The invention concerns a system for interaction of at least two loosely coupled independently acting networked objects for collaboration in a network with an environment of dynamically changing and possibly multiple ownerships to which an object is connected and is outside the control of the owner wherein each object includes at least one processor and a memory comprising an API to be executed by said processor allowing said object be able to communicate through a wireless or wire link to some sort of internet or internet network and to interact each other, said system using a Uniform Resource Indicator (URI) to identify objects within a "global" namespace including many local namespaces, said system being characterized in that in it includes a minimal basic set of mechanisms for ownership coordination and or transfer, said minimal basic set of mechanisms consisting of interface methods and or globally standardized rules.

**Figure 1**
**System of dynamically managed/self-organizing object networks**

**FIELD OF INVENTION**

The present invention relates to the field of management of devices communicating or sharing information with each other and composing an infrastructure such as for example a manufacturing plant (energy plant, for instance) or Internet of Things (IoT) and the configuration of said infrastructure, more precisely the invention concerns a system for managing and/or governing dynamically changing ownerships of objects in a network of interacting objects.

**BACKGROUND OF THE INVENTION**

The Internet of Things (IoT) is a system of interrelated computing devices, mechanical and digital machines, objects, animals or people that are provided with unique identifiers and the ability to transfer data over a network without requiring human-to-human or human-to-computer interaction. A thing, in the Internet of Things, can be a person with a heart monitor implant, a farm animal with a biochip transponder, an automobile that has built-in sensors to alert the driver when tire pressure is low -- or any other natural or man-made object that can be assigned a communication address (for example and without limitation an IP address) and provided with the ability to transfer data over a network.

Connected devices in the Internet of Things, for example, are currently limited in usability and marketing of created value by a very simple concept of ownership. As soon as the "physical owner" of a thing (the one who has physical access to it and connects it to the internet) connects it to the Internet (through WiFi or wire based) and a cloud based or privately
based service, control of the device and it's transported data and economic value is dramatically dropping. The system or service, the device is connected to is outside the control of the owner of the device. Data control can be managed by the Provider of the device/service without transparency and influence of the "Physical Owner" of the respective device(s). The Apps and GUls provided by the server to handle the device(s) (e.g. switching on/off lamps) suggest, that the user has control over the device, but as there may be any other interfaces and technical means to execute control over the device, there cannot be reliability as to what the data/devices are used for. If the central server/service the device is connected to goes out of service, this renders the device pretty useless as the larger part of the Internet of Things service is no longer existing, and the devices cannot be "reallocated" to a different service/provider. In a digitized environment there is high value of not only a simple thing/cloud relationship but of a complex network of relationship supporting different purposes at the same time. For example, the thing/object might be related to one cloud for data management (e.g. Fitbit for Health Data). For maintenance/repair/operating the same thing might be related to a different cloud/organization (e.g. Managed Service Provider), for Charging/Billing/Taxes, the same thing might need to maintain a relationship to a third organization. During the lifetime of the object/thing these relationships may need to change (transfer of ownership, changing contracts etc.). This requires to managing these relationships in a coordinated way handling dynamically changing multiple ownerships.

Usability of Internet of Things devices in the context of cloud-based services are either:

- wired to the respective cloud services (e.g. the service it connects to is coded into the device and cannot be changed by the "Physical Owner" of it (e.g. Sonoff (Brand Name) "Intelligent Wall Sockets").
- can be setup handling the device directly to setup the target server (e.g. by Built in Webserver to change persistent
parameters), which requires coding/parametrization and extensive knowledge about the technology (not end user orientated).

In the first case, end of life of the central server renders the device connected but useless and all data produced during the lifetime of it is lost.

In the second case, when considering a huge number of devices (e.g. hundreds/thousands of Things) reallocating these devices to a new "logical owner" or cloud is a tedious and error prone activity (even if the devices are physically accessible, i.e. not incorporated in systems or plants).

Especially the namespaces for things in the cloud are not centrally managed, so the data points of multiple devices may overlap and thus render reliable analysis/control useless (multiple devices use the same descriptor for different data sources thus resulting in ambiguous behavior)

Currently there are no known concepts/solutions where IoT devices can at the same time be owned by different "Logical Owners" (e.g. for a "Fitness Device" the "Physical Owner" of it and a Doctor to supervise medical health). If there are means to interact, this is handled in a non-transparent mechanism within the cloud and not under control of the "Physical Owner" (lack of governance).

SUMMARY OF THE INVENTION

The present invention has as its object to obviate certain drawback of the prior art by proposing an infrastructure for managing and/or governing dynamically changing ownerships of objects.

This goal is achieved by a system for interaction of at least two loosely coupled independently acting networked objects for collaboration in a network with an environment of dynamically changing and possibly multiple ownerships to which an object is connected and is outside the control of the owner wherein each object includes at least one processor and a memory comprising an API to be executed by said processor allowing said object be able to
communicate through a wireless or wire link to some sort of intranet or internet network and to interact each other, said system using a Uniform Resource Indicator (URI) to identify objects within a “global” namespace including many local namespaces, said system being characterized in that it includes a minimal basic set of mechanisms for ownership coordination and or transfer, said minimal basic set of mechanisms consisting of interface methods and or globally standardized rules for:

- remapping local and intermediate namespaces dynamically into a global namespace;
- describing/offering the service of the object in a self-descriptive way like JSON or ontologies;
- making the description/offering available for perception by another object on a generally agreed, well defined place in the global namespace;
- withdrawing the visibility from the currently defined place;
- presenting the services, i.e. actively make the services known according to the description known to possibly interested parties;
- applying for deployment on the network within the context of one or many owners to offer services embedded on information contained within each object;
- refining the service offerings, qualities, constraints and side conditions of the ownership relation dynamically
- hiring an object into the context of one or more larger object(s) to govern accessibility and services within this larger object, as well as concurrently acting on behalf of multiple organizations like a freelance contractor;
- accepting the hiring contract and locating the resource into the organization to make use of it and including the resource into the organizational context.
According to another feature, the refining functionality enables to negotiate duration of contract, exclusive or shared ownership, compensation/cost for deliverables and enabling negotiation of contracts like in a free market exchange.

According to another feature, the system or each object comprises an arrangement to trigger manually or automatically the describing/offering method.

According to another feature, the triggering arrangement is a physical button or a voice command or a visual trigger or a face recognition means.

According to another feature, each object of the system comprises an arrangement for deriving information about the published object so as to make decisions to connect to the said published object.

