΧΗΣ:	ΟΙΚΟΝΟΜΙΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ ΑΘΗΝΩΝ (ΟΠΑ)
ΣΥΝΤΑΚΤΕΣ ΠΑΡΑΔΟΤΕΟΥ:	Γ. Θάνος, Κ. Κουρκουμπέτης, Ε. Μαρκάκης, Γ. Δ. Σταμούλης

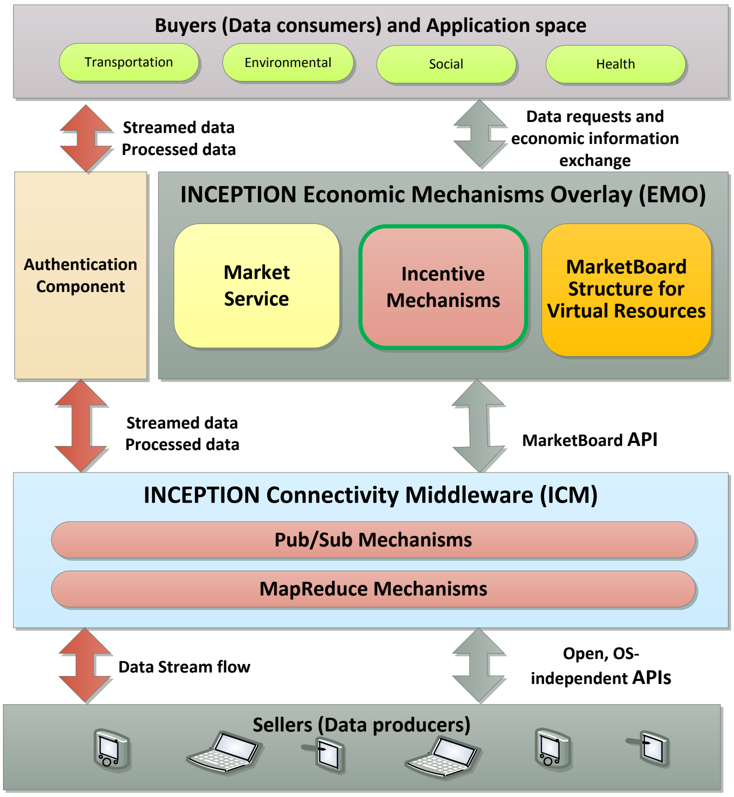
Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013 πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Table of Contents

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

Εισαγωγή και Περίληψη

Ο στόχος αυτού του παραδοτέου είναι να περιγράψει τις προδιαγραφές και το σχεδιασμό της *υπηρεσίας Αγοράς* (Market Service) και των σχετικών μηχανισμών παροχής κινήτρων που ανήκουν στο υποσύστημα του Economic Mechanisms Overlay της αρχιτεκτονικής του INCEPTION, όπως αυτή παρουσιάζεται στο *Παραδοτέο 2.1: Απαιτήσεις συστήματος, αρχιτεκτονική και σενάρια αξιολόγησης* και απεικονίζεται στο επόμενο σχήμα, αλλά και να παράσχει μια θεωρητική ανάλυση για την επίδοση και την αποτελεσματικότητα των μηχανισμών αυτών.



Σχήμα 1. Αρχιτεκτονική INCEPTION

Συγκεκριμένα, η 2η ενότητα του Παραδοτέου παρουσιάζει τις προδιαγραφές της υπηρεσίας αγοράς (η οποία θα αναφέρεται στο επικείμενο παραδοτέο και απλώς ως «αγορά»), ενώ η 3η ενότητα αναλύει τις βασικές έννοιες της οικονομικής θεωρίας, όπως οι δημοπρασίες και η θεωρία σχεδίασης οικονομικών μηχανισμών, που θα εφαρμοστούν στον σχεδιασμό της υπηρεσίας αγοράς. Επίσης, περιλαμβάνει τον ορισμό βασικών εννοιών και όρων όπως οι συναρτήσεις χρησιμότητας και ζήτησης, η μεγιστοποίηση της κοινωνικής ωφέλειας, και παρουσιάζει πως οι βασικές αρχές του σχεδιασμού μηχανισμών θα προσαρμοστούν και στη συνέχεια θα εφαρμοστούν στην περίπτωσή της αγοράς του αναπτύσεται στα πλαίσια του INCEPTION για δεδομένα αισθητήρων.

.

Επιπλέον, η 4η ενότητα παρουσιάζει τη θεωρητική ανάλυση και το σχεδιασμό της αγοράς και των μηχανισμών κινήτρων που θα υλοποιηθούν στην επόμενη φάση του έργου και εξετάζει ορισμένες αποφάσεις που πρέπει να ληφθούν κατά την εξέταση θεμάτων όπως το πρόβλημα της εφικτότητας (feasibility) της ικανοποίησης της ζήτησης (ή μέρους αυτής), η κατανομή του κόστους ανά τους ενδιαφερόμενους χρήστες και η επιλογή των συναρτήσεων υπολογισμού

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

του κόστους ανά χρήστη που πρέπει να χρησιμοποιηθούν στο σύνθετο, από την άποψη της προσφοράς και ζήτησης απαιτήσεων, περιβάλλον του έργου.

Η ανάλυση όλων των παραπάνω, σύμφωνα με τις απαιτήσεις της αρχιτεκτονικής όπως παρουσιάζεται στην Π2.1 ολοκληρώνεται με την επιλογή και το σχεδιασμό τριών διαφορετικών μηχανισμών που θα υλοποιηθούν εντός του επόμενου έτους του έργου, συγκεκριμένα των εξής:

1. Ο μηχανισμός Moulin Shenker που ανήκει στην κατηγορία Strategyproof μηχανισμών για εξισορρόπηση κόστους (cost balancing).
2. Ο μηχανισμός οριακού κόστους: Marginal Cost Pricing (or VCG-like) που ανήκει στην κατηγορία Strategyproof μηχανισμών για μεγιστοποίηση της κοινωνικής ωφέλειας.
3. Ο αλτρουιστικός μηχανισμος (Altruistic) που ανήκει στην κατηγορία συνεργατικών μηχανισμών (Cooperative mechanisms) που μεγιστοποιούν την ωφέλεια από την χρήση αισθητήρων.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

1. Introduction and Executive Summary

The aim of this deliverable is to provide the **specification** as well as a **theoretical analysis** and **design** of the **"Market Service" component** and the associated **incentive mechanisms** that reside in the Economic Mechanisms Overlay component of our architecture, as this was presented in *Deliverable 2.1: INCEPTION requirements, architecture and evaluation scenarios* and depicted in the next figure.

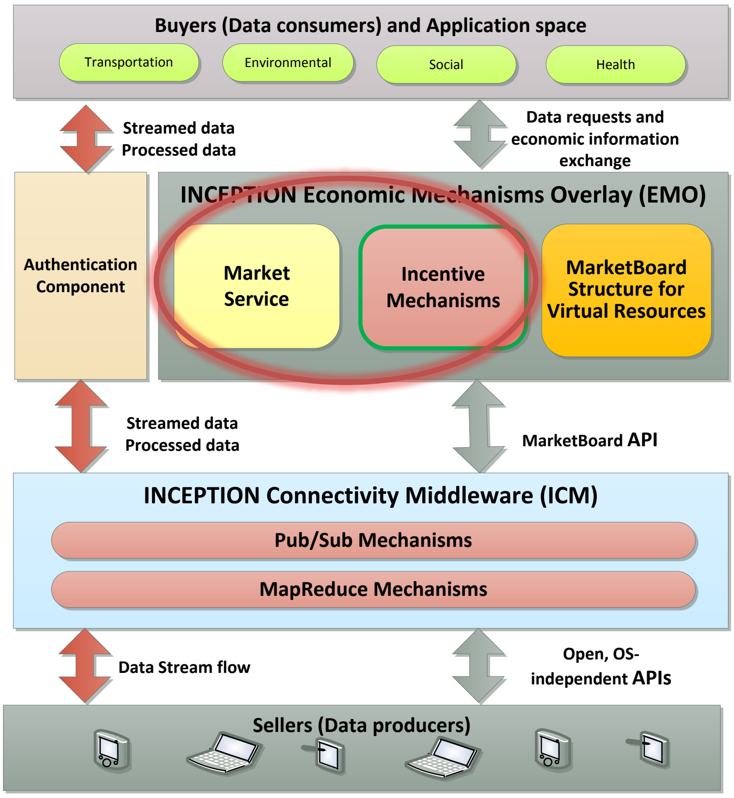


Figure 2. INCEPTION Architecture and EMO component

The next section will present the high-level specification of the market service (which we will also simply refer to as "market"), whereas Section 3 will analyse the basic theoretic concepts from economics, auctions and mechanism design theory that will be applied in the design of the market service. This will include the definition of terms like utility and demand function, social welfare maximization and will study basic principles of mechanism design which will all be analyzed, adapted, and later applied in respect to our case.

Moreover, Section 4 will present the theoretical analysis and design of the service and incentive mechanisms that will be implemented in the next phase of the INCEPTION project and will elaborate on some implementation decisions to be taken when dealing with feasibility issues, cost sharing and cost functions to be used in our composite, in terms of supply and demand requirements, environment.

This analysis of all the aforementioned, in line with the requirements of the architecture as presented in D2.1 will conclude in the selection and design of three different mechanisms that will be implemented within the next year of the project, namely the following:

a. The Moulin Shenker mechanism belonging in the Strategyproof mechanisms for cost balance category.
b. The Marginal Cost Pricing (or VCG-like) mechanism belonging in the Strategyproof mechanisms for social efficiency maximisation category.
c. The Altruistic mechanism belonging in the Cooperative mechanisms that maximise utility from sensor usage category.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

2. High-level specification of the market service

Our goal is to define a market for sensor information. The first definition is related to the goods that are bought and sold. Then, as in any market, we need to define supply and demand for the goods traded in the marker (sensors). Our market operates in discrete time (time slots of a given physical duration).

The goods traded: The basic good that is available in this market is a sensor. To make this more precise, we consider the case where 'buying' a sensor consists of buying the right to have access to the sensor information (data) for a given time slot. We abstract from the market the process of declaring sensor availability and accessing the sensor data. These services are offered as part of the implementation services of the market. At the abstract level of our model we assume that at any given time there is set of sensors (goods) available in the market and this information is globally available (to buyers and sellers). By definition of the sensor good, sensors cannot be consumed since information from the same sensor can be duplicated at (almost) zero cost. Hence sensors can be 'bought' simultaneously (i.e., supply the right to have access to their information) by more than one buyers. This non-consuming aspect of goods makes our market special as we will see.

