Abstract—This paper addresses the problem of building a unified structure to describe a peer-to-peer system. Our approach uses the well-known notations in the P2P area, and provides a global architecture that puts a separation between the platform specific characteristics and the logical ones. In order to enable the navigation of the peer across platforms, a roaming layer is added. The latter provides a capability to define a unique identification of peer and assures the mapping between this identification and those used in each platform. The mapping task is assured by special wrapper. In addition, ontology is proposed to give a clear presentation of the structure of the P2P system without interesting in the content and the resource managed by the peer. The ontology is created according to the web semantic paradigm and using OWL language; so, the structure of the system is considered as a web resource.

Keywords—Peer to peer, ontology, owl.

I. INTRODUCTION

Peer to peer is becoming one of the most important areas in computer science, especially in the Internet topic. This is due to the nature of P2P model that have several advantages as self-organization, load balancing, adaptation, and fault tolerance. The P2P system is characterized by the autonomy of each peer and a high degree of decentralization. Each peer acts as client and as server in the same time, so it demands and provides services without using the same typology of a client/Server model.

In P2P systems, peers (nodes) are working to achieve specific needs. The needs can be classified into several areas as: file sharing (music or other), distributed computing, distributed storage, communication...

In practice, one can classify P2P systems in two main architectures: unstructured and structured approaches. Unstructured P2P approach are loosely controlled, there is not special peer to control the execution of query or to maintain a global repository over others peers. This permits to have a high dynamic behavior of the system. So, if a peer joins or leaves the overly network, there are no added tasks to do.

All peers have equal role. Gnutella [3], KaZaa [7] and Freenet [5] belong to this type.

In the case of the structured P2P system, the Distributed Hash Tables play a key role to determine exactly the location of the peer and its resources. Many systems are proposed as: Chord [4], CAN [13]. In these systems, a hash function is used to obtain a valid key.

On the other hand, the semantic web is considered as the new vision to the web that try to give semantic to the web resources. Since it is based on a logical foundation, the web semantic and its language OWL provide a good solution to semantic problems and ontology presentation.

One remarks that several P2P systems are proposed; each system is based on a specific platform and appropriate technique. In these systems, a peer is presented with a specific structure and it is influenced by the technical aspect of the node. In our work, we aim to provide a new capability to peer that is the roaming service. As in mobile telephony, the peer can navigate across P2P systems and it maintains a unique profile to enhance the result’s quality. The unified profile is assured by our proposed ontology. Special wrappers permit the mapping task between the global profile and the specific ones. In addition, the use of OWL language to describe the structure of the P2P system permits to export the P2P system itself as web semantic resource. So, the structure can be used by others with strength semantic.

The remaining of this paper is organized as follow: in section 2, we give an overview of the P2P technology and the web semantic paradigm. In section 3, we present some important related work in the domain of web semantic and P2P. The fourth section introduces our approach that is base on a new roaming layer and presents the proposed ontology to manage the structure of the system. The paper is ended by a conclusion.

II. PEER-TO-PEER SYSTEM AND WEB SEMANTIC

The peer to peer paradigm allows giving the capabilities of client server paradigm to each node of the network.

In this section, one will give an overview of the Peer-To-Peer paradigm and the web semantic technology.

A. Peer-To-Peer Characteristics

The P2P is characterized by:

- Autonomy of peers: Nodes participating in a P2P system operate autonomic entities. Each peer can decide to leave the system, forward or performs a query.
- Ad hoc nature: There is no control of any node. This
characteristic may poses a big problem in the organization of the whole system. So, a mechanism is incorporated to ensure a self-organization.

Fault tolerance: In P2P, the system do not crashes if a number of peers cannot provide services (ex. leave the system).

B. Peer-To-Peer Substrates

As shown in Fig 1, the substrate is a key component in any peer architecture. It provides a mechanism for managing peers and resources. For peers, it permits joining and leaving services. For resources, it permits the storing and the locating of every resource in the node.

P2P systems are divides according to their P2P substrate mechanisms. Therefore, one can divide the P2P substrates into two mains categories: unstructured P2P and structured ones.

Unstructured P2P substrates are loosely-controled. There is not a special peer to organize or to maintain the system; the system is organized as follow.