According to another feature, each object comprises an arrangement or program to check the descriptions against their goals/embedded rules not to violate possible existing "hiring contracts" with other objects.

According to another feature, cooperating objects comprise at least an arrangement to manage a part of the “global” namespace, the “intermediate” namespace and the “local” namespace in an aligned and coordinate way so as to get a joint “intermediate” namespace between objects/things.

According to another feature, objects interact with other objects either through function calls or by publishing messages to an URI in a joint namespace, said objects using a publishing/subscribe method or function calls approach.
According to another feature, the system comprises further methods for enhancing the minimalistic set of methods, said further methods comprising at least one of:

- implementing identity/resource management within an object such as handling over and using references/IDs access credentials to other objects that are needed to fulfill the agreed functionality of the resource like scheduling, deployment;
- implementing commercial management within an object such as budget planning/forecast, payments/value transfers, charging/billing, payroll, taxes.

According to another feature, the objects using the enhanced set of methods are configured to dynamically transfer important context to adapt to varying situations that allow said objects to behave different in different situations within the constraints of contracts that may be pre-negotiated or dynamically negotiated by the objects based on need and situation.

According to another feature, the objects using the enhanced set of methods are configured to have bookkeeping functionalities to selectively apply these contexts to the specific interactions with specific objects.

Another goal of the present invention is to provide an architecture allowing collaboration in a network environment for dynamically changing ownerships.

This goal is obtained by an object for interaction of loosely coupled independently acting networked objects for collaboration in a network environment for dynamically changing ownerships which is connected and configured for managing logical ownership on behalf of the physical owner said object including at least a processor using an operating system and a
memory comprising an API executed by the processor and communication means through a wireless or wire link to some sort of intranet or internet network to be able to interact with other objects, the object being characterized in that said API comprises a manager service for implementing at least one of the public or external methods “present”, “apply”, “hire”, “refine” for presenting, applying, hiring, refining data and giving access to the manifest of said object as well as providing processing functionality, said methods and manifest being contained in the memory of the object publicly accessible, the manager service communicating on one side with private or internal methods “register”, “unregister” and “remap” memorized in a first private memory of the object for respectively registering/unregistering and or remapping, and on another side with private variables memorized in a second private memory of the object for saving information concerning at least local root node as the root for its local namespace.

According to another feature, the “present” method of a superior object is called by an inferior object to offer it’s services by making the superior node aware of the current location in the global namespace and by providing the formal description (manifest) of the offered services and the top node of the local namespace.

According to another feature, the “apply” method of an inferior object is called by the superior object to express the intent to temporarily (probation/qualification period) incorporate it into the superior’s ownership/namespace, make the inferior knowledgeable about the services the superior is offering (manifest) and the “temporary” place in the intermediate namespace the inferior is to use during the probation/qualification period, allowing the inferior object to take decision if this transfer is compatible or not with the current state.
According to another feature, the “hire” method of an inferior object is called by the superior object to express the intent to permanently incorporate it into the superior’s ownership/namespace, make the inferior knowledgeable about the services the superior is offering (manifest) and the place in the intermediate namespace the inferior is later on to be working with, allowing the inferior object to take decision if this transfer is compatible or not with the current state.

According to another feature, the “refine” method of an inferior is called by a superior to make it aware of:

• changed situation (constraints);
• changed employment/ownership conditions being a means to “renegotiate” the ownership contract (like in a free market situation, where a person applies for many jobs at the same time) and to decide for the best terms and conditions among a set of possible terms and conditions;
• a transfer of ownership completing after one or more exchanges of “apply”-“refine” iterations ending with a successful “hire” method call.

According to another feature, the private or internal method “register” is used during initial object creation to include the object into the global namespace or even within a well-defined “superior” intermediate namespace, the private or internal method “unregister” being used during transfer of ownership to remove the respective object’s local namespace from the “superior” intermediate namespace.

According to another feature, the private or internal method “remap” consist in the transformation of the local namespace into the relevant intermediate namespace of the new owner, all instances of the “local”
namespace being changed by prepending prefixes describing the "intermediate" namespace to the respective names thus renaming all items to perfectly fit into the - now common - namespace of both objects, all names of the object being visible/accessible for other objects.

According to another feature, the object includes a "resource" service which provides at least one of the following methods:

- `getIdentity` which allows to get a unique identity within the owner object;
- "getAccess" which allows to use resources of the owner without knowing where they are.

According another feature, each object of the system comprises a mechanism or a program for "transformation of names" which modify the own "names" (with no meaning outside the object) into "name" that have a meaning for all other objects in the global namespace.

BRIEF DESCRIPTION OF THE DRAWINGS

Other features and advantages of the present invention will appear more clearly upon reading the following description, given with reference to the appended drawings, in which:

- figure 1 is a schematic representation of the structure of an object of the system, according one embodiment,
- figure 2 is a schematic representation of the structure of an object of the system, according another embodiment,
- figure 3 is a schematic representation of the initial state of a multi-level dynamic namespaces flexible relationship, according one embodiment
- figure 4 is a schematic representation of simple connected state of a multi-level dynamic namespaces flexible relationship, according one embodiment;
- figure 5 is a schematic representation of a multiple connected state of a multi-level dynamic namespaces flexibles relationship, according one embodiment;
- figure 6 is a schematic representation of a “market” node in a multi-level dynamic namespaces flexible relationship, according one embodiment;
- figure 7 is a schematic representation of the transfer of ownership from “market” node to “employer” node in a multi-level dynamic namespaces flexibles relationship according one embodiment;
- figure 8 is a schematic representation of a two level static namespaces hardwired relationships. The object (bracelet) is hardwired to one single node in the cloud of service provider, according an embodiment;
- figure 9 is a schematic representation of interacting objects working on shared/managed context, according one embodiment
- figure 10 is a schematic representation of interacting objects working on shared/managed context, according one embodiment

DETAILED DESCRIPTION

The present invention concerns a system for dynamic interaction of loosely coupled independently acting networked objects (1).