The supply part: The supply part of the market consists of the suppliers of sensor information. We assume that there is a set of sensor operators (i.e., the individuals owning the mobile or fixed devices that host the sensors). Each such operator makes available his sensors to the market in an autonomous way. He specifies the minimum price he is willing to charge for a particular sensor in his responsibility in order to be operational and supply its sensing information. We refer in many cases to this price as the 'reserve price' of the sensor. In more general situations he may ask for a single charge defined over the set of the sensors instead of pricing individual ones.

There are two ways that a supplier can make available his sensors for access. The first corresponds to selling a right to that information. Hence he may sell a number of such rights (one per buyer), charging for each one a price. In this case, he offers a good (the access right to his sensor) in a practically unlimited quantity (we assume that there are no technical restrictions), using a pricing tariff of his choice, the simplest one being a uniform price p for each such right. In this case the corresponding good is consumed (a buyer consumes a right), but has no cost implication to the supplier. It is the case of a traditional market where buyers pay for the units of the goods they consume.

The second corresponds to the supplier selling all the rights for accessing his sensor to some intermediary (in our case it will be the market operator) for a given price. Then this intermediary can decide on how to charge the buyers when offering these access rights. For instance, the supplier might charge his cost for turning the sensor on plus a premium, and the market operator might share that cost among the buyers. In this case we exploit the fact that information is not consumed merely to the benefit of the buyers. In both cases, the supplier gets compensated for offering the right to others of accessing his sensor. In the first on a per usage basis, in the second he obtains a lump sum amount.

The demand side: The demand side consists of the applications that are interested in acquiring the sensor information in order to provide services to their customers. We think of each such application as a distinct customer in our market whose demand is expressed by specifying the set of sensors he interested in. Our model of customer preferences includes a variety of types. We allow for customers that are 'inelastic' in the sense that they care for a specific set of sensors and if these sensors are not available or they cost more than what they can afford, they walk away. We also allow for customers that are 'elastic' in the sense that can settle for a reduced set of sensors if the maximal desired set is not accessible to them or is too expensive. We assume that each customer specifies his maximum willingness to pay for the set of sensors he requests. In the case of elastic customers, they specify their willingness to pay for each different nested subset of sensors that can meet their operational requirements.

The market mechanism: This mechanism **defines the rules of the interaction between buyers and sellers.** In the simplest case, it consists of an information service where buyers and sellers posts their *bids* (what I am willing to buy and at what maximum price) and *asks* (what I am willing to supply and at what minimum price), and of a matching service that finds matches between such bids (allocates them to buyers) and asks helping the clearing of the market. A traditional market where sellers of sensor access rights post their prices and buyers come and shop what is best for them belongs in this category. Here market prices are affected by supply and demand and we can use traditional market analysis tools to analyse its performance.

Another type of a market is a specialized mechanism that is tailored to a specific finite set of buyer and seller types. Here we assume that the specific agents (buyers and sellers) are randomly selected from a set of specific types, and each agent knows his type but has only an expectation on the types of the other agents. The mechanism is a function (a set of rules) that requests inputs from the agents (their supply and demand functions) and as a result computes the allocations of the goods and the payments from the agents. For instance, buyers and sellers make their bids and asks and the mechanism decides what each buyer gets and how much he pays and what each seller sells and what he gets as a function of the inputs of the agents. Here an agent's payoff (net benefit) depend on the declarations (inputs) of the other agents. Hence the situation is a game where the concept of *Nash equilibrium* (NE) applies. Different choices of mechanisms lead to different NEs. Hence the mechanism is optimized for certain efficiency goals (measured at the resulting NEs) and uses the prior knowledge of the distribution of the types of the agents that participate. Designing such optimal mechanisms is a *Mechanism Design* problem.

The goal of this work in INCEPTION is to define such mechanisms for running the sensor market. We will not be concerned with complex theoretical issues about incentives and the corresponding analysis of the NEs that may arise (our model assumes that buyers and sellers are rational economic agents, which make rational decisions to maximize their profit). Since we target at a market with eventually a very large number of buyers and sellers, we expect that the size of the market will reduce the individual bargaining power of the agents and discourage their strategic behaviour. In practical terms we expect that buyers and sellers will truthfully declare their values and costs for the services to the market place, and under this assumption we will evaluate the performance of each proposed mechanism. Nonetheless, most mechanisms that we are going to evaluate are truthful by design (theory), so we don't need to make that assumption (more on that discussed in the respective theory sections).

3. Definitions and basic concepts

In this section we define some of the basic concepts we will use in our modelling and analysis of the INCEPTION market service and mechanisms. They include definitions from the economics literature and important concepts and mechanisms for cost sharing that we use in the sequel.

We first discuss key economic definitions that traditionally apply to markets and their relevance to our models. As we will see, our market definitions are not standard from the consumer and the supply side. **The key difference is in the price determination mechanism since our goods (sensor information) can be duplicated at no cost and there is substantial cost sharing**. We start with the consumer side of the market. In our case, consumers are applications that need the sensor information in order to provide services to their customers. In some cases we provide general definitions although in our case our goods are not infinitely divisible (one can acquire a single sensor of a given type, or two or more).

3.1 Utility function, surplus maximization and demand function

Consider a market in which n customers can buy k goods. In traditional markets goods are consumed by the buyers, i.e., when one buys one unit of a certain good, this good is not available to other buyers. In the next definitions for utility, nets benefit and demand, this 'consumable good' assumption is not important since we assume an infinite amount of goods available. They play a role when we try to match supply and demand, a role played by the market mechanism that we will discuss later.

Denote the set of customers by $N = \{1, \dots n\}$. Customer i can buy a vector quantity of goods $x = (x_i, \dots, x_k)$ for a payment of $h_i(x)$. The profit π_i of a *customer* from a transaction in the market consuming *x* is called the consumer's *net benefit* or *consumer surplus*,

$$CS_i(x) = u_i(x) - h_i(x). \quad (1)$$

where $u_i(x)$ is the *utility* to customer i of having the vector quantities of goods x. One can think of $u_i(x)$ as the maximum amount of money he is willing to pay, i.e., his *'willingness to pay'*, to receive the bundle that consists of these goods in quantities $x_1, \dots, x_k$. Equivalently, it is the revenue the user can obtain by reselling the good in some market. In many cases we use the term 'value' to refer to the utility of the user obtained when consuming a given good. Note that in our definitions the utility function is expressed in the same units as the payments (dollars or euros, etc.).

In (1) the consumer surplus is the utility of x minus the amount paid for x. Or the difference between his willingness to pay for x and the corresponding payment. If a customer has the choice, he will choose to buy the quantity that maximizes the difference between his willingness to pay and his payment. Hence he will chose the value of x that solves

$$CS_i = \max_x [\, u_i(x) - h_i(x)]. \quad (2)$$

Note that we assumed for simplicity that the available amounts of the k goods are unlimited. Otherwise we had to constrain (2) so that one consumes only what is available.

Let us suppose for simplicity that:

$$h_i(x) = p^\tau x = \sum_j p_j x_j,$$

for a given vector of *prices* $p = (p_1, \dots, p_k)$. Then customer i seeks to solve the problem

$$x^i(p) = arg \max_x [u_i(x) - p^\tau x]. \quad (3)$$

It is usual to assume that $u_i(\cdot)$ is strictly increasing and strictly concave for all i. This ensures that there is a unique maximizer in (3) and that demand decreases with price. If, moreover, $u_i(\cdot)$ is differentiable, then the marginal utility of good j, as given by $\partial ui(x)/\partial xj$, is a decreasing function of x_j .

The vector $x^i(p)$ is called the *demand function* for customer i. It gives the quantities $x^i = (x_1^i, \dots, x_k^i))$ of goods that customer i will buy if the price vector is p. The *aggregate demand function* is $x(p) = \sum_{i \in N} x^i(p)$ this adds up the total demand of all the users at prices *p*.

Similarly, the *inverse aggregate demand function*, $p(x)$, is the vector of prices at which the total demand is x. We must note that $xp(x)$ is not the willingness to pay for the goods in x, which is $u(x)$. If a user is charged his willingness to pay, then he is left with zero surplus.

In our sensor market model, the sensor demand of applications may get the general form discussed above (the case of different willingness to pay for different sensor subsets, i.e., a utility function that accounts for different allocations), or might be very simple by demanding a specific set of sensors (goods) only. This later is the case of inelastic demand.

Sometimes we refer to the case that the customer has a quasilinear utility function, of the form $u(x, w) = w + u(x)$, where w is the amount of money in the bank or his income, and x as before. Assuming his income is large enough that $w - p^T x > 0$ at the optimum, he must solve a problem that is equivalent to (3). It is valid to assume a quasilinear utility function when the customer's demand for goods is not very sensitive to his income, i.e., expenditure is a small proportion of his total income, and this is the case for most known communications goods.

Consider the case of two goods, x, y. We say that these are *independent* if $u(x, y) = u_1(x) + u_2(y)$ for some functions u_1, u_2. In our case this corresponds to an application that obtains value from using these sensors in a way the value from sensor 1 is not influenced from the usage of sensor 2. But in many cases this is not the case and the goods are related in the value they generate. If having access to more of the first good makes the second marginally more valuable, we say that goods are *complements*. The reverse (i.e., more of good x makes y less valuable) corresponds to the goods being *substitutes.*

More precisely, if

i. $u(x + 1, y) - u(x, y)$ (marginal utility of x) is increasing in y, the goods are complements.
ii. $u(x + 1, y) - u(x, y)$ is decreasing in y, the goods are substitutes
iii. If it does not depend on y they are independent.

In the case of sensors we may get all three possibilities. Sensors of the same type (i.e., measuring the same quantity in close geographic locations) might be substitutes if their information is to some degree redundant. Sensors of different types which are all required for the application to perform its computations are complements. Without some of them the value of the rest is reduced since a reduced set of sensors produces inferior results.

3.2 The supplier's problem

Suppose that a supplier produces quantities of k different goods. Denote by $y = (y_i, \dots, y_k)$ the vector of quantities of these goods. For a given network and operating method the supplier is restricted to choosing y within some set, say Y^j, usually called the *technology set* or *production possibilities* set of the given supplier j in the economics literature. In our case this may correspond to the set of physical sensors that can be turned on by the supplier.