Two peers that maintain a connection are called neighbors. A peer can have more than one neighbor; the number of these neighbors is called outdegrees. In an unstructured P2P system, messages (queries or results) are only routed along an open connection. Therefore, if there is not an open connection between two peers. The message must use a path and pass along other peers that are considered as bridge. The length of path is called hop.

At the peer level, if a user sends a query, the peer becomes a source that sends the query to all its neighbors. Other techniques are proposed to reduce the number of the sent message. They consist to send to a set of neighbors rather than to all ones. Each message is accompanied with a time to leave (TTL) values that specify the number of hop. This number decrease on each peer and the sending ends when the TTL =0.

In Fig 2, Peer 1 has two neighbor peers (Peer2, peer3). Peer1 send a message to its neighbors, the latter decrease the TTL value, peer2 do the same task. See Fig 2.

In the follow, we present some important systems in the unstructured peer category.

Gnutella[3] is considered as one of the most widely deployed file sharing system. The Gnutella peer is called servant (server/client). As presented before, the peer uses flooding technique to propagate a message; the flooding is controlled by a TTL value whose default value is 7.

Gnutella system provides a high degree of reliability. However, since the peers have equal responsibility, there is no distinguishing between the peers according to their bandwidth or performance. In addition, the use of flooding technique with all the neighbors consumes the bandwidth without guaranty the quality of results.

In order to solve this problem, solutions are proposed. In [2], algorithms to reduce the bandwidth consummation by sending messages to a set of neighbors are proposed. An other solution consists of the introduction of the Super-Peer approach [1].

In a super-peer system [1], the network is divided into groups, each one contains one special peer called a supper-peer. All the peers in the group are connected to the supper-peer and the latter is connected to other supper peers of other groups. In the terminology of the supper-peer approach, the group (supper-peer and its peers) is called cluster. The main problem in the supper peer approach is the risk for the dysfunction of the supper peer. If the latter is disconnected, the entire cluster is disconnected. In order to resolve the problem a supper-peer redundancy is proposed. Fig 3 shows two clusters, the first with one supper peer, and the second with two supper-peers. The connection between the two clusters is assured by the supper peers.

Structured P2P substrates. In this category, the placement of the resource and the peer is tightly associated to the
network structure. The substrate component uses a Distributed Hash Table (DHT) to locate the peers and the resources. The DHT technique guarantees the locating the file or the peer if they are connected.

DHTs organize the system in different manners. One can uses different geometrical shapes as tree, ring, hypercub. The DHTs are strong technique to lookup for peers resources. However, it suffers of some problems like a) DHTs support an exact match query, so, a little change in one characteristics of the resource causes the fail of the lookup. b) The peers can join or leave much time in a few times, this needs more operations for the management of the DHT.

Many works belong to this category; in (Content Addressable network) CAN the network is considered as virtual d-dimensional Cartesian coordinate. CAN uses dynamic partitions, each ones contain a set of peers. Chord associates to nodes an m-bit identifier using a special hash function as SHA-1. The identifiers values are ordered in a circle manner. Each node maintains a local routed table called the finger table. The system uses the IP addresses and port number to determine the physical location of the node.

C. Semantic Web

The semantic web is a new technology that intends to make web resources more readily from users and machines. The semantic markup extensions permit to obtain this goal. These markup extensions are considered as meta-data annotations to describe their contents.

In the area of the web semantic, ontologies play a key role to describe information sources and permit an efficient share of meta-data. This need prompts the development of several languages as OIL, DAML+OIL, and OWL [11]. The OWL language provides some semantic capabilities of RDFs language as: declaration of classes, organization of classes in subsumption hierarchy. Moreover, OWL gives other extensions for RDFs as: putting that a property is transitive, symmetric, or it is an inverse of another property.

In our work, we aim to use the semantic web to export the P2P system as a web resource.

III. RELATED WORK


When analyzing the P2P systems, one remarks that several architecture and technique are proposed to assure the lookup service and the organization task. For the lookup service, techniques as DHT (in structured P2P) provide identifications for the peer and the resources. However, this identification differs according to the used system; one may have the same peer with the same resources but the identification changes.