In some embodiments, the system for interaction of at least two loosely coupled independently acting networked objects (1) for collaboration in a network with an environment of dynamically changing and possibly multiple ownerships to which an object (1) is connected and is outside the control of the owner, comprises at least a set of objects (1) and at least a network/communication means or arrangement for communication between said objects (1). Each object (1) includes at least one processor and a
memory comprising an API to be executed by said processor allowing said object (1) be able to communicate through a wireless (for example and without limitation WLAN/Bluetooth/WIFI) or wire (for example and without limitation Ethernet) link to some sort of intranet or internet network and to interact each other (the interaction may either be between objects (1) directly (peer to peer) or networked (across multiple intermediate objects (1), but even in this case the direct peers would only see peer to peer connections), said system using a Uniform Resource Indicator (URI) to identify objects (1) within a "global" namespace including many local namespaces (2c), said system being characterized in that it includes a minimal basic set of mechanisms for ownership coordination and or transfer, said minimal basic set of mechanisms consisting of interface methods and or globally standardized rules for:

- remapping local (2c) and intermediate (2b) namespaces dynamically into a global namespace (2a);
- describing/offering the service of the object (1) in a self-descriptive way like JSON or ontologies;
- making the description/offering available for perception by other object (1) (passively) on a generally agreed, well defined place in the global namespace (2a) (like a white board or market). For this activity - the first appearance in the namespace - it is recommended to use common mechanisms to mark the object (1) with a unique ID to make it distinguishable from other objects (1) also registered there;
- withdrawing the visibility from the currently defined place (e.g. when no longer available via the previous namespace);
- presenting the services, i.e. actively make the services known according to the description known to possibly interested parties
- applying for deployment on the network within the context of one or many owners to offer services embedded on information contained within each object (1).
• refining the service offerings, qualities, constraints and side conditions of the ownership relation dynamically
• hiring an object (1) into the context of one or more larger object(s) (1) (like a worker with a given set of skills into an organization that makes use of it) to govern accessibility and services within this larger object (1), as well as concurrently acting on behalf of multiple organizations like a freelance contractor;
• accepting the hiring contract and locating the resource into the organization to make use of it and including the resource into the organizational context (by using related IDs, getting access to and granting access to resources owned by the hiring organization).

By object (1), we mean a physical “thing” or device that may be part of the Internet of Things (IoT) or a manufacturing system or infrastructure comprising devices communicating, through a network, with each other and with devices external to said manufacturing system belonging to another infrastructure or entity.

Unlike physical things or devices that can be on one place only and handled only by one person at the same time, the "logical things" of a physical “thing” or object (1) (for example, its state, its functionalities, etc.) can at the same time be controlled by several entities (logical, legal, natural ones).

A user of the IoT, may "own" the “logical things” of an object (1) or device because it can send input to it, change its state (represented by data internal to the object (1) even to the extreme to make the object (1) unusable or worthless for others), for example and without limitation the locking/unlocking of a smartphone or deleting its data by means of a program running on a remote device such as a computer or a server communicating with said smartphone. In an "exclusive" case, this would only have consequences for the "one and only" owner of the object (1) or device. In a
"shared" sense (concurrent ownership at the same time), an activity like this would also technically as well as legally impact other entities or owners (natural/legal persons) that also have some claims for rights of use for the same object (1) or device.

By namespace, we mean a set of symbols used to organize objects (1) of various kinds, so that said objects (1) are referred to by name. It may be seen as a dictionary mapping names to objects (1). A namespace allows to uniquely qualify objects (1). Then, one knows to which domain of definition a given object (1) is related and how it must be interpreted according to its specification.

The namespace specific to an object (1) is called a local namespace (2c) and the namespace including all the local namespaces (2c) is called a global namespace (2a).

For remapping of namespaces all interacting objects (1) need to understand how the global namespace (2a) is structured and to which place (URI) they need to map their local namespace (2c) to. In some embodiments, the system may comprise a program for defining a convention for structure allowing each object (1) to determine the structure of the global namespace (2a).

In some embodiments, the system or each object of the system may comprise a mechanism or a program for "transformation of names" which modify the own "names" (with no meaning outside the object (1)) into "name" that have a meaning for all other objects (1) in the global namespace (2a).

The system or object (1) may comprise an arrangement to trigger manually or automatically the describing/offering method. For example, and without limitation, said arrangement may be a physical button, voice command, visual trigger, face recognition or even some data collected to make the object (1) valuable for some "hirer".
The describing/offering requires at least:

- a formal description in a machine-readable form initially within the object (1) itself (not visible to the outside world)
- making this description visible to the outside world. This may be done by publishing this description to a globally visible place (e.g. a namespace or to a globally visible repository/registry or database), where it can be read and analyzed by other objects (1).
- the objects (1) that read the description need to be able to derive information about the published object (1) to make decisions (by coded logic or by intelligence) to connect to the said object (1).

In some embodiments, each object (1) of the system comprises an arrangement for deriving information about the published object (1) so as to make decisions (by coded logic or by intelligence) to connect to the said published object (1).

Applying requires rules (logical or intelligence) implemented in the interacting objects (1) to take adequate actions to keep semantic integrity in the situation that in this case the "applying" object (1) is not yet "hired" (owned by the hiring object (1)) but no longer "unbound" (like "quarantine" or "probation/qualification period"). The constraints of the "apply" phase are described in the description of the service offering to guide these decisions.

Refining is the digital equivalent to negotiating and closing a contract. It starts with an offer, followed by a counter offer and ends when both parties agree on the terms and conditions and close the deal.

In order to set up the refining phase the system may comprise at least a computing architecture to execute a set of rules for:

- read and understand the descriptions;
• check the descriptions against predefined criteria (like compliance to standards);
• compare the outcome against own goals (functional or other);
• draw conclusions about how to modify the own description or suggest modifications to the description of the other object (1);
• modify said descriptions;
• make these available to the "other" object (1) for checks;

The refinement phase ends either with hiring (descriptions of both objects (1) are aligned and agreeable), with "apply" (agree with limitations of the contract and the opportunity to get to the hiring state) or with no longer communicating with each other (terminating communication or negotiations).

The hiring phase is the act of accepting the services of the applying object (1), communicate the "place" where to integrate/map into the hiring object (1)'s structure (like position or department for humans), grant access to resources of the hiring object (1) (for example, time, money, space, rights).

For the "hired" object (1) this means that it binds its services to the hiring object (1), and to adjust its own logic/working principles to make use of the environment of the hiring object (1).

In some embodiments, the object (1) may comprise an arrangement or program to check the descriptions against their goals/embedded rules not to violate possible existing "hiring contracts" with other objects (1). For example, which services to expose, which data of other contracts to disclose or keep secret. This is like systems working in a "multitenancy mode" sharing some resources/data and strictly separation for others.

In some embodiment, the objects (1) may incorporate commonly used computer functionality like "identity (ID) management systems"
(managing unique ids and allocating rights and roles to ids) like LDAP or 
active directory.

