GWD-TYPE
Category: TYPE

Ian Taylor, Dimitris Lioupis, Ashley D. Lloyd,
Sergio Mendiola, Dejan Milojicic, & Krishna Sankar

RG: Appliance Aggregation Architecture Group (APPAGG) March 2003

Appliance Aggregation Architecture Terminology, Survey, and Scenarios

Status of This Memo

This memo provides information to the Grid community [topic]. [for non-recommendations track documents, add: It does not define any standards or technical recommendations.] Distribution is unlimited.

Copyright Notice

Abstract

This document presents a survey of the field of appliance aggregation and describes its role within the Grid Community from a social and scientific perspective. The introduction describes a specific context in which the Grid might be expected to have a socio-economic impact. It relates the existing study of technology usability, uptake and 'social-shaping' to the interface between personal communication and computational devices, interoperating with the user, each other and the Grid through an appliance aggregation architecture. A terminology section is given followed by an overview of existing standards relating to this field and a comparison of the current research and development of related technologies within the GRID and P2P worlds. Finally, a taxonomy and three user scenarios or case studies are presented.

Contents

i.j.taylor@cs.cardiff.ac.uk

i.j.taylor@cs.cardiff.ac.uk

1. Introduction: Usability, Uptake and the 'Business' Case

The Grid's potential contribution as and for a social 'good' requires an exploration of how its capabilities are socially shaped through the creation, adaptation and adoption of applications by society. The rate of uptake in turn determines the rate of return on investment in the infrastructure and hence the availability of capital for wider social access to be delivered. Appliance Aggregation sits at the interface between the Grid and established personal computing and communication devices through which the impact of the Grid will be measured in terms of its impact on social interactions, and vice versa.

The first recognisable Grid, from which a number of later analogies with computational grids have been drawn [1] *(Foster and Kesselman, 1999)*, was Edison's power distribution grid in New York. This first supplied power to Wall Street in 1882 and had to compete at the same price as the existing dominant technology hence costs had to be kept down. The core of the economic analysis here turned out to be Ohm's law, and it was used by Edison to consider every aspect of the cost of generating, distributing and using electricity [2] *(Hughes,1999)*. The choice of New York and in particular Wall Street was made for the same reasons – Electricity could only compete with Gas if the density of population was high enough to provide an economic return given the cost relationships defined by Ohm's Law. Of course Edison also had to choose an area in which the density was not only high, the availability of capital for switching costs between technologies could also be found. Here the wider social implications of any new infrastructure as a social 'good' become clear – put simply, if those investing in a new infrastructure do not receive an economic return (which implicitly accounts for the time value of money), the speed with which that infrastructure is made available to the wider community will reduce. The trend established by Edison in New York is still echoed in the differences of quality of access to electricity and telecommunications in city versus rural communities in many countries worldwide. The complexity of the required analysis was acknowledged by Edison in 1914 [3] (Runes, 1948): "Economic questions involve thousands of complicated factors which contribute to a certain result. It takes a lot of brain power and a lot of scientific data to solve these questions."

Today grid protocols and architectures establish new 'laws' that relate activity to the cost of that activity, and lay the foundation for product and service qualities that can be communicated to a market, and will require market acceptance if the cost-volume relations are to break even. This latter issue is a social-shaping issue, and here the analysis is necessarily more complicated as the use to which APPAGG devices can be put far exceed those of the first electric lights who were clearly direct substitutes for an existing, inferior, technology. The case of 3G uptake in Europe and the contrasting uptake of NTT DoCoMo's iMode in Japan versus WAP in Europe [4] [Ratliff, 2002] demonstrates that superior computing and communication technologies do not always provide an economic return, with direct consequences for wider social access to the network.

APPAGG is core to allowing economic returns on the Grid infrastructure as provides a direct interface with and between devices that are already active in the domestic market. Such appliances are increasing in functionality and in resources and are now embedded in may social as well as business processes, from mobile 'phones and PDAs to MP3 players and digital cameras. In the future, the range of such devices, the capabilities of individual devices and their embeddedness in societies' processes (Binford, 1979) is expected to increase, with watches, jewelry and clothes becoming capable of integration. However, such devices must be connected together within a personal area network coherently in order to be utilized in an effective manner.

The interactions supported by APPAGG's architecture relate directly to social interactions and hence to another form of 'Grid' defined by Mary Douglas in the 1970s that relates to social organization. Her definition of Grid relates to systemic constraints; for example formal rules or economic constraints. She also describes a spectrum over which the strength with which these constraints are applied has a direct impact on social behaviour. A 'strong' grid means severe

constraints on individual opportunities for voluntary agreements with other individuals, interactions are regulated and constraints are formalized institutionally. A weak grid allows significant autonomy and freedom to transact, with the result that competition is significant [5] (Douglas, 1982). Though Douglas's work focused on social rather than the broader socio-technical frameworks of later authors [6] (Law and Bijker, 1997), the convenient analogue with the potential implications of Grid technology and the spectrum of individual, group, business and government interactions that global information infrastructures support leads to a typology in which the Internet might be described as a 'weak grid', and the Grid, through projects such as Globus, potentially described as a 'strong grid'. An appreciation therefore of the potential impacts of APPAGG architecture design on the effective 'grid strength' of the resulting infrastructure would allow analysis of compatibility with differing social norms found within the various communities of practice that comprise the global Grid community. We contend that such compatibility could be an important determinant of the uptake of Grid technologies as although standardisation as a design solution can offer benefits within a global market [7] [Levitt, 1983; [8] Ohmae, 1989], important regional distinctions exist [9] [Rugman, 2001] and local, regional, and national factors need to be accommodated in order to optimise business practices [10] [Hofstede, 1980; [11] [Trompenaars, 1993].