Profit, or *producer surplus*, is the difference between the revenue that is obtained from selling these goods, say $r(y)$, and the cost of production, say $c(y)$. An independent firm j having the objective of profit maximization, seeks to solve the problem of maximizing his profit π_j

$$PS_j = \pi_j = \max_{y \epsilon Y}\left[r(y) - c_j(y)\right] \ , \tag{4}$$

where $r(y)$ is the revenue obtained in the given market if the supplier tries to sell exactly y quantity of the goods produced (prices might change since the total capacity available of the goods changes by the addition of y). Note that the producer surplus is defined for all values of output y, and $PS_j(y) = r(y) - c_i(y)$, the difference of revenues minus costs when producing *y*.

An important simplification of the problem takes place in the case of *linear prices*, when

$r(y) = p^\tau y$ for some price vector p. Then the profit is simply a function of p, say $\pi(\varrho)$, as is also the optimizing y, say $y(p)$. Here $y(p)$ is called the *supply function*, since it gives the quantities of the various goods that the supplier will produce if the prices at which they can be sold is p. As in the case of the consumer, we get that the optimal amount to be supplied by j in the case of exogenously defined prices p for the goods is

$$y^j(p) = arg \max_{y}\left[p^\tau y - c_j(y)\right] . \tag{5}$$

The way in which prices are determined depends on the prevailing market mechanism. We can distinguish three important cases. If the supplier is a *monopolist*, that is, the sole supplier in an *unregulated monopoly*, then he is free to set whatever prices he wants. His choice is constrained only by the fact that as he increases the prices of goods the customers are likely to buy less of them. In the case of a sensor market, there may be cases that sensors of a certain type, our good y in (4), are supplied by a single supplier (for instance because he controls the sensors in a specific geographical area). In this case he is free to set the prices he is asking for making his sensors available to the market.

If the supplier is a small player amongst many, or prices are determined by a regulator, then he may have no control over p, and thus he is a price-taker, with no freedom except his choice of y. This is a typical case. An appropriate model is linear prices which are independent of the quantity sold. This is also the case for a regulated monopoly, where the price vector p is fixed by the regulator, and the supplier simply supplies the goods that the market demands at the given price.

A middle case, in which a supplier has partial influence over p, is when he is in competition with a few others. In such an economy, or so-called *oligopoly*, suppliers compete for customers through their choices of p and y. This assumes that suppliers do not collude or form a cartel. They compete against one another and the market prices of goods emerge as the solution to some non-cooperative game.

In the case of supplying sensors all the above definitions are relevant. For instance, a sensor of a given type might be available at different quantities supplied by different providers. This occurs when different suppliers own sensors (of the same type) distributed over the geographical area of interest and customers care to obtain information from a random number among them independently of the supplier and the exact location. In this case we expect sensors to be priced with a uniform price *p* determined by market conditions since these specific sensors are a commodity. A single supplier will not be able to influence that price and hence its decision to offer y sensors in this market is determined by (4) where $r(y) = p^{\tau}y$. This supplier is a price taker. In the case of few suppliers contributing sensors we get the case of an oligopoly.

The case of a regulator for prices p is more interesting. In our approach for designing a market we will use the concept of a `mechanism' (see later discussions). This mechanism will define the rules of the market, i.e., the interaction between buyers and sellers. A simple mechanism we will propose is the **Price Setting Mechanism (PSM).** This mechanism sets prices for selling and buying sensors in the market. For instance, it specifies that price for supplying a sensor of type A is 10*c*, while the price of accessing such a sensor is 8*c*. The logic of choosing these prices will be examined later. But given these prices, supply and demand are decoupled, since suppliers and buyers decide their supply and demand that maximizes their net benefit. In this case the mechanism plays the role of a regulator that specifies the rules of the market operation. Of course, he might change these rules in slow timescales to optimize the operation of the market.

3.3 Welfare maximization

Social welfare (which is also called *social surplus*) is defined as the sum of all agents' net benefits, i.e., the sum of all consumer and producer surpluses

$$SW = \sum_i CS_i(x^i) + \sum_j PS_j(y^j) \qquad (6),$$

$$= \sum_i u_i(x^i) + \sum_j c_j(y^j) \qquad (7)$$

If goods produced are also actually consumed then we also need $\sum_i x^i = \sum_j y^j$ since we need supply and demand to match. In the case of goods that may not be consumed, such as sensor access rights, then the equivalent constraint is that sensors whose access rights are sold in a positive number need to be turned on and the revenue from the buyers of these rights need to cover the costs of the suppliers of the above sensors. Note that there is no need to require this constraint to hold on a per sensor basis. We only need the payments from the buyers to cover the cost of the sellers

$$\sum_i h_i(x^i) = \sum_j c_j(y^j). \quad (8)$$

We refer to the above condition as the **budget balance condition**.

We speak interchangeably of the goals of social welfare maximization, social surplus maximization, and `economic efficiency'. The key idea is that, under certain assumptions about the concavity and convexity of utility and cost functions, the social welfare can be maximized by setting an appropriate price and then allowing producers and consumers to choose their optimal levels of production and consumption. This has the great advantage of maximizing social welfare in a decentralized way.

One can always begin by supposing that the social welfare maximizing prices are set by a supervising authority, such as a regulator of the market. Suppliers and consumers see these prices and then optimally choose their levels of production and demand. They do this on the basis of information they know. A supplier sets his level of production knowing only his own cost function, not the consumers' utility functions. A consumer sets his level of demand knowing only his own utility function, not the producers' cost functions or other customers' utility functions. Individual consumer's utility functions are private information, but aggregate demand may be commonly known to the regulator.

Later we will discuss perfectly competitive markets, i.e., markets in which no individual consumer or producer is powerful enough to control prices, and so all participants must be price takers. It is often the case that once prices settle at the market equilibrium to values at which demand matches supply, the social welfare is maximized. Thus a perfectly competitive market can sometimes need no regulatory intervention. This is not true, however, if there is some form of market failure, such as that caused by externalities.

A simple way to design a competitive market for our sensors is through the use of PSM and adapting its parameters to balance costs. Suppliers declare their costs for turning on and operating their sensors. Then the PSM operator (our `regulator') defines a selling price for a sensor by dividing the cost of the sensor by the number of the *expected* access right requests for that sensor (by the buyers). Then demand materializes for a number of time slots (our model assumes that the market operates in discrete time slots where supply and demand are independently distributed) and the actual demand (number of buyers) for the sensor is revealed. The operator then adapts the selling prices accordingly and the system continues. It might turn out that some sensors are priced very highly by their suppliers. In this case their demand may drop to zero since there is competition from other sensors that are substitute (have comparable information) or because the buyers are not able to afford these prices. This will become known to the suppliers which might lower their prices (or raise them in case the opposite occurs). This simple market mechanism is close to how actual markets work. And if suppliers don't hold monopoly power over certain important sensor types and the number of buyers and sellers is large, we expect that the result will be socially optimal in the long run.

3.4 Mechanisms

A mechanism is the definition of the rules and the actions of a game in which a given

set of economic agents participate. The goal is to design the mechanism in a way that the resulting game has a certain outcome as an Nash Equilibrium (NE), or that the NEs of the game have a certain property like maximize social efficiency among all possible outcomes. In simple terms, since we cannot force agents to make decisions, we like to **define a context (the mechanism) under which the best rational choices of the agents are also the ones that the social planner would wish them to make. We call this compatibility of the incentives regarding the actions to take between the agents and the social planner *incentive compatibility.***

More precisely, consider the traditional Bayesian game setting $B = (N, O, \Theta, p, u)$ where N is the set of agents, O is the set of outcomes, Θ is the set of agent types, p is a distribution over the agent types, and u is a set of utility functions over the set of outcomes and agent types, one per agent. Hence for a given vector of types and a given outcome we can compute he payoff for each agent by applying his utility function. In the above setting, agents have private information since each agent knows his type but has only a probability distribution for the actual types of the other agents. A Mechanism for a Bayesian setting B is a pair (A, M) where

$A = A_1 \times \cdots \times A_n$ where A_i is the set of action available to agent $i \in N$, and

- $M: A \to O$ maps each vector of actions by the agents to an outcome (more general to a distribution $\Pi(O)$ over the set of outcomes).
- We say that agents have *quasilinear preferences* with *transferable utility* if the set of outcomes is $O = X \times \Re^n$ for a set *X*, and the utility of an agent i can be written as

$$u_i(o, \theta) = u_i(x, \theta) \; where \; o = (x, p) \in O \; and \; u_i : X \times \Theta \to \Re$$

Hence the designer of the mechanism must specify the actions that are available to each agent, and the mapping of actions to outcomes. In the case of quasilinear preferences with transferable utility, the outcome of the mechanism consists of an allocation $x \in X$ (the resources that each agent will get) and a side payment (positive or negative). The capability to include payments as part of the action space makes our mechanisms very powerful and increase the range of achievable outcomes.

Some more terminology: If the mechanism defines as the action space the set of agent types (i.e., agents are asked to report their types) it is called a *direct* mechanism, otherwise (more general) it is an *indirect* mechanism. In direct mechanism, in general, truthful reporting is not an NE strategy. *The revelation principle* ensures that if a certain NE is achieved by an indirect mechanism, then it can be also achieved by a direct mechanism where truth telling is the NE strategy employed by the players. In other words, we don't restrict ourselves if we restrict our mechanism design space to mechanisms that ask players to report their types and being truthful is an NE strategy. A stronger definition is about s*trategyproofness*. **A mechanism is *strategy proof* if it is direct and truthtelling is a *dominant strategy* (not just an equilibrium strategy at some NE). We also discuss a stronger form of incentive compatibility, *group strategyproofness*, where no coalition of agents has an incentive to jointly misreport their true willingness to pay.**

For example, in our setting, we ask from the sensor suppliers to declare their costs (an action in this case is a declaration) and the buyers to declare their utilities. Here we can easily model that the type of an agent is a parameter that specifies his cost or utility for supplying/using sensors. The rules of the mechanism are known to all agents and determine a function that takes as input the declarations and generates as an output the sensor allocations (which suppliers and which buyers will supply and consume and by how much) and the payments (how much buyers should pay and how much sellers should receive). This defines a game where for every vector of actions the players receive a certain payoff (their net benefit from the allocation minus the payment). The solution of this game is an NE, and there may be multiple equilibria. A mechanism design problem is to define the rules of the mechanism such that the resulting NE is also the operating point that the social planner would choose based on the actual agent profiles (types). The revelation principle ensures that we can search for such a mechanism while also requiring that agents act truthfully in reporting their costs and values. In some cases we may be lucky and come up with a mechanism that is strategyproof, and hence agents choose truthtelling as a dominant strategy and don't have to worry about what other agents do.