The difficulty does not lie (or to a smaller degree) in implementing 
this functionality but to offer and handle it a generic, self-descriptive way so 
that all interacting objects (1) may seamlessly use and offer this. It is like 
making all human resource (HR) organizations within a set of companies all 
look alike that persons moving between companies can easily identify what 
to do and always act the same.

One common function within IoT is to keep communication secret.

So, one anticipated function is to ask, grant and withdraw crypto keys to 
related objects (1) to do that.

Another one is to utilize resources that are owned by the hiring object 
(1) (money, information, space). As these resources tend to be protected 
against unauthorized use, authorization needs to be granted as needed. This 
requires methods for granting and withdrawing authorization to use 
resources.

In some embodiments, the refining functionality enables to negotiate 
duration of contract, exclusive or shared ownership, compensation/cost for 
deliverables and enabling negotiation of contracts like in a free market 
exchange.

In some embodiments, the “global” namespace is managed outside 
of the object (1)/thing and the “local” namespace is managed within the 
object (1)/thing.

In some embodiments, cooperating objects (1) manage a part of the 
“global” namespace, the “intermediate” namespace and the “local” 
namespace in an aligned and coordinate way to get a joint “intermediate” 
namespace between objects (1)/things. The combination of the namespaces 
(intermediate and local for forming the joint intermediate namespace (2b)) 
enables a dynamic cooperation that is completely managed by both
namespace (the local and the intermediate namespaces (2b)) and thus forms the foundation for cooperation. This also implies “decentralized” namespace management (no central control is necessary).

In some embodiments, objects (1) may interact with other objects (1) either through function calls or by publishing messages to an URI in a joint namespace, said objects (1) using a publishing/subscribe method or function calls approach. Each object (1) may comprise at least an arrangement for subscribing to said URI in said joint namespace and make itself visible and accessible for such published messages.

In some embodiments, the system comprises further methods for enhancing the minimalistic set of methods, said further methods comprising at least one of the following methods (that may be common to organizations and companies like):

• implementing identity/resource management within an object (1) such as handling over and using references/IDs access credentials to other objects (1) that are needed to fulfill the agreed functionality of the resource like scheduling, deployment etc. The purpose is to give and take data between acting object (1), that enable using of resources in a shared way.
• implementing commercial management within an object (1) such as budget planning/forecast, payments/value transfers, charging/billing, payroll, taxes etc.

The enhancement of the minimalistic set of methods allows a higher degree of sharing functionality and resource between a set of object (1).

In some embodiments, the object (1) using the methods are able to dynamically transfer important context to adapt to varying situations that
allow the object (1) to behave differently in different situations within the constraints of contracts that may be pre-negotiated (commonly shared by preexisting context) or dynamically negotiated by the object (1) based on need and situation.

In some embodiments, the object (1) using the methods are able to have bookkeeping capabilities to selectively apply these contexts to the specific interactions with specific object (1) (e.g. apply Identity A/credentials of A for interacting with object (1) A, while applying Identity B/Credentials of B for interacting with B).

Figure 8 illustrates an example of a situation without the application of the system of the present invention. When an IoT object (1) or device like a connected health bracelet (or wall socket or power meter) is first deployed, it is connected to the Internet (state of the art) and associated to a "logical owner" (a person making use of the object (1) from remote). Currently it is by coding related to the cloud of the vendor of the device. There is one "global namespace (2a)" through which the device's local namespace (2c) and related methods/variables can be addressed. The mapping is static.

Before any connection/interaction can take place, the object (1) in question needs to be purchased from some vendor, brought to the planned destination place and connected to e.g. a WIFI network.

The (in our case "natural") person doing that needs to have control over the object (1) (fetch it, move it, plug it). So, the person is not only "logical owner" but even "physical owner" of the object (1). In a later situation, where no physical access is needed any more (when the thing is connected), the person will be only a "logical owner" (making use of the object (1) from remote).

The system of the present invention introduces a "Multi-Level namespace" that may be controlled by different instances. As illustrated in Figure 3, there is a "global namespace (2a)" as in Figure 8, which is shared
by all instances that apply the concept of the invention by a shared convention/understanding. Then, there is the “local namespace (2c)” as in Figure 8 managed by the object (1). Additionally, there are one or many “intermediate namespaces (2b)” managed by one of many object (1) (represented as “superior node”, as it is hierarchically above the incorporated/inferior node(s)), that later on will incorporate functionality and namespaces of other objects (1) by exercising “ownership” of them. In Figure 3, the “top level node” of the local namespace (2c) has no relation to the global namespace (2a) and thus is not addressable/useable. Despite that is has its local namespace (2c) completely controllable by itself.

The “Shared convention or understanding” in this statement means that all parties involved need to understand how a location in the global namespace (2a) is “written”, for example and without limitation “www.atos.net” as the location where all “Atos-Stuff that is provided and used by Atos is located. This enables all interested parties to find Atos stuff from wherever they come.

Figure 4 corresponds to an embodiment showing the state of the overall system, when the inferior node (top level node of the local namespace (2c)) is already “owned” (incorporated) by the superior node and part of the “intermediate (2b)” and the “global namespace (2a)”. This enables the object (1) to be used/addressed in the shared context of the superior nodes namespace. The “incorporation process” is implemented by a set of methods both of the superior node as well as the “inferior node”. Actually, at different points in time and for different sets of objects (1), the roles may also change (a superior may become an inferior of its former inferior or another objects (1)). So, all objects (1) have the same rules implemented.

“Incorporation Process” is a synonym for “transfer of ownership” from “external” to “internal”. So at the end the object (1) is within the other object (1).
Figure 5 corresponds to an embodiment showing the state of the overall system, where one “inferior object (1) (for example top level node of the local namespace)” is at the same time incorporated into two different superior objects (1) (for example superior nodes). This means that the behavior and data of it can be addressed via the intermediate namespaces (2b) of two superior objects (1). Therefore, it is owned by two superiors and from a pure logical point of view the inferior object (1) exists in two places. To make it clear, the inferior object (1) instance (state, data, logic) is existing exactly once. But, the object (1) may decide to behave differently when addressed from the context of different superiors (“overloaded” operation). For example, and without limitation, the object (1) “lamp” may turn on with full light if activated by a button 1 and with dimmed light when activated by a button 2.

By “Incorporated into two different superior nodes”, we mean that the “subordinate node” belongs to (is owned by) two other nodes (for example and without limitation a law office may concurrently work for two different companies at the same time and accept orders from both, even if the law office is legally not part of the customer company).