So, A peer p1 can be identified in P-Grid with ID = “pgrids://01001101” and the same peer can be reference in JXTA with ID="jxta:uuid-150325033CD144F82DED74E". Furthermore, a peer must belong to a unique P2P system in order to be referenced. So the peer cannot navigate between P2P systems because of the characteristics of each system. In addition, each system uses a specific platform and peers are relied to the specific characteristic of its platform.

In order to enhance the capability of a P2P system, we propose a new P2P design that has as objectives:
- To assure a clear separation between the physical and technical aspect of a peer and the logical ones.
- To propose a unique identification of peers unless they use different platforms.
- To permit to the peer to navigate between P2P systems and in the same time to assure a transparency to the user.
- To export the peer structure as web semantic resource by proposing ontology to unify the notation.

IV. OVERVIEW OF THE PROPOSED APPROACH

Our approach is based on platform independent identification. A peer has a unique identification that can be mapped into specific one, according to the platform or the system used. In order to provide a unique view of the peer a clear description of the peer structure is needed. The description must be independent from the technical aspect of the used platform. In this case; the ontology and the OWL language provide a good solution to express the semantic of the structure. The OWL language proposes some semantic relations that are needed in our work as Restriction; transitive; symmetric, transitive…

This ontology provides two mains services. One of the most important benefic of this ontology is the roaming service.

A. Peer Roaming Service

Peer roaming is a new service that consists of a few services provided for a peer in order to permit:
- An easy navigation across a P2P system like in the mobile telephony. Therefore, the peer can participate to several P2P systems.
- To export the meta-data to several systems at the same time.

The roaming service is used within other components (cf fig 1). Fig. 4, shows the global architecture of our approach.
The peer’s structure in the platform layer is called "physical". However it is called “logical” in the roaming one.  
- The roaming layer provides the following services:  
  - The interception of the JoinQuery. This query is sent by other peers to join each platform.  
  - The interception of the AddRessource query. The latter is done by the user in order to publish some resources to share.  
  - Establishing a connection between the logical peer and the physical one. This is done by mapping the ID from global to local.  
  - Storing the profile of the peer as OWL document. This permits to use it with other platform or with the same platform in other time.  
  - Establishing a correspondence between the logical peer and the physical one. This is done by mapping the ID from global to local.  

For each platform is one associated a special wrapper. The latter uses a correspondence table to map from the logical to the physical representation of the peer.  
The platform layer contains the specific platform, techniques to manage the system. The characteristics of each platform are conserved so long as possible.  
In the next section, we present the proposed ontology to manage the system.

B. The Proposed Ontology

In order to build ontology, one must use an appropriate methodology or processes. In our case, we use an Ontology Development Process for the Semantic Web [10]. In this process, we follow five phases that are needs specification, conceptualization, formalization, codification and finally verification.  
In the following, we directly present the terms dictionary and the codification in OWL language.

<table>
<thead>
<tr>
<th>Concept Name</th>
<th>Attributes</th>
</tr>
</thead>
<tbody>
<tr>
<td>PeerCommunity</td>
<td>-PeerComID</td>
</tr>
<tr>
<td></td>
<td>-PeerComdesignation</td>
</tr>
<tr>
<td></td>
<td>-Description</td>
</tr>
<tr>
<td>PeerCluster</td>
<td>-ClusterID</td>
</tr>
<tr>
<td></td>
<td>-NumberOfPeers</td>
</tr>
<tr>
<td></td>
<td>-NumberOfRedundancy</td>
</tr>
<tr>
<td></td>
<td>-ClusterBandwith</td>
</tr>
<tr>
<td>PeerProfile</td>
<td>-OnlineState</td>
</tr>
<tr>
<td></td>
<td>-PlatformInUse</td>
</tr>
<tr>
<td></td>
<td>-PeerQuality</td>
</tr>
<tr>
<td></td>
<td>-IsSupperPeer</td>
</tr>
<tr>
<td>PeerPlatformProfile</td>
<td>-PeerUpBandwidth</td>
</tr>
<tr>
<td></td>
<td>-PeerDawnBandwidth</td>
</tr>
<tr>
<td></td>
<td>-PeerPlatformID</td>
</tr>
<tr>
<td></td>
<td>-PeerTypeMachine</td>
</tr>
<tr>
<td></td>
<td>-PlatformID</td>
</tr>
<tr>
<td>PeerRessource</td>
<td>-PeerRessourceID</td>
</tr>
<tr>
<td></td>
<td>-PeerRessourceType</td>
</tr>
<tr>
<td></td>
<td>-PeerRessourceURI</td>
</tr>
<tr>
<td>Peer</td>
<td>-PeerID</td>
</tr>
<tr>
<td>PeerQuery</td>
<td>-QueryID</td>
</tr>
<tr>
<td></td>
<td>-PeerQueryContent</td>
</tr>
<tr>
<td></td>
<td>-QueryType</td>
</tr>
</tbody>
</table>