3.5 Other criteria for optimality of the allocations

So far the main criterion for choosing among different allocation schemes is **the total net benefit of the players,** which is also equal to the sum of the utilities minus the sum of the costs incurred. A second criterion is **budget balance**, see (8). We may require from a mechanism to induce payments that exactly cover costs. In general, such a requirement is defined ex-ante or ex-post (in an expected sense, or always, after each instance of the mechanism is executed). In our case, since the system is running repeatedly we are satisfied with the looser ex-ante definition of budget balance.

A third criterion might involve the **sustainability of the system** in the long run. One way to define sustainability is by considering besides budget balance also other practical criteria such as satisfaction from the market operation and probability of having a successful market transaction. It is true that most models define costs in the short run, including only the costs from operating the sensors. But such a definition fails to include the cost of applications that go to the market and come back empty handed. This suggest other metrics to use to judge market effectiveness, such as **maximize the set of sensors that are being activated (the `footprint' of the allocation)** by maximizing the number of sensors that a bidder gets allocated, or maximizing the total value generated by running the system without subtracting the cost incurred by the sensor providers, as long as budget is balanced (providers get paid for their costs). Our practical mechanisms are evaluated using all the above criteria.

3.6 The Moulin-Shenker mechanism

We give first a quick overview of the Moulin-Shenker setting (we will refer to this as the *"MS setting"*) [MS01]. This is simply a special case of our setting, hence our problems are at least as difficult as theirs. Nevertheless, the ideas from Moulin-Shenker are inspiring for designing cost-sharing schemes in our setting as well and we will use them along with some variations.

Suppose that a set of people $N = \{1, \dots, n\}$ (potential customers) is interested in receiving a service, offered by a provider. The setting here is binary, meaning that the service is either granted or declined to each user. The service comes with a cost which depends solely on the set of customers chosen to receive it. Hence, for each $R \subseteq N$, C is the cost of the provider for providing the service only to set R. It is also assumed that the cost function is submodular

$$C(empty) = 0;\ R \subseteq T \Rightarrow C(R) \leq C(T)$$

$$C(R \cap T) + C(R \cup T) \leq C(R) + C(T)$$

An equivalent property of a sub modular cost function is that the marginal cost decreases as the size of the coalition increases, i.e.

$$R \subseteq R' \Rightarrow C(R' + i) - C(R') \leq C(R + i) - C(R)$$

Each user j has some value u_j for receiving the service (his private information) which is the maximum amount of money that he is willing to pay for receiving the service (his utility for the service). He is asked to declare a bid b_j, which may not necessarily be equal to u_j. The mechanism deployed by the provider collects the bids from the bidders and then needs to decide

a) which set $S \subseteq N$ is going to receive the service, and
b) how much to charge each $j \in R$.

According to our previous definitions it is a direct mechanism.

The usual objectives that are studied for such mechanisms include cost-covering and social welfare maximization.

For cost-covering, the mechanism proposed by Moulin-Shenker depends on the existence (definition) of an underlying *cost-sharing* method. A cost-sharing method is a function $\xi(\cdot,\cdot)$ such that $\xi(j, R)$ determines the cost-share of agent j $(j \epsilon R)$ when *R* is the set to be serviced. We demand that a cost-sharing method satisfies

$$\sum_{j \in R} \xi(j, R) = C(R) for\ all\ R \subseteq N, \qquad (9)$$

i.e., that the sum of the payments balance the budget (cost) for serving the coalition of the agents.

The following is an important property satisfied by many cost-sharing methods.

Definition 1 A *cost-sharing* method is *cross-monotonic* if

$$\xi(j, R) \geq \xi(j, T)\ for\ R \subseteq T\ and\ j \in R \qquad (10)$$

The above property simply says that the cost-share of an agent should not become bigger when more people receive service.

Given a cross-monotonic cost-sharing method, one can define now the mechanism below for determining who gets the service along with the cost-shares. This class of mechanisms has certain nice properties as described later.

The **Mechanism $M(\xi)$** (for a given cost-sharing method ξ):

i. Start by trying to offer the service to all agents, with cost-share $\xi(j, N)$). Kick out any agent who cannot cover this cost-share, i.e., anyone for which $b_j < \xi(j, N)$.
ii. If no-one was kicked out, then stop here, otherwise let R^1 be the set of agents that remain in the game.

iii. See if we can service R^1 with a cost-share of $\xi(j, R^1)$ $for\ every\ j \in R^1$. Again kick out those who cannot afford this price.

iv. Continue like this and in every round obtain the set $R^{t+1} = \{j \in R^t : b_j \geq \xi(j, R^t)\}$.

v. Stop whenever we reach a set in which all agents can afford to pay their cost-share. Let $\hat{R}(\xi, u)$ be the final set. We call this final set stable in the sense of MS (Moulin-Shenker). We say that a set R is stable if it is also the final set after applying the above mechanism.

We denote the above mechanism by $M(\xi)$ (the `MS mechanism' using ξ), since it is derived from the cost-sharing method ξ. One of the main results in [MS01] is the following:

Theorem 1 *Given a cross-monotonic cost-sharing method ξ, and a submodular cost function $C(\cdot)$, the mechanism $M(\xi)$ described above is budget-balanced and group-strategyproof (it satisfies other properties too, like voluntary participation, since at the equilibrium any agent never pays more than his utility for the service).*

The interpretation of $M(\xi)$ as a game is that agents declare their values through their bids, then the mechanism operator runs the algorithm described above and determines the set $\hat{R}(\xi, u)$ which is also the largest stable set. For this set $\hat{R}(\xi, u)$ he computes which agents will be served, i.e., the indicator function $q_j(u)$ that is one of agent j is selected and zero otherwise, and informs them about their payment $x_j = \xi(j, \hat{R}^t)$. Strategyproofness implies that declaring $b_j = u_j$ is a dominant strategy for each agent j.

An important property proved in [MS] is that for any mechanism $M(\xi)$ satisfying basically the properties of strategyproofness, budget balance and agents making non-negative payments and participating in a voluntary fashion to the system (can never get negative profits), the social welfare (sum of utilities minus total cost) achieved by M can be achieved by $M(\xi)$ for some appropriately chosen cross-monotonic cost sharing function ξ. Hence if one looks to maximize the social welfare while balancing costs and preserving strategyproofness, it is enough to look among mechanisms in the class of $M(\xi)$, the key being the right choice of a cross-monotonic ξ. For completeness we recall that the social welfare achieved by $M(\xi)$ is

$$SW_{M(\xi)} = u_{\hat{R}(\xi,u)} - C(\hat{R}(\xi, u)) \,, \tag{11}$$

where $u_{\hat{R}(\xi,u)}$ is the sum of the utilities of the agents that get served at the stable set $u_{\hat{R}(\xi,u)}$. The optimal social welfare without imposing any constraints on the mechanisms to achieve it (such as budget balance, etc.) is

$$SW^*(N, u) = \max_{R \subseteq N} [u_R - C(R)], \tag{12}$$

where $u_R = \sum_{j \in R} u_j$. Let $R^*_{SW}(N, u)$ be the (an) optimal set of customers that achieves the maximum. An important property of $M(\xi)$ proved in [MS] is that a cross-monotonic ξ that maximizes the social welfare among all such strategy proof and budget balanced mechanisms is the Shapley value.

3.7 VCG-like mechanisms

If we relax our assumption for balancing the budget while keeping strategyproofness and voluntary participation (an agent is never charged an amount higher than his utility of the service and hence he always get a non-negative profit from participation) we can use the traditional definition of Groves mechanisms where

$$x_j(u) = u_j q_j(u) - (SW^*(N,u) - h_j(u_{-j})). \quad (13)$$

$SW^*(N,u)$ is the optimum social welfare defined in (12), h_j is a function that depends on the values $u_k,\ k \neq j$ and $q_j = 1,\ j \in R^*_{SW}(N,u)$indicates the agents in the optimizing set of agents. The *marginal cost* pricing (MCP) mechanism is defined by (13) with $h_j = SW^*(N-j,u)$. An interesting property is that it never runs a budget surplus (the amount collected is below total cost).

One can compute the worst case budget deficit induced by the MCP mechanism (in the sense of considering all possible utility functions $\in \mathfrak{R}^n$. This is equal to

$$\sum_{j \in N} C(N-j) - (n-1)C(N).$$

Another interesting mechanism for our case is the *pivotal* mechanism (PM) in [GL79a] and [GL79b] which assigns to each participating agent the net utility

$$u_j - x_j(u) = SW^*(N,u) - \max_{T \subseteq N-j} \left[u_T - \frac{|T|}{|T|+1} C(T+j) \right]. \quad (14)$$

This mechanism has the property of never running a budget deficit (always runs a non-negative surplus), but fails voluntary participation since some agents may be charged more than their utility. Although this might be a problem in some applications, in our system if the same application is visiting the system on a repeated basis, it will care for the average net benefit it obtains by such a payment rule. And this quantity might be a positive number for all agents.

3.8 Cooperative mechanisms that maximize utility from sensor usage

So far we proposed mechanisms where agents act rationally when a single market interaction is assumed. In the case of repeated interactions one might expect that agents might sometimes when their willingness to pay is high subsidize others in order to maximize the total utility created by turning on more sensors than under an egalitarian cost sharing. Also in this case the participating agents that would otherwise been unable to participate contribute their 'unfairly small' contribution to the total cost of the system, making it possible to turn more sensors on. If agents' types are not fixed but are drawn from a probability distribution each time they enter the market, it may be the case that they may like such mechanisms and act truthfully because some implicit 'tit-for-tat' mechanism punishing the selfish agents. We are not concerned with the details of such incentive mechanisms here. We like to define some plausible mechanisms that maximize total utility obtained from the system of sensors and share the resulting cost.