Figure 6 introduces the concept of a “well-defined” superior node in the global namespace (2a), the so called “Market”, according an embodiment. “Market” is just a name for an entry point for all objects (1) that are currently not “owned” by others (like a free-market exchange where anybody can offer and hire). This is a means to make otherwise “unconnected” or “unowned” objects (1) addressable, accessible and visible for other objects (1) that intend to take ownership. As all objects (1) there also expose their “manifest”, this also take the role of an explicit inventory of all functionalities available in the global namespace (the capabilities of each object (1) in a machine-readable way).
Figure 7 shows the handover of ownership of an inferior object (1) from one superior to another. In this case from the “Market” to the “employer”. But the mechanism is the same for transfer from one employer to the other. The inferior is addressable and visible through the intermediate namespace (2b) of its current superior (here the “Market”).

The Employer initiates a transfer sequence with the object (1) based on the minimal basic set of methods that is implemented by all objects (1). On completion of the transfer, the inferior object (1) is removed from the “old” superior’s namespace and incorporated into the “new” superior’s namespace.

The present invention also concerns an object (1)

In some embodiments, object (1) for interaction of loosely coupled independently acting networked objects (1) for collaboration in a network environment for dynamically changing ownerships which is connected and is capable of managing logical ownership on behalf of the of the (physical) owner (the person having physical control over the object (1) may inhibit any transfer of logical ownership by just disconnecting or powering down said object (1)) characterized each object (1) includes at least a processor using an operating system, for example and without limitation a Java operating system or an embedded operating system, and a memory comprising an API executed by the processor and communication means through a wireless or wire link to some sort of intranet or internet network to be able to interact with other objects (1). said API comprising a manager service (10) for implementing at least one of the public or external (exposed to and callable from other object (1)) methods “present”, “apply”, “hire” and “refine” for presenting, applying, hiring, refining data and, for giving access to the manifest of said object (1) (the manifest is the formal description of the services offered by an object (1)) as well as providing processing functionality, said methods and manifest being contained in the memory of the object (1) publicly accessible. The manager service (10) communicates
on one side with private or internal (not callable/useable from outside the
object (1)) methods “register”, “unregister” and “remap”, memorized in a first
private memory (11) of the object (1), for respectively registering/unregistering
and or remapping, and on another side with private
variables memorized in a second private memory (12) of the object (1) for
saving information concerning at least local root node as the root for its local
namespace (2c).

Figure 1 illustrates the basic mechanism embedded in all objects (1)
of the system, that is necessary to implement the above-mentioned
scenarios of transfer of ownership and to map the inferior’s local namespace
(2c) into the superior’s and thus the global namespace (2a). This is
accomplished by providing a basic “manager service (10)” that offers a well-defined set of “methods” as well as by an internal set of methods (not accessible via the namespace, i.e. local or “private” methods). The latter is
executed internally, when some public methods are executed. Additionally,
there is a data object (1) exposed to the namespace that contains a
machine-readable description of the services the objects (1) offer to other
objects (1). This data object (1) is called “manifest” and may be implemented
using for example, and without limitation, JSON, Ontologies or other means.
This manifest at least describes the mandatory service “manager”.

In some embodiments, the “present” method of a superior object (1)
is called by an inferior object (1) to offer it’s services by making the superior
node aware of the current location in the global namespace (e.g. on the
Market intermediate space) and by providing the formal description
(Manifest) of the offered services and the Top node of the local namespace;

In some embodiments, the “apply” method of an inferior object (1) is
called by the superior object (1) to express the intent to temporarily
(probation/qualification period) incorporate it into the superior’s
ownership/namespace. The “apply” method also makes the Inferior knowledgeable about the services the superior is offering (manifest) and the “temporary” place in the intermediate namespace (2b) the Inferior is to use during the probation/qualification period, allowing the Inferior object (1) to take decision (through, for instance, human decision or artificial intelligence rules) if this transfer is compatible or not with the current state (e.g. other existing ownerships).

In some embodiments, the “hire” method of an inferior object (1) is called by the superior object (1) to express the intent to permanently incorporate it into the superior’s ownership/namespace. The “hire” method also makes the inferior knowledgeable about the services the superior is offering (Manifest) and the place in the intermediate namespace (2b) the inferior is later on to be working with, allowing the inferior object (1) to take decision (through, for instance, human decision or artificial intelligence rules) if this transfer is compatible or not with the current state (e.g. other existing ownerships);

The inferior object (1) may also decide not to perform the hire activity but to apply with a different offering/Manifest to optimize interaction. Indeed, in some embodiments, the object (1) may comprise a program or a set of rules to be implemented depending on certain facts.

In some embodiments, the “refine” method of an inferior is called by a superior to make it aware of:

- changed situation (constraints) ;
- changed employment/ownership conditions being a means to “renegotiate” the ownership contract (like in a free market situation, where a person applies for many jobs at the same
time) and to decide for the best terms and conditions among a set of possible terms and conditions;
- a transfer of ownership completing after one or more exchanges of “applyTrefine” iterations ending with a successful “hire” method call.

Figure 2 shows an object (1) with additional functionalities for self-management and cooperation in larger dynamic contexts, according an embodiment. Basic principle is to have objects (1) behave like intelligent, self-optimizing entities like Companies/Organizations that have higher structural capabilities and working principles. It makes every object (1) behave like a Company on its own that cares for planning, keeping track on cost, created values, expenses and resources it uses and creates and increases value all the time. For that, design patterns for organizational behavior may be implemented within the object (1) and be executed via services interface like the one of the "manager service (10)". One of the basic principles used for organization it that of “Identity”. In enables objects (1) to act within a common context, where decisions/operations are based on rules that require acting objects (1) to impersonate specific identities to behave legally or logically correct. As a contract requires that negotiators to be legally entitled to form a binding contract by having a correct “legal identity” (e.g. being the CEO of a company), the interacting objects (1) will need the “correct identities” to operate as expected (like for purchasing goods or for financing expenses).

In some embodiments, private or internal method “register” is used during initial object (1) creation (instantiation) (create, for example, the state of the object (1) so far unknown outside of its own to make it visible for all other objects that have access to the global namespace. Therefore, the local knowledge (state, functionalities, etc.) is globally visible. To include the object (1) into the global namespace (2a) (preferably in the well-known “Market” intermediate namespace (2b)) or even within a defined “superior”
intermediate namespace (2b) (e.g. development of a new service/offering within an already existing organization/company). The private or internal method “unregister” is used during transfer of ownership to remove the respective object’s (1) local namespace (2c) from the “superior” intermediate namespace (2b).