On each attribute, we define a restriction on its type or the values that can contain. The restrictions are part of the OWL language and are used according to its syntax.

Ex: the concept “PeerCluster” contains an attribute “Number of redundancy”. The value of the latter is between 0 and 3. So, by using


An other example of a restriction is the fact that we indicate that some values are allowed for an attribute. The concept “PeerPlatformProfile” contains an attribute “PeerTypeMachine”. We determine that a set of values are allowed (PC; PocketPc…). With OWL; we use the "<owl:one of>.

We according to [10], a binary relation table that contains the relation between the concepts must be created. It permits to indicate the source cardinalities (SC), the target cardinalities (TC), the name of the relations and the inverse ones. In addition, we create an attributes table to describe each attribute, logical axioms table and instances table.
The OWL language provides many possibilities to enhance the semantic of the relations.

- To indicate that the relation “ContainsPeer” is the inverse of “BelongsToCluster” we use “owl:inverseOf rdfs:resource=“#ContainsPeer””.
- To indicate that the relation “HasNeighbor” is symmetric (If P1 “HasNeighbor” P2 then P2 “HasNeighbor” P1), we use owl:ObjectProperty rdf:ID=“HasNeighbor” rdf:type owl:Restriction owl:onProperty owl:inverseOf owl:Restriction rdfs:subPropertyOf rdfs:subClassOf owl:Class owl:Class owl:Class owl:Class rdf:resource=“http://www.w3.org/2002/07/owl#SymmetricProperty”.
- To indicate that the “HasNeighbor” is transitive (If P1 “HasNeighbor” P2 and P2 “HasNeighbor” P3 then P1 “HasNeighbor” P3) we use owl:ObjectProperty rdf:ID=“HasNeighbor” rdf:type owl:Restriction owl:onProperty owl:inverseOf owl:Restriction rdfs:subPropertyOf rdfs:subClassOf owl:Class owl:Class owl:Class owl:Class rdf:resource=“http://www.w3.org/2002/07/owl#TransitiveProperty”.

In the following, one directly presents the ontology formalism according to SHIQ DL,

PeerCommunity:=(∃ PeerComID.String)∩(∃ PeerComDesignation.String)∩(∃ Description.String)∩(1 ContainsCluster PeerCluster)

PeerCluster:=(∃ClusterID.String)∩ (∃ NumberOfPeers.Integer)∩ (∃NumberOfRedundancy.Integer)∩ (∃ClusterBandwith.Double)∩ (∃ BelongsToCommunity PeerCommunity)∩(∃ Peer CommunityPeer)∩(1 HasNeighbor PeerCluster)

Peer:=(∃ PeerID.String)∩(∃ HasPeerProfile PeerProfile)∩(1 HasPeerPlatformProfile PeerPlatformProfile)∩(∃ SendsQuery PeerQuery)


The following, we present a part of the ontology coded in the OWL language.

...
V. CONCLUSION

In this work, we presented our work to enhance P2P architecture with a roaming service. The latter allows the peer to navigate across different platforms and to maintain a history of the peer. This is assured by the use of a global profile in the roaming layer; this profile is coded with OWL language. The latter gives strong semantic relations as transitive, symmetric, inverse … The structure of the whole system is exported as a web resource. A special wrapper maps the global ID to local one according to the target platform. The main benefit of such ontology is to permit to other platform or user to share a global structure and permits to locate a peer independently to a specific platform.

REFERENCES