The utility maximization problem with the total budget balance constraint is

$$\max_{R \subseteq N} U_r \; such \; that \; u_R - C(R) \geq 0 \,. \qquad (15)$$

The resulting set R of customers has a total willingness to pay that covers the total cost and there is in general some remaining budget $u_r - C(R)$ that has not been spent. There are many different ways to split this among the participants.

Based on this concept we will develop a set of *altruistic mechanisms* (customised to each of our demand scenarios) in which the "rich" customers that can pay their shares and still have sufficient budget left to spare, subsidize customers that cannot pay their shares thus maximising the satisfaction in our system (more sensors are "on" meaning we have higher possibility of returning customers thus increasing the sustainability of our system). Evidently, depending on the scenario the subsidization rules are different (i.e. who subsidizes which user and on what constraints).

4. The market service and mechanisms – Design and theoretical analysis

We will relate our INCEPTION system of sensors and customers (the applications) with the MS setting introduced earlier. First we need to define precisely our set of sensors.

Let I denote the set of sensor *basic types*, $I = \{1, 2, \dots K\}$. The type of a sensor describes the measurement information that the sensor provides, and a specification of such basic types defines an equivalent class of sensors: two sensors are of the same type $i \in I$ if they are 'equivalent' with respect to a set of applications (our 'customer' types in this market). Clearly, different sets of applications will define different such equivalent classes (some may care for a more refined definition of the attributes of the sensor, while some might not). We leave out of this part of the work the procedure one needs to follow to define the set I. For simplicity we assume that it is fine enough to distinguish sensors that our applications consider as being different. Since geographical location can be important to applications, we assume that sensors measuring the same phenomenon in different locations may also correspond to different basic types in the set I. In general we expect to have a number of different sensors of the same basic type available (for instance, when exact location is not important within a larger geographical area, which makes these sensors equivalent with respect to the applications).

A unique distinguishing factor of sensors of the same basic type is their cost. Since sensor cost is important in selecting sensors, we augment the concept of sensor type to be the tuple (i, k) where $i \in I$, and $k \in 1,2, \dots$ denotes the relative cost ranking of the specific sensor compared to other sensors of the same basic type. For instance, $(2, 1)$ refers to the cheapest sensor of (basic) type 2. This is useful because when we ask for 3 different sensors of type i, we always refer to the specific sensors $(i, 1), (i, 2), (i, 3)$ - we consider the 3 cheapest ones. Since sensors may have the same cost, in order to unambiguously refer to a specific physical sensor we assume that our ordering takes into account other attributes when needed (i.e., use lexicographic ordering when two sensors have the same cost). Hence we assume that (i, k) specifies a unique physical sensor in our system.

Also it is not always the case that a sensor of a specific simple sensor type is available (e.g., $(i, 1), (i, 2)$ may be available but not $(i, k), k \geq 3$. Let $\mathcal{L}$ be the set of simple sensor types, i.e., the Cartesian product of *I* with positive integers. A *supply instance* in our market is modelled by a *location function* $I(i, k)$ that maps a simple type (i, k) to a specific physical sensor (when such a sensor type is available) or is undefined (when such a sensor is not available).

To simplify our notation, depending on the context, we use i and the term 'type' to refer to a simple type or to a basic type (usually the former). We will use the indicator function $a(i)$, $i \in \mathcal{L}$ that returns 1 iff i is available in the market, and $c(i)$ to denote the cost of the corresponding physical sensor ($c(i)$ is defined only if $a(i) = 1$).

Definition 2 *A set S of sensor (simple) types is feasible if* $a(i) = 1 \; \forall \, i \in S$.

Unless otherwise stated, $C(S)$ is defined only for sets S that are feasible. We remind the reader that we use R to denote sets of customers and S to denote sets of sensor types. Hence $C(R)$ refers to the cost of servicing the set of customers in R by turning on their requested sensors, while $C(S)$ refers to the cost of the sensors in S.

Let us relate now our system to the MS setting. In our case we assume that the 'service' in the MS setting corresponds to providing access to sensors. In particular, we assume an initial set N of customers that request service (as in the MS setting). Each customer $j \in N$ has a different service profile, i.e., needs a different set of sensors. Let $S(j) \subset \mathcal{L}$ denote the set of sensors requested by customer j. Then for a set $R \subseteq N$ of customers, the cost of serving it $C(R)$ is the sum of the cost of all the sensors that are required by at least some customer in R. Using our definitions above, this is

$$C(R) = \sum_{i \in S(R)} c(i)\,, where\ S(R) = \cup_{j \in R} S(j). \qquad (16)$$

In this notation we defined as $S(R)$ the union of the sets of sensors requested by customers in R. Our definition of the simple sensor types can be visualized as if in the *x*-axis in the plane we have the basic types $I = 1, 2, \ldots$ and in the *y*-axis the ranking according to cost. In this setup it does not make sense for a customer to request the kth sensor of some basic type when he is not requesting also the $k - 1th$, since the later one is less expensive. Hence for any customer $j, S(j)$ must satisfy the structural property that $(i, k) \in S(j)$ only if $(i, k - 1) \in S(J)$, unless $k = 1$.

Property 1 *For any set of sensors S corresponding to the demand of some customer*

$$(i, k) \in S\ only\ if\ (i, k - 1) \in S\ unless\ k = 1. \qquad (17)$$

The following important property holds for the cost function $C(S)$

Proposition 2 *The function $C(R)$ is submodular*

Proof: It is enough to prove that $C(R + j) - C(R) \geq C(R' + j) - C(R')\ when\ R \subseteq R'$. The cost difference of adding an extra customer *j* is due to the extra sensors that need to be turned on that are requested by this customer and were not requested by the existing customers in R, i.e., the sensors in $S(i) - S(R)$ (set difference). But if $R \subseteq R$, in R' we already turned on the same sensors we had in *R* and possibly more. Hence the extra sensors needed by j cannot be more than before. Formally, since $R \subseteq R' then\ S(i) - S(R') \subseteq S(i) - S(R)$.

And a note about cost. In our work <u>we define as cost of a sensor the amount that the sensor provider asks for turning the sensor on and making it available to the market in order to be shared among the applications that need to use it.</u> We are not concerned if this is the true cost of the sensor or if the provider is asking for a higher price that his true cost. We assume that because the system is large and there are many providers of sensors, the <u>supply side will have small bargaining power and prices will reflect true costs</u>. But of course, in the case of sensors that have some kind of a monopoly in their type or geographical location, one will expect the cost charged by the owner of such a sensor to be above actual cost. In any case, the definition of the social welfare in our system assumes as costs the amount that the market operator needs to pay to the sensor providers for the usage of their sensors. Hence, sensor providers are not part of our ecosystem and we don't consider their net benefit from the transactions of the market as part of the total welfare of the system.

So far we discussed a model where demand is *inelastic*, i.e., each customer j specifies the unique set $S(j)$ that he requires. We like to extend this demand model and allow for *elastic* demand, where customers are allowed to express their willingness to pay not just for a single set of sensor types. For example, an application j may be like to use as many sensors measuring temperature in a certain area as possible to improve its accuracy. Assuming that a large number of such sensors (say of basic type i) are available, and if $u(k)$ is the utility of using data from k sensors, it needs to determine the value of *k* that maximizes its net benefit $u(k) - \sum_{1 \le l \le k} c(i,l)$. In this case it expresses its desire to acquire any possible set of sensors of basic type i and

$S(j) \in \{(i,1)\}, \{(i,1),(i,2)\}$ ... Hence its demand for sensors is a nested (ordered) family of sets as in the definition below.

Definition 3 *The elastic demand for customer j is defined as*

$$F(j) = \{S(j,1), S(j,2), \dots\} \; where \; S(j,k) \subset S(j,k+1)\,,$$

where each set $S(j,k)$ must satisfy the structural Property 1. For inelastic demand $F(j) \equiv S(j)$.

We must stress that our definition of elasticity corresponds to a customer asking for increasing quantities of sensors, all from the same basic type. We don't model the general case where the demand of a customer is a disjunction of arbitrary sensor sets. Intuitively, a customer wants different quantities of the "same" sensors, and he can only specify his desire for 'more' (not strictly).

In many cases we refer to the *'economic demand'* of the customer, i.e., to the pair of sets of sensors required and his utility for that set. In the case of elastic demand this is fully specified by combining $F(j)$ with the vector of utilities $u(j) = u(j,1), u(j,2)$ ...

In the above definitions demand might take very general forms. In this work we will focus to three general demand scenarios. We assume that customers come from the set $N = 1, \dots, n$ and each customer may be of a different type.

1. **General inelastic demand.** Customers present their required sets of sensor types together with their utilities (willingness to pay).
2. **Elastic demand for multiple instances of the same tuple** (vector) of basic types $(i_i, \dots, i_m)$, $i_l \in I, 1 \le l \le m$. Each customer $j \in N$ expresses his willingness to pay for any number $1, \dots, n_j$ of these tuples. We call this the *nested elastic demand* case.
3. **Inelastic demand where the demand sets $S(j)$, $1 \le j \le n$ of the customers are nested.** We call this the *nested inelastic demand* case.

The potential market mechanisms we like to analyse fall into three categories (types):

1. **Strategy proof mechanisms that balance cost.** We apply the MS setting introduced in Section 3.6 hopping to achieve good social welfare results.
2. **Strategy proof mechanisms that maximize social efficiency but do not balance the budget.** We apply the VCG-based mechanisms introduced in Section 3.7. We also define a variant of this mechanism that tries to balance the budget in the long run.
3. **Cooperative mechanisms maximizing to total utility obtained in the system**. We apply our 'altruistic' mechanism proposed in Section 3.8.

Not all mechanisms apply in a straightforward way to the case of elastic demand mentioned above. For instance, the PSM is generic enough to apply for all demand cases. But the rest of the mechanisms require some modifications to be applied in the case of elastic demand. Before analysing the three cases of demand, we state some general implementation considerations.

4.1 Implementation considerations

We recall that our market service operates in discrete time. At the beginning of each time slot there is a set of customers that express their demand for sensors. Let's summarize some implementation complications that our setting has when applying the approach.