In some embodiments, private or internal method “remap” consist in the transformation of the local namespace (2c) into the relevant intermediate namespace (2b) of the new owner, all instances of the “local” namespace (2c) being changed, in said transformation process, by prepending prefixes describing the “intermediate” namespace to the respective names thus renaming all items to perfectly fit into the - now common - namespace of both objects (1), all names of the object (1) being then visible /accessible for other objects (1) (in fact it is a sort of dynamic linking).

In some embodiments, an object (1) (extended), as illustrated in Fig. 8, includes in addition “resource” service (13) of the object (1) (extended.) provides the methods “getIdentity” and “getAccess” that are:

- getIdentity is a method to get a unique identity within the owner object (1) (like the personal ID a43... in a corporation authorizing many activities within a company are, this is a prerequisite to “act on behalf” of the owner);
- “getAccess” is a method to use resources of the owner without knowing where they are.

In the following, we present some example so as to capture the principle of the present invention. In the examples, an object (1) may not be a device, however it must be understood that the aim of said examples is just to illustrate how the system of the present invention works.
For example, and without limitation, within the shared eco-system of a “System for dynamically managed/self-organizing object (1) networks” may exist the objects (1) “Market”, “Manufacturing Company” and “Milling”. “Market” is a well-defined intermediate namespace (2b) within the global namespace (2a) of the eco-system.

Figure 9, illustrates the situation in which “Manufacturing Company” and “Milling” have published their existence to the “Market” and thus made themselves known and accessible to the eco-system. The object (1) “Manufacturing Company” offers services like “Management” (required by the minimum set of methods as explained in the invention) “Sales”, “Resource Control”, “Production” and “Financial” to represent the business (non-exhaustive).

The services “Sales”, “Resource Control”, “Production” or “Financial” are examples for typical services an organization may have. Each one may have methods associated that “do business”, for example in the case of the service “Financial” the method “grantBudget” or the method “approveExpenses”. These methods will in practice have a huge variety based on vendor/use case of the object (1). Due to the nature of Dynamic APIs (self-descriptive) the embedded object (1) may learn by analysis of this description of which methods are there and what they can do.

The object (1) “Milling”, represented by the services “Management” (required by the minimum set of methods of the invention), “Resource Control”, “Production” (incorporating the Resource “Milling Machine”) and “Financials” and the resource with “Milling” having given capabilities, history (machine cycles used, incidents experienced) and economic value (“owned by “Financials” like: onetime costs, monthly recurring costs, depreciation value, creditor) may be deployed by being allocated to a legal entity (e.g. a manufacturing company). The “Financials-Service” of the Milling machine in this example has code and data implemented to respond to questions like
“What is your current remaining asset value?” or “How many pieces have been produced in the past?” etc.

Both layouts of the services are represented in machine readable form “Manifest” at the “Market”. Each time, an object (1) is created, it registers itself at the only one position in the namespace that is “well-known” (predefined by convention), the “Market”. There it is visible with its “Manifest” (machine readable description of functionality (services and methods)). By this it is prepared for entering contracts (by hire).

In the initial state “Manufacturing Company” and “Milling” are not related in any kind.

In Figure 10, by scanning “Market” and reading the set of “Manifests” there, the “Manufacturing Company” identifies “Milling” as a possible target for “employment”. As “Market” is a well identified location in the global namespace, and the manifests are machine readable, any object (1) can read them and identify objects (1) that fit to their requirements (rule-based decision making). So, it offers to “hire” “Milling”. After some negotiations of “apply’Trefine” interactions, “Milling” accepts hiring and grants ownership to “Manufacturing Company”. This may be achieved by applying the basic functionality of the invention.

By taking ownership of the machine, the company assigns its new position to the “organization” “Production” and thus allocates it to the respective intermediate namespace (2b) (Manufacturing company, Production, Milling) and binds the machine to comply to corporate standards (have a plant ID, a location, a responsible person, an organizational ID, alert to contacts in case of incidents etc.). The corporate standards are processes (in this case machine readable rules), that describe how to handle new hiring. These are now applied by the owner to the newly hired “owned” object (1).
This also implies that “Milling” can from now on act in the context of “Manufacturing Company” having the right IDs, and access to “Sales”, “Resource Management”, “Financials” etc.

As may be visible from the explanations above, the behavior of the whole network of things is determined by the rules embedded in the objects (1) as well as machine-readable descriptions that are interpreted during build up and changing the relationship of objects (1) while they interact with each other.

One technology for describing semantics and purpose of technical structures and organizations are “Ontologies”.

For the duration of the ownership, the machine will apply all these rules.

If budget is required to replenish material or to get maintenance services, the “Milling” would apply “Financials” of “Manufacturing Company” to “grantBudget” (Exhibit 9) and to “approve” (Exhibit 9). If it needs to identify itself as being part of the company, it would call “Resource Management” to get the right “Identity” and access credentials (name, password) to do so.

This also applies to encryption (to keep interaction between related instances of objects (1) private). In all these cases, “Milling” is still visible to “Market” for other employers. It could accept other “hire”s concurrently (if the current contract would allow “shared usage” mode) or sequentially (then terminating the “ownership contract” with “Manufacturing Company”). If an object (1) is “employed” to several different “Superiors’OWNers” it is strongly encouraged that the “Manifest’VServices include descriptions about how the conflict of interests is going to be handled during (and after) the employment duration.

When the machine ownership is changing, (within the legal entity or across legal borders) it is temporarily separated (not connected) and then attached to a different owner, that requires different rules to be applied. They
can technically be handled by inheritance of class- or instance values as well as object (1) methods from the new superior object (1). These rules may dictate many different roles to assume and activities to be executed at the same time (e.g. report power consumed to the commercial department, report incidents to the operational department, report production status to the plant supervisor). So, at any given point in time the machine is owned (has responsibilities for) different owners.

Technically the "rules" exchange and execution may be done applying state of the art procedures and protocols (e.g. publish and subscribe of the MQTT Standard (Message Queue Telemetry Transport), or by Web Services). "Object Orientation" methodologies as well as implementations are considered helpful in implementing the concept. It is also foreseen, that these technical means may change over time without compromising the organizational integrity as required by the governing contract. This may imply deployment of dynamic APIs to express, negotiate, terminate, extend, reduce or rename attributes, rules, verbs, values, of the contract between objects (1).

The present application describes various technical features and advantages with reference to the figures and/or various embodiments. Those skilled in the art will understand that the technical features of a given embodiment can in fact be combined with features of another embodiment unless explicitly stated otherwise, or unless the combination does not provide a solution to at least one of the technical problems mentioned in the present application. In addition, the technical features described in a given embodiment can be isolated from the other technical features of this embodiment unless explicitly stated otherwise.
It must be obvious to those skilled in the art that the present invention allows embodiments in many specific forms without departing from the field of application of the invention as claimed. Consequently, the present embodiments must be considered as illustrations, but can be modified in the area defined by the scope of the appended claims, and the invention must not be limited to the details given above.
CLAIMS

1. A system for interaction of at least two loosely coupled independently acting networked objects (1) for collaboration in a network with an environment of dynamically changing and possibly multiple ownerships to which an object (1) is connected and is outside the control of the owner wherein each object (1) includes at least one processor and a memory comprising an API to be executed by said processor allowing said object (1) be able to communicate through a wireless or wire link to some sort of intranet or internet network and to interact each other, said system using a Uniform Resource Indicator (URI) to identify object (1)s within a “global” namespace including many local namespaces (2c), said system being characterized in that in that it includes a minimal basic set of mechanisms for ownership coordination and or transfer, said minimal basic set of mechanisms consisting of interface methods and or globally standardized rules for:
   • remapping local (2c) and intermediate (2b) namespaces dynamically into a global namespace (2a);
   • describing/offering the service of the object (1) in a self-descriptive way like JSON or ontologies;
   • making the description/offering available for perception by another object (1) on a generally agreed, well defined place in the global namespace (2a);
   • withdrawing the visibility from the currently defined place;
   • presenting the services, i.e. actively make the services known according to the description known to possibly interested parties;
   • applying for deployment on the network within the context of one or many owners to offer services embedded on information contained within each object (1);
   • refining the service offerings, qualities, constraints and side conditions of the ownership relation dynamically
• hiring an object (1) into the context of one or more larger object(s) (1) to govern accessibility and services within this larger object (1), as well as concurrently acting on behalf of multiple organizations like a freelance contractor;

• accepting the hiring contract and locating the resource into the organization to make use of it and including the resource into the organizational context.

2. A system according to claim 1, characterized in that the refining functionality enables to negotiate duration of contract, exclusive or shared ownership, compensation/cost for deliverables and enabling negotiation of contracts like in a free market exchange.

3. A system according to claim 1, characterized in that the system or each object (1) comprises an arrangement to trigger manually or automatically the describing/offering method.

4. A system according to claim 3, characterized in that the triggering arrangement is a physical button or a voice command or a visual trigger or a face recognition means.

5. A system according to claim 1, characterized in that each object (1) of the system comprises an arrangement for deriving information about the published object (1) so as to make decisions to connect to the said published object (1).

6. A system according to claim 1, characterized in that each object (1) comprises an arrangement or program to check the descriptions against their goals/embedded rules not to violate possible existing "hiring contracts" with other objects (1).

7. A system according to claim 1, characterized in that cooperating objects (1) comprise at least an arrangement to manage a part of the "global" namespace, the "intermediate" namespace and the "local" namespace in an aligned and coordinate way so as to get a joint "intermediate" namespace between objects (1)/things.
8. A system according to claim 1, characterized in that objects (1) interact with other objects (1) either through function calls or by publishing messages to an URI in a joint namespace, said objects (1) using a publishing/subscribe method or function calls approach.

9. A system according to claim 1, characterized in that it comprises further methods for enhancing the minimalistic set of methods, said further methods comprising at least one of:

- implementing identity/resource management within an object (1) such as handling over and using references/IDs access credentials to other objects (1) that are needed to fulfill the agreed functionality of the resource, like for instance scheduling, deployment;
- implementing commercial management within an object (1) such as for instance budget planning/forecast, payments/value transfers, charging/billing, payroll, taxes.

10. A system according to claim 9, characterized in that the objects (1) using the enhanced set of methods are configured to dynamically transfer important context to adapt to varying situations that allow said objects (1) to behave different in different situations within the constraints of contracts that may be pre-negotiated or dynamically negotiated by the objects (1) based on need and situation.

11. A system according to claim 9, characterized in that the objects (1) using the enhanced set of methods are configured to have bookkeeping functionalities to selectively apply these contexts to the specific interactions with specific objects (1).

12. An object (1) for interaction of loosely coupled independently acting networked objects (1) for collaboration in a network environment for dynamically changing ownerships which is connected and configured for managing logical ownership on behalf of the physical owner said object (1) including at least a processor using an operating system and a memory
comprising an API executed by the processor and communication means through a wireless or wire link to some sort of intranet or internet network to be able to interact with other objects (1), the object (1) being characterized in that said API comprises a manager service (10) for implementing at least one of the public or external methods “present”, “apply”, “hire”, “refine” for presenting, applying, hiring, refining data and giving access to the manifest of said object (1) as well as providing processing functionality, said methods and manifest being contained in the memory of the object (1) publicly accessible, the manager service (10) communicating on one side with private or internal methods “register”, “unregister” and “remap” memorized in a first private memory (11) of the object (1) for respectively registering/unregistering and or remapping, and on another side with private variables memorized in a second private memory (12) of the object (1) for saving information concerning at least local root node as the root for its local namespace.

13. An object (1) according to claim 12, characterized in that the “present” method of a superior object (1) is called by an inferior object (1) to offer it’s services by making the superior node aware of the current location in the global namespace (2a) and by providing the formal description (manifest) of the offered services and the top node of the local namespace (2c).

14. An object (1) according to claim 12, characterized in that the “apply” method of an inferior object (1) is called by the superior object (1) to express the intent to temporarily (probation/qualification period) incorporate it into the superior’s ownership/namespace, make the inferior knowledgeable about the services the superior is offering (manifest) and the “temporary” place in the intermediate namespace (2b) the inferior is to use during the probation/qualification period, allowing the inferior object (1) to take decision if this transfer is compatible or not with the current state.

15. An object (1) according to claim 12, characterized in that the “hire” method of an inferior object (1) is called by the superior object (1) to express the intent to permanently incorporate it into the superior’s
ownership/namespace, make the inferior knowledgeable about the services
the superior is offering (manifest) and the place in the intermediate
namespace (2b) the inferior is later on to be working with, allowing the
inferior object (1) to take decision if this transfer is compatible or not with the
current state.