1. Feasibility issues. We already mentioned that it will not always be the case that we can serve all the customers even if prices were not an issue, simply because we may not have enough sensors to do so. Hence, we first have to run some algorithm that checks feasibility of the demand expressed by each customer. If the instance is not feasible we also have to decide which subset of the requests to serve, and clearly we need to pick the maximal such subset which is unique. This remark applies to both cases of elastic and inelastic demand.
2. In the all the mechanisms we considered above, the cost function is explicitly required and is assumed as given. Here our cost function is the minimum amount of money we need to collect in order to pay the suppliers who will satisfy the demand. Determining this is not always an easy minimization problem. For example if the suppliers do not ask for a price per sensor but instead for a total price in order to have all their sensors available, then it is not always obvious what is the best solution for sharing this total cost among the sensors. Clearly such a decision will depend on the optimization criterion we like to achieve. But it raises interesting issues of fairness regarding how we treat our customers which are finally affected by the cost allocation to sensors. This issue remains an open topic of research.
3. In the case of elastic demand the problem of cost sharing is complex and simple heuristics may not work. For instance, even when all customers have the same demand, i.e., characterized by the same nested sequence of sets of sensor types, applying a traditional cost sharing mechanism (such as MS with egalitarian cost sharing) is not straightforward. The heuristic of applying the mechanism to each demand set in the nested structure is not optimal and one should consider any combination of such sets, one for each elastic customer (see more in the corresponding analysis). This raises many implementation complexity issues.

Next we present our specific market mechanism proposals for the three demand scenarios introduced earlier. For each scenario our implementation must deal first with the feasibility issue and determine the maximal set of customers for which their requested sensor types are available (feasibility according to Definition 2). For simplicity we assume that all such requests are feasible.

Next, we need to implement efficiently our cost function $C(S), S \subset \mathcal{L}$. This function returns a value for each set S of sensor types, or is undefined if S is not feasible. We will not deal with the practical aspect of implementing such a function, because they depend on the way the sensor information is represented in the database of the market, and if sensor information is available on a push or on a pull basis. For the description of our mechanisms we assume that $C(S)$ is available.

We remind the reader that our market operates in discrete epochs and the arrival of customers that request service in each epoch is i.i.d. (*independent and identically distributed)* from a known type distribution. Hence, the set N of customers that are present in each market instance is a random variable. A simple way to construct it is to assume that $N = (1, \dots, n)$ where n is fixed, but the type of customer $j \in N$ is randomly chosen from a finite set of types. The type of a customer specifies its demand $S(j)$ and its utility $u(j)$ (see definitions above).

4.2 General inelastic demand

In this scenario each customer $j \in N$ is interested in receiving access to sensors in the set $S(j) \subset \mathcal{L}$ of simple types. We call such customers *single-minded* in analogy to single-minded bidders in combinatorial auctions. We assume that the function $C(S)$ is available. By its construction it returns the cost of the cheapest available sensors of the basic types specified in the request of a customer. We remind the reader that this due to Property 1 of the structure of any demand set S. We discuss next the various mechanisms that we use to serve the customers.

4.2.1 Strategyproof mechanisms for budget balance

We apply the MS setting introduced in Section 3.6, which always achieves budget-balance, and we would like to evaluate the performance of these mechanisms with respect to the social welfare they achieve. The cost sharing function ξ we use is the egalitarian one. Let's see this in more detail.

What we assume as given, is a set of customers *R*, and for each customer his demand for sensor types $S(j), j \in R$. For each sensor type $i \in \cup\ S(j), j \in R$ define his sharing parameter $y(i)$ as the number of customers j in R that have i in their demand set $S(j)$. Egalitarian cost sharing occurs if each such customer j pays a share $c(i)/y(i)$ of the total cost $c(i)$ of the given sensor type. Hence our cost sharing function is defined as

$$\xi(j, R) = \sum_{i \in S(j)} \frac{c(i)}{y(i)}. \tag{18}$$

The market mechanism becomes now the mechanism $M(\xi)$ as described in Section 3.6. This mechanism starts with the initial set of customers N and computes the stable set $\hat{N}(\xi, u) \subseteq N$ of the customers that will finally receive service.

$M(\xi)$ has some interesting theoretical properties.

Proposition 3 *The cost-sharing method described by (18) is cross-monotonic.*

Proof: Adding an extra customer in the set *R* cannot increase the cost share $\xi(j, R)$ since the new values of $y(i)$ for each sensor type i can only increase.

Theorem 4 *The Mechanism $M(\xi)$ for single-minded bidders is budget-balanced and group-strategyproof.*

Proof: The mechanism is clearly budget-balanced. That it is also group-strategyproof follows from the same arguments as in the Moulin-Shenker setting.

4.2.2 Strategyproof mechanisms for social efficiency

We apply the VCG-based mechanisms introduced in Section 3.7. The resulting mechanism is

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

the Marginal Cost Pricing (MCP) Mechanism. In this mechanism each customer j declares (bids), his utility $\hat{u}(j)$ for the set of sensors $S(j)$. The mechanism decides as a function of all the declarations $\hat{u} = (\hat{u}(1), \ldots, \hat{u}(n))$ which customer finally participates in the sensor allocation by solving the social welfare maximization problem

$$SW^*(N, u) = \max_{R \subseteq N} [u_R - C(R)], \quad (19)$$

where $\widehat{u_R} = \sum_{j \in R} \hat{u}(j)$, choosing the ones in the argument $R^*_{SW}(N, \hat{u})$ that achieves the maximum in (19). Then, it charges the payments

$$x_j(\hat{u}) = \hat{u}_j q_j(\hat{u}) - (SW^*(N, \hat{u}) - SW^*(N - j, \hat{u}) \quad (20)$$

for the customers in the set R^* (q_j=1 $for\ j \in R^*_{SW}(N, \hat{u})$).

We know the above mechanism is strategyproof [V61], [V71], [G73] i.e., the dominant strategy of the agents is to declare their true utility value. An interesting property is that it never runs a budget surplus (the amount collected is below total cost) and a participating customer is never charged above his utility, hence he has always the incentive to participate in the mechanism.

Implementing this mechanism however, and solving (19) is a difficult computational issue. In the general case it is an *NP*-hard problem. Hence we need to resort to some heuristic or approximation algorithms. If we do that, the payments in (20) do not necessarily imply a strategyproof mechanism any more. Clearly, these are purely theoretical results. In practice, we expect the participants to act truthfully, even when the MCP mechanism uses approximations of the $SW^*(N, \hat{u})$. In our experimental section we will describe such approximations.

4.2.2.1 The Heuristic algorithm for optimal SW approximation

In our experiments we have implemented the exact VCG mechanism, which however can work well for up to 20 sensors (however we should have in mind that for a typical participatory sensing environment we should not expect more than 5 sensor types to be used from smart devices). For larger scale instances, we have implemented a greedy heuristic as follows: The idea is that the heuristic simply tries to add any buyer that causes an increase in the current social welfare.

Step 0: Set B^* = emptyset, S^* = emptyset (by the end of the algorithm, B^* will be the set of buyers who will be served, and S^* will be the set of selected sensors).

Step 1. Find which buyer on his own achieves maximum difference of *utility - SumSensorCost* i.e. his NB. Include him in B* and the sensors he wants in S*. At this point the current social welfare equals this player's net benefit. We denote by SW(B^*) the current solution of the heuristic

In the next steps we check to see if we can find some buyer, such that by adding him to the current solution, we have an increase in the social welfare. We check first for buyers whose demand set has a non-empty intersection with the current set of selected sensors S^*.

Step 2. Check if anyone else wants the exact same set of sensors as S^* or a subset of S^*. If yes include him in B* and calculate the new SW(B^*). We know for sure that this increases the social welfare, since we are only adding more utility and the cost remains the same.

Step 3. Check if there is any buyer who wants a set with at least one common sensor with the currently selected set S*. If by adding this buyer to the current solution we have an increase in the social welfare, then update B* and S* accordingly.

Step 4. Check then for bidders that want one sensor (regardless if it is included in S* or not). If they increase SW(B*) then update B* and S* again.

Step 5: Finally check to see if there is any other buyer whose addition to B* increases the social welfare.

Step 6: Repeat the previous steps until we cannot find any buyer whose addition to B* increases the social welfare.

4.2.2.2 An adaptive algorithm for balancing budget

We already mentioned that using the MCP mechanism generates optimal social welfare, but it does not balance the budget (we run short). A way we propose to remedy that when the system operates over long time periods is to raise some extra revenue by imposing a fixed fee to customers in order to allow them to bid and obtain resources. This fee should not depend on the declarations of the customers in order to preserve truthful behaviour. There are several choices for defining this fee.

- Make the fee depend on the type of customer. Here we assume that a customer is inelastic in his requirements and cannot find it profitable to impersonate a different customer type. By running our MCP mechanism we can estimate the average net benefit $nb(j), j \in N$ and the average deficit d of the mechanism. Then we define such fixed fees $f(j)$ that are proportional to the net benefits of the various customer types and add up to the average deficit. In particular, if by j we denote the type of a customer and $n(j)$ the average number of such customers in a market instance, we require that

$$\sum_j n(j)f(j) = d \text{ and } \frac{f(j)}{f(k)} = \frac{nb(j)}{nb(k)} \tag{21}$$

It easy to see how we can satisfy this in the steady state by repeating the approach in the algorithm introduced previously. We start with some initial guesses for $d^0, n^0(j), nb^0(j), j \in N$ and compute the corresponding $f^0(j)$s using (21). Now we observe how these affect the system and estimate $d^1, n^1(j), nb^1(j), j \in N$ based on the actual market data, recompute the $f^1(j)$s using (21), etc.

- Make the fee depend on the type of sensor. In this case we like to charge more for sensor access in order to recover our deficit. The issue is how to allocate this extra cost per sensor access. Sensors will have a different value of the sharing parameter $y(i)$, $i \in \mathcal{L}$ (the average number of customers that share this sensor type). We define our fee $f(i)$ per sensor type i access to depend on this parameter and the absolute cost value of the sensor. In analogy to (21) we have

$$\sum_{i \in \mathcal{L}} y(i) f(i) = d \; and \; \frac{f(i)}{f(l)} = \frac{c(i)}{c(l)}. \quad (22)$$

The adaptation procedure is similar as in the previous case.

4.2.3 Cooperative mechanisms for utility maximization

Here we exploit the ideas introduced in Section 3.8. Our proposed algorithm is described below:

Step 1. Calculate initial sets B_R(ich) and B_P(oor) i.e. we compute:

a) the demand per sensor and the equal share if allocated to all buyers that want it

b) how many can pay their share with their budget, and the exceeding amount. The total from all of these is our "bucket" that we also in Scenario 2 of Section 4.3.3. We include these bidders in Set *B_R*

c) how many cannot afford to pay and the missing amount. We include these in Set *B_P*.