16. An object (1) according to claim 12, characterized in that the
“refine” method of an inferior is called by a superior to make it aware of:

• changed situation (constraints) ;
• changed employment/ownership conditions being a means to
“renegotiate” the ownership contract (like in a free market
situation, where a person applies for many jobs at the same
time) and to decide for the best terms and conditions among a
set of possible terms and conditions;
• a transfer of ownership completing after one or more
exchanges of “apply”Trefine” iterations ending with a
successful “hire” method call.

17. An object (1) according to claim 12, characterized in that the
private or internal method “register” is used during initial object (1) creation to
include the object (1) into the global namespace (2a) or even within a defined
“superior” intermediate namespace (2b), the private or internal method
“unregister” being used during transfer of ownership to remove the respective
object (1)’s local namespace (2c) from the “superior” intermediate
namespace (2b).

18. An object (1) according to claim 12, characterized in that the
private or internal method “remap” consist in the transformation of the local
namespace (2c) into the relevant intermediate namespace (2b) of the new
owner, all instances of the “local” namespace being changed by prepending
prefixes describing the “intermediate” namespace to the respective names
thus renaming all items to perfectly fit into the - now common - namespace
of both objects (1), all names of the object (1) being visible/accessible for other objects (1).

19. An object (1) according to claims 12 to 18, characterized in that it includes a “resource” service (13) which provides at least one of the following methods:

- `getIdentity` which allows to get a unique identity within the owner object (1);
- “`getAccess`” which allows to use resources of the owner without knowing where they are.

20. An object (1) according to claims 12 to 19, characterized in that each object of the system comprises a mechanism or a program for “transformation of names” which modify the own “names” (with no meaning outside the object (1)) into “name” that have a meaning for all other objects (1) in the global namespace (2a).
Figure 3
Figure 5
Figure 6
Figure 7
Figure 9
INTERNATIONAL SEARCH REPORT

INTERNATIONAL application No
PCT/IB2019/054414

A. CLASSIFICATION OF SUBJECT MATTER

INV. H04L29/08 H04W4/70
ADD.

According to International Patent Classification (IPC) or to both national classification and IPC

B. FIELDS SEARCHED

Minimum documentation searched (classification system followed by classification symbols)

H04L H04W

Documentation searched other than minimum documentation to the extent that such documents are included in the fields searched

Electronic data base consulted during the international search (name of data base and, where practicable, search terms used)

EPO-Internal, WPI Data

C. DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td>X</td>
<td>US 9 860 677 B1 (AGERSTAM MATS [US] ET AL) 2 January 2018 (2018-01-02) abstract; figures 1-3, 6, 7a, 8, 9, 10 column 2, line 15 - column 4, line 40 column 5, line 24 - line 67 column 7, line 7 - line 26 column 8, line 40 - line 62 column 11, line 6 - column 12, line 17 column 14, line 3 - column 15, line 53 column 18, line 13 - column 19, line 52 -----</td>
<td>1, 2, 9-12, 16 3-8, 13-15, 17-20</td>
</tr>
<tr>
<td>Y</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Further documents are listed in the continuation of Box C.

See patent family annex.

"A" document defining the general state of the art which is not considered to be of particular relevance
"E" earlier application or patent but published on or after the international filing date
"L" document which may throw doubts on priority claim(s) or which is cited to establish the publication date of another citation or other special reason (as specified)
"O" document referring to an oral disclosure, use, exhibition or other means
"P" document published prior to the international filing date but later than the priority date claimed

"T" later document published after the international filing date or priority date and not in conflict with the application but cited to understand the principle or theory underlying the invention
"X" document of particular relevance; the claimed invention cannot be considered novel or cannot be considered to involve an inventive step when the document is taken alone
"Y" document of particular relevance; the claimed invention cannot be considered to involve an inventive step when the document is combined with one or more other such documents, such combination being obvious to a person skilled in the art
"&" document member of the same patent family

Date of the actual completion of the international search
26 September 2019

Date of mailing of the international search report
08/10/2019

Name and mailing address of the ISA/
European Patent Office, P.B. 5818 Patentlaan 2
NL - 2280 HJ Rijswijk
Tel. (+31-70) 340-2040,
Fax. (+31-70) 340-3016

Authorized officer
Stevanovic, Natasa

Form PCT/ISA/210 (second sheet) (April 2005)
## DOCUMENTS CONSIDERED TO BE RELEVANT

<table>
<thead>
<tr>
<th>Category</th>
<th>Citation of document, with indication, where appropriate, of the relevant passages</th>
<th>Relevant to claim No.</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>abstract; figure 12</td>
<td>1,2, 9-12, 16</td>
</tr>
<tr>
<td></td>
<td>paragraph [0003] - paragraph [0004]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0025] - paragraph [0026]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0119] - paragraph [0123]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0135] - paragraph [0137]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0166] - paragraph [0167]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0309] - paragraph [0310]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>abstract; figure 1</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0017] - paragraph [0020]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0057]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0071] - paragraph [0073]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td></td>
<td>abstract</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0009] - paragraph [0012]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0021]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0060]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0073]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0083]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0097] - paragraph [0103]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0113]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0155] - paragraph [0157]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>paragraph [0168]</td>
<td></td>
</tr>
<tr>
<td></td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td>Patent document cited in search report</td>
<td>Publication date</td>
<td>Patent family member(s)</td>
</tr>
<tr>
<td>----------------------------------------</td>
<td>-----------------</td>
<td>-------------------------</td>
</tr>
<tr>
<td>US 9860677 B1</td>
<td>02-01-2018</td>
<td>DE 112017004996 T5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 9860677 B1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 2018063603 A1</td>
</tr>
<tr>
<td>US 2016019294 A1</td>
<td>21-01-2016</td>
<td>CN 106663143 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 3170111 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 6482645 B2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2017527016 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>KR 20170035961 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2016019294 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 2016011361 A</td>
</tr>
<tr>
<td>US 2011161478 A1</td>
<td>30-06-2011</td>
<td>CN 102687486 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 2520070 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2011161478 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>WO 2011080549 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>BR PI0504205 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CA 2517538 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CA 2833834 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CN 1755694 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>EP 1643730 A2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 4879547 B2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>JP 2006107501 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>MX PA05009679 A</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2006074876 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2006117024 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2009327312 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2010005071 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2012331103 A1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>US 2012331122 A1</td>
</tr>
</tbody>
</table>