Step 2. Selecting subsidization pairs

If there is no bidder in *B_P* then there is no need to run ALT and we can perform the MS allocation. If not we perform the following steps:

a) Choose the bidder from *B_P* that has the least "missing amount". If there is more than one, we pick the bidder with the most sensors requested
b) Check if there are bidders in *B_R* that want the same sensors (at least one) requested with the bidder selected from *B_P*. Include these in the subsidizers list as they have the incentive to subsidize this user and lowering the equal shares to be paid by all.
c) If no subsidizers with at least one common sensor can be found then terminate subsidization and go to 5 (try the next poor bidder). An alternative here would be to consider that users with no common sensors can subsidize him in a more extreme altruistic case.

Step 3. Subsidizing

After creating the subsidizers list we:

a) Check if the poor user can be subsidized i.e. if his remaining amount can be covered by all subsidizers. If not we go to step 5 and remove this bidder from the *B_P* list and find the next.

b) Divide the remaining amount for the bidder to be subsidized amongst the subsidizers in the list.

This is done in an egalitarian way as in Scenario 2 presented in 4.3.3. However, this might not be the fairest way as they should pay a contribution depending on how many common sensors they have.

c) Check if the equal subsidization share can be paid by all. If not we take the most out of the ones who cannot pay their subsidization share, and divide amongst the rest the remaining amount until we can cover the remaining amount of our poor bidder.

Step 4. Updating sets

a) Update the remaining budget for bidders in *B_R* that were used to subsidize the poor bidders

b) Include the poor bidder that has been subsidized to set *B_S(ubsidized)* and remove him from B_P. Now his remaining budget would be 0.

c) Clear the subsidizers list

Step 5. Iterate for the remaining poor bidders

Repeat steps 2-4 for all bidders in *B_P*. Stop either when there is no bidder left there or when we cannot subsidize the remaining ones.

Step 6. New share and allocations

For each bidder from *B_P* whom we were not able to find a subsidy, we kick them out. We also return the money (subsidy-shares) back to the subsidizers. We run again the typical MS routine, and repeat all steps 1-6 and continue until we have a set that all can pay. If we cannot find such a set, ALT has failed.

4.3 Elastic demand for multiples of the same basic types

We adopt for elastic demand the Definition 3. Hence in our model each elastic customer j is specified by his (finite) elastic demand for sensors $F(j) = \{S(j,1), S(j,2), \dots, S(j,m_j)\}$ and his associated vector of utilities $u(j) = \{u(j,1), u(j,2), \dots, u(j,m_j)\}$. We remind the reader that $S(j,k) \subset S(j,k+1), 1 \geq k \geq m_j$ and that $u(j,k)$ is the willingness to pay for the sensors in $S(j,k)$.

Our basic results specialize to the case where the demand has a very special structure. Namely it consists of multiples of the same basic sensor types. More precisely, let $A \subseteq I$ be this subset of basic sensor types (we call it the 'tuple' of sensors), and (A,k) denote the set of sensors $\{(i,l)|I \in A, 1 \leq l \leq k\}$, i.e., the k cheapest sensors from the basic types in A. Let $F = \{F(1), F(2), \dots\}$, where $F(k) = (A,k)$. We require that for each customer j, $F(j) = F(1), \dots, F(m_j)$. Hence the demand of different customer is defined over the same support set. Note that we could assume that the maximum number of sensors m_j per customer can be infinity, and set $u(j,k) = 0, k > m_j$.

For simplicity we assume that feasibility is not an issue in our mechanism description. Otherwise simply assume the cost of for sensors that are not available is an extremely large number (infinity). The cost $C(F(k))$ is the sum of the costs of sensors in the set $F(k)$. Since by increasing k we include more expensive (not strictly necessarily) sensors, it follows that $C(F(k))$ is increasing and also that $C(F(k)) - C(F(k-1)) \geq C(F(k-1)) - C(F(k-2))$.

The reason we consider this restricted elastic demand case is that the general case seems intractable, because to solve any optimization problem involving social welfare or utility under budget constraints involves the consideration of all possible combinations of demand sets, one per customer j, from his allowed family of sets $F = \{F(1), F(2), \dots\}$. Hence the problems are harder than the similar problems in the case of customers with inelastic demand. Restricting the sets $S(j,k)$ to be the same for all customers j simplifies the structure of the problem and leads to manageable solutions.

We discuss next the various mechanisms that we use to serve the customers.

4.3.1 Strategyproof mechanisms for cost balance

The application of the MS mechanism is not any more straightforward. The reason is that there one assumes that each set of customers has a unique cost for satisfying its resource requirements. In the case of elastic customers, each customer j corresponds to a set of potential 'sub-customers' each with a different resource requirement. Then, a potential set of customers that request the sensors corresponds to an element of the Cartesian product of the above sets. Here each customer j is represented by some sub-customer j_k corresponding to the demand of $F(k)$ with utility $u(j,k)$. Assuming that our initial elastic customers are the set $N = \{1, \dots, n\}$, a candidate set of customers to apply the MS mechanism is any vector $(s_1,\dots,s_n), s_j \in 1,2, \dots$, where s_j denotes the representative of customer j.

The application of the MS setting should consider all such combinations of such customers, and for each one find the stable set. Then choose the combination that is more efficient. Although this approach is valid in theory, it cannot be applied in practice because of its complexity. One could restrict these combinations to sub-customers of the same index k. i.e., run the mechanism when all customers ask for $F(1)$, when all customers ask for $F(2)$, etc, and choose the most efficient result. Unfortunately there is no guarantee that this is optimal. We show this by a simple counterexample.

Consider the case where our demand set consists of a single basic sensor type in quantities 1 and 2. Denote the set of sensors by $\{1,2\}$ There two bidders $N = \{1,2\}$. Both bidders are elastic and both are interested in getting $\{1\}$ or $\{1,2\}$. Assume that the costs are $c(1) = 2,\ c(2) = 2$ and the utilities are $u(1,1) = \varepsilon,\ u(1,2) = 10,\ u(2,1) = 1,\ u(2,2) = 1 + \varepsilon$. We like to check if N is stable according to MS with egalitarian allocation of costs, i.e., whether there is some allocation of sensors to customers in N which is stable sensor costs are shared equally. One can easily check that if we consider the allocation of the same sets of sensors to both customers (both customers get $\{1\}$ or $\{1,2\}$) we get empty stable sets, i.e., no market solution is found. But if we allocate $\{1,2\}$ to customer 1 and only $\{1\}$ to customers 2 makes N stable. Hence the socially optimal allocation requires in general arbitrary combinations of sensor subsets per customer.

There is an interesting remark we can make. To achieve a stable solution we may not allow access for a customer to sensors that will be turned on by other customers! In our example, customer 1 pays for both sensors 1, 2 to be turned on. But customer 2 is not allowed access to sensor 2 although this sensor is tuned on (by customer 1). The reason is that if we allow this, then customer 2 will need to pay his egalitarian share of the cost of sensor 2, which he cannot afford.

4.3.2 Strategyproof mechanisms for social efficiency

The first step before discussing such mechanisms is to define the social welfare maximization problem in the specific case of our elastic demand. An important property of this maximization problem is that once we decide for a set of sensors to be turned on, we don't lose by making them available to all customers since their utility functions are non-decreasing in the number of tuples they get. The optimization problem becomes

$$SW^*(N,u) = \max_{k \geq l}[\sum_{j \in N} u(j,k) - C(F(k))] \tag{24}$$

and we denote by k^* the value of k that achieves the maximum in (24). Under mild assumptions on the shape of the utility functions and the cost function[1] we can guarantee that there is always a unique solution in (24). The k^* may in general not be unique, but this is not important in our case since we can use any such value. The following also holds.

Property 2 *When maximizing SW, either no customer participates (case of $SW^*(N,u) \leq 0$ in (24)) or all customers participate (case of $SW^*(N,u) > 0$ in (24)).*

Applying now the VCG marginal cost mechanism is straightforward and follows the same lines as in Section 4.2.2. Each customer declares his utility function $\hat{u}(j) = \hat{u}(j,1), \ldots$. Then the mechanism solves for $SW^*(N,\hat{u})$ and computes the optimal number of sensors $k^*(\hat{u})$. If $SW^*(N,\hat{u}) > 0$ it asks from each customer a payment

$$x_j(\hat{u}) = \hat{u}_j(k^*(\hat{u}) - \left(SW^*(N,\hat{u}) - SW^*(N-j,\hat{u})\right). \tag{25}$$

One can observe that this payment is zero if the total SW to others is the same if *j* is or is not present. Hence a customer pays only if he is 'pivotal' and because of him at the optimum SW the value of k^* increases. He pays for the reduction of the social welfare (with him excluded) the system suffers at the new k^* (with him) compared to the old k^* (without him). In our case all the above quantities are easy to compute.

We can design an adaptive algorithm for balancing budget along the same lines as in Section 4.2.2.2.

4.3.3 Cooperative mechanisms for utility maximization

Because of the elastic demand we run into the same problems when trying to adapt an MS-like approach with altruism, such as the one introduced in Section 3.8 and adopted in the case of inelastic demand in Section 4.2.3. There is interesting extension that one can make in the case of elastic demand. The idea is to start by sharing the cost of the demand sets in the order $F(1), F(2), \ldots$ and stop when no more budget is available. Again customers are asked to pay for their equal share of the cost, but remain in the system if their budget (utility) is not enough; the customers that have more budget pay for covering the extra cost. The algorithm follows the following steps as in before.

We have the set $N = \{1, \ldots, n\}$ of customers. Our algorithm works in rounds. In each round we add a tuple of sensors and try to cover its cost. The algorithm in each round is the altruistic mechanism informally introduced in Section 3.8, with the difference that customers have an additional budget equal to the budget carried over from previous rounds. Observe that the cost

[1] For example assume that utility increments $u(j,k) - u(j,k-1)$ are not increasing and that cost increments $C(F(k)) - C(F(k-1))$ are not decreasing.

of adding the kth tuple is $C(F(k)) - C(F(k-1))$. The total budget of the *n* customers to spend for the first *k* tuples (i.e., for obtaining $F(k)$) is $\sum_{1 \leq j \leq n} u(j,k)$.

Let $b(j,k)$ and $p(j,k)$ be respectively the budget available at the beginning of round k and the total payment made during round k for customer j. In each round the customers share in an 'as much as egalitarian as possible' way the common cost (they share the same sensors). Some customers spend all their available budget (becoming 'poor' for this round) while some others have some budget left which is added to their budget of the next round. Note that in each round *k* customers renew their budget with the extra amount $u(j,k) - u(j,k-1) \geq 0$. The budget available (to spend) in round k by customer j is $u(j,k) - \sum_{1 \leq l \leq k-1} p(j,l)$.

During each round we partition *N* into two subsets, the 'poor' customers X, initially $X = \emptyset$, and the 'rich' customers Y, initially $Y = N$. Poor customers have no more budget to spend in this round in contrast to the rich that have some more. Our algorithm moves customers from Y to X as they exhaust their budget.

We start for simplicity with the first round.

1. Round 1. Initially $b(j,1) = u(j,1)$ and $p(j,k) = 0$.

   a) Step 1. Start with the set $F(1)$ and cost $C(1) = C(F(1))$. Divide equally $C(1)$ among the |*Y*| rich customers and ask each such customer to pay his share $s1 = C(1)/|Y|$.

   b) Step 2. A customer j that has less budget for his cost share (or he can marginally pay), i.e., $b(j,1) \leq s1$, spends all his budget and is moved to the poor customer list *X*.

   c) Step 3. A customer *j* that had strictly more budget to pay than his share, i.e., $b(j,1) > s1$, remains in the list Y, and pays his share $s1$. His remaining budget for round 1 is $b'(j,1) = b(j,1) - s1$.

   d) Step 4. We compute the uncovered cost $C'(1)$ that is the initial cost $C(1)$ minus the sum of the above payments. If this cost is zero, then this round is finished. Else $C'(1)$ must be covered now by the remaining rich customers in the list Y. We repeat the steps 1 - 4 until either all customers end up in the poor list and part of the cost remains not covered (in which case the whole process terminates and we don't try to do any further allocation of sensors), or the cost of $C(1)$ is fully covered. In this later case we repeat the procedure by moving to the next round.

   Let $\delta(j,1) = b(j,1) - p(j,1)$ be the leftover budget of customer *j* in list *Y* at the end of the round. The new budget for Round 2 is set to $b(j,2) = u(j,2) - u(j,1) + \delta(j,1)$. It corresponds to the extra utility for acquiring a second tuple of sensors plus the leftover budget from Round 1.

2. Round 2 (etc.). We repeat the logic of the previous round (Round 1) with the target to cover the additional cost of the next tuple of sensors that is $C(2) = C(F(2)) - C(F(1))$. If this round terminates successfully and the cost is covered, we repeat the procedure for covering the next cost $C(3) = C(F(3)) - C(F(2))$, etc. The procedure terminates at round k when the total utility of the customers is not sufficient to cover the cost of $C(F(k))$.

Our algorithm terminates if during some round we cannot cover the cost. This is justified because we assume that utility increments $u(j,k) - u(j,k-1)$ are non-increasing and that cost increments $C(F(k)) - C(F(k-1))$ are non-decreasing. Hence if at some round k total utility is less than total cost, for larger k this difference will become even larger. Alternatively, if we allow for arbitrary utility and cost functions, we may need to run the algorithm up to some maximum number of rounds (the range of definitions for the utility and the cost functions).

4.4 Nested inelastic demand

This case of inelastic demand is a special case of the general inelastic demand we analysed in 4.2. It corresponds to the case where the demand sets $S(j), j \in N$ of the customers are nested. Let us assume with no loss of generality that $F(1) \subseteq F(2) \subseteq \cdots \subseteq F(n)$. We discuss the same mechanisms as in the previous sections and investigate how the special structure of the demand can be taken into consideration.

For the design of strategyproof mechanisms for cost balance again the specific demand structure does not play a role. The application of the MS mechanisms with egalitarian cost sharing follows the lines of the description in Section 3.6. A similar comment holds for the extension that provides the long-run balancing of the budget by adding fixed fees.

In the maximization of the social welfare, the demand structure plays a role and allows for some interesting characterization of the optimum.

Proposition 5 *If* N^* *is the set of customers that maximizes social welfare, then if* $j \in N^*$ *we must also have* $j' \in N^*$ *for all* $j' < j$.

Corollary 6 *The social welfare optimization problem in the case of nested inelastic demand is*

$$\max_{1\le l\le n} \sum_{1\le j\le l} u(j) - C(F(l) \quad . \tag{26}$$

Proof: Suppose that $j \in N^*$ and $j' \notin N^*$ for some $j' < j$. Then we can strictly increase the total social welfare by adding customer *j*' since the sensors he requires are already available in the set $S(\mathrm{N}^*)$. Hence the set of customers $N^* + j'$ generates strictly higher social welfare and N^* cannot be optimal.

Using the above result allows for a simpler computational procedure to calculate the maximum social welfare and apply the marginal cost mechanism (see Section 4.2.2)

The problem of maximizing total utility under the constraint of balancing the cost is again simplified, as in the case of maximizing social welfare. One can prove using the same arguments that

Proposition 7 *If* N^* *is the set of customers that solves*

$$\max_{R\subseteq N} \sum_{j\in R} u(j) \; such \; that \sum_{j\in R} u(j) \ge C(R) \tag{27}$$

then if $j \in N^*$ *we must also have* $j' \in N^*$ *for all* $j' < j$.

Corollary 8 *The utility optimization problem in (27) becomes*

$$\max_{1 \le l \le n} \sum_{1 \le j \le l} u(j) \text{ such that } \sum_{1 \le j \le l} u(j) \ge C(F(l)) \ . \qquad (28)$$

We can use the above results in combination with the ideas introduced in Section 3.8 for the design of a cooperative mechanism that maximizes utility while covering costs. We first compute the optimal set N^* of customer by solving (28). Then we use the altruistic mechanism introduced in Section 3.8 for sharing the costs when serving the customers in N^*.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

5. Conclusion

This deliverable presented the specification as well as a theoretical analysis and design of the Market service component and the associated incentive mechanisms that reside in the Economic Mechanisms Overlay component of the INCEPTION architecture.

Work in the following period will focus on the implementation and the evaluation of the described mechanisms. Evaluation will be carried out primarily by means of simulations (in MATLAB) as well as by means of small-scale proof-of-concept trials with the demo software implementation of the entire INCEPTION system. Therefore, the implementation of such mechanisms will progress in both directions, accordingly.

In detail, the future work in WP3 of INCEPTION will include the respective research tasks:

1. Development of the three mechanisms described in Section 4 for the particular setting belonging in three different categories:
    a. The Moulin Shenker mechanism belonging in the Strategyproof mechanisms for cost balance category.
    b. The Marginal Cost Pricing (or VCG-like) mechanism belonging in the Strategyproof mechanisms for social efficiency maximisation category.
    c. The Altruistic mechanism belonging in the Cooperative mechanisms that maximise utility from sensor usage category.
2. Extensive evaluation by means of simulations (in MATLAB) of the developed mechanisms under three selected scenarios: general inelastic demand, nested elastic demand and nested inelastic demand and particular Key Performance Indicators investigating issues like cost balance, social welfare maximization and sustainability.
3. Optimisation and fine-tuning of the proposed mechanisms by designing and development of appropriate heuristics.

The results will be documented in deliverable D3.2: Implementation and Evaluation of the market service and incentive mechanisms.

Ευρωπαϊκή Ένωση
Ευρωπαϊκό Κοινωνικό Ταμείο
ΕΠΙΧΕΙΡΗΣΙΑΚΟ ΠΡΟΓΡΑΜΜΑ
ΕΚΠΑΙΔΕΥΣΗ ΚΑΙ ΔΙΑ ΒΙΟΥ ΜΑΘΗΣΗ
επένδυση στην κοινωνία της γνώσης
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ
ΕΣΠΑ 2007-2013
πρόγραμμα για την ανάπτυξη
ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ
Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης

6. References

[BN07] L. Blumrosen and N. Nisan. Combinatorial auctions. In N. Nisan, T. Roughgarden, E. Tardos, and V. Vazirani, editors, Algorithmic Game Theory, chapter 17, pages 267-299. Cambridge University Press, 2007.

[C71] E. H. Clarke. Multipart pricing of public goods. Public Choice, 11:17-33, 1971.

[CST06] P. Cramton, Y. Shoham, R. Steinberg. *Combinatorial Auctions*, MIT Press, 2006.

[DLA05] G. Danezis, S. Lewis, and R. Anderson, "How Much is Location Privacy Worth?", in WEIS '05:Proceedings of the Fourth Workshop on the Economics of Information Security, Harvard University, 2005.

[G73] T. Groves. Incentives in Teams, *Econometrica,* 41, 617-631, 1973.

[GL79a] Green, Jerry and Jean-Jacques Laffont. "On Coalition Incentive Compatibility." Review of Economic Studies, 46, 243–254, 1979

[GL79b] Green, Jerry and Jean-Jacques Laffont. "Satisfactory Mechanisms for Environments with Consumption Lower Bounds." Journal of Economic Theory, 19, 359–375, 1979

[K13] I. Koutsopoulos, "Optimal Incentive-driven design of participatory sensing systems", to appear, Proceedings of IEEE INFOCOM 2013.

[LH10] Juong-Sik Lee, Baik Hoh, "Dynamic pricing incentive for participatory sensing", Pervasive and Mobile Computing 6(6): 693-708, 2010.

[M99] H. Moulin. "Incremental cost sharing: Characterization by coalition strategy-proofness". *Social Choice and Welfare*, 16:279--320, 1999.

[MS01] H. Moulin and S. Shenker. Strategyproof sharing of submodular costs: Budget balance vs. efficiency. Econ. Theory, 18:51-533, 2001.

[NRTV07] N. Nisan, T. Roughgarden, E. Tardos, V. Vazirani (editors). *Algorithmic Game Theory*, Cambridge University Press, 2006.

[P09] Carlota Perez, Technological revolutions and techno-economic paradigms, Cambridge Journal of Economics, Vol. 34, No.1, January 2009, pp. 185-202

[V61] W. Vickrey. Counterspeculation, auctions, and competitive sealed tenders. Journal of Finance, 16(1):8-37, Mar 1961.