Ultimately these factors determine a return on investment, reduce the cost of capital for further investment in an infrastructure with wider social as well as economic objectives, and hence determine the socio-economic outcomes of investing in this technology. However the move from understanding use of one information infrastructure technology to informing the design of a future technology is not automatic. [12] Palen et al. (2001) attempt this by establishing a typology for mobile communications that separates technology into hardware, software, netware and bizware. [13] Churchill and Wakeford (2002) also attempt decomposition, this time into 'design dimensions' for mobile communications, from tight to loose mobility and close information to far information. This is reported to have helped designers at Fuji-Xerox to design technology that fits particular interaction settings rather than the interaction being forced to adapt to the technology. Though this is challenging for the design of mobile telecommunications infrastructure alone, the Grid and APPAGG will be required to inter-operate with a much wider set of 'ubiquitous' devices that scale by orders of magnitude in computing power and communications speed; to provide low-cost universal access to the commodity market; and secure highly efficient execution of parallelised code on high-performance computing nodes – all coupled through a global common infrastructure. One of the tasks therefore of the APPAGG research group will be to draw on these lessons and use them to inform current design.

Terminology

This Section attempts to define terms defining the appliance aggregation [24]. The following three terms are directly related to appliance aggregation:

- **Appliance**: a device (communicating on a network) capable to aggregate with other devices; a smallest unit of aggregation
    - examples: camera, PDA, laptop, watch, hotspot, etc.
    - non-examples: no communication, shared & queued devices.
- **Ensemble**: a group of appliances aggregated to perform a function greater than their parts
    - examples: a camera, PDA, and a watch used in concert to capture, annotate, and communicate pictures
    - not an ensemble: a client-server relationship such as a users home PC used to access Yahoo!

- **Appliance aggregation**: presents the illusion of multiple appliances operating together as a single entity for a period of time
    - examples: use the screen of another device for display; exe-cute a single application across multiple appliances
    - non-examples: execute a distributed algorithm on a work-station cluster; send a presentation to a printer

The following terms are essential for understanding the appliance aggregation architecture. These terms are defined n the context of the appliance aggregations rather than being defined in general.

- **Trust:** each ensemble should maintain trust boundaries for its owner. Each ensemble should wok for its owner and be unusable for other users unless granted by the owner.
- **Data and sharing:** an ensemble should support sharing of data and content across the ensemble in a sufficiently transparent way. “Sufficiently” relates to problems with disconnection and disappearance of devices.
- **Application sharing:** it should be possible to share applications across an ensemble. This relates to being able to start an application on one device and being able to continue using it from another device.
- **Service sharing:** it should be possible to share services and functionality across an ensemble, similarly to applications.

Related technologies

- **Grids:** compared to Grids, Appagg addresses personal devices and devices in locale. Its focus is on the user data, applications, and services, rather than computational and data grids. Similarities encompass being able to aggregate a number of devices and its resources to serve a common purpose for a user, otherwise not possible.
- **P2P:** Appagg relies on P2P techniques and the whole aggregation is essentially P2P-centric. Compared to most P2P systems, Appagg primarily addresses personal appliances and appliances in local rather than hose on the Web. However, Appagg shares with the P2P the following attributes: decentralization, ad-hoc, self-organization, cost-of ownership, and disconnection.
- **Middleware:** Appagg can be classified as yet another middleware in a way that it is really a layer between operating system and applications. Similarly to differences with Grids and P2P, Appagg primarily addresses devices in locale rather than those on Internet. Therefore, the issues, such as scalability, disconnection, security, etc. have entirely different perspective in Appagg compared to systems, such as DCE, CORBA, and Jini.

2. Appliance Aggregation: Setting the Scope of Interactions

Appliance aggregation involves connecting such devices in a simple but coherent fashion so that there exists a common model of ownership, shared state, shared applications, and shared functionality. Such communities in appliance aggregation are known as an **ensemble**. By using ensembles, users can enable easier control over appliances, transparent synchronization of data when and if needed and continuous access to applications and functionality from any appliance.

There are various technologies that provide appliance aggregation support but these differ in approach and differ in at which point in the OSI stack that they achieve this. The APPAGG group are looking at this from a lower level, something just above network protocols, such as Bluetooth

but below distribution architectures such as Jini, JXTA, Sony HAVi and Microsoft UPnP. This is illustrated in figure 1 below.

In this figure, the various layers are shown along ranging from the low level network layer to the applications and users. The middleware layer is further sub-divided into middleware systems that based on certain programming languages or protocols (e.g. Jini basically allows a programmer to distribute Java objects as services) to those that provide true abstraction from the underlying transport protocols, programming languages or operating systems. An example here is JXTA which does not rely on any of these but further, JXTA actually provides access to lower level mechanisms such as routing, hence, its protrusion into the lower-level mechanisms. Bluetooth on the other hand is basically a network protocol but implements some behaviour on top of these and provides interfaces for the exchanging of various types of message content e.g. data, voice, and content-centric applications.

The APPAGG group covers the domain above the transport protocols concerning additional interfaces to allow devices to form ensembles and interoperate. For example, to implement this, one could extend Bluetooth or integrate and evolve JXME or use any other number of solutions offered that will be discussed in this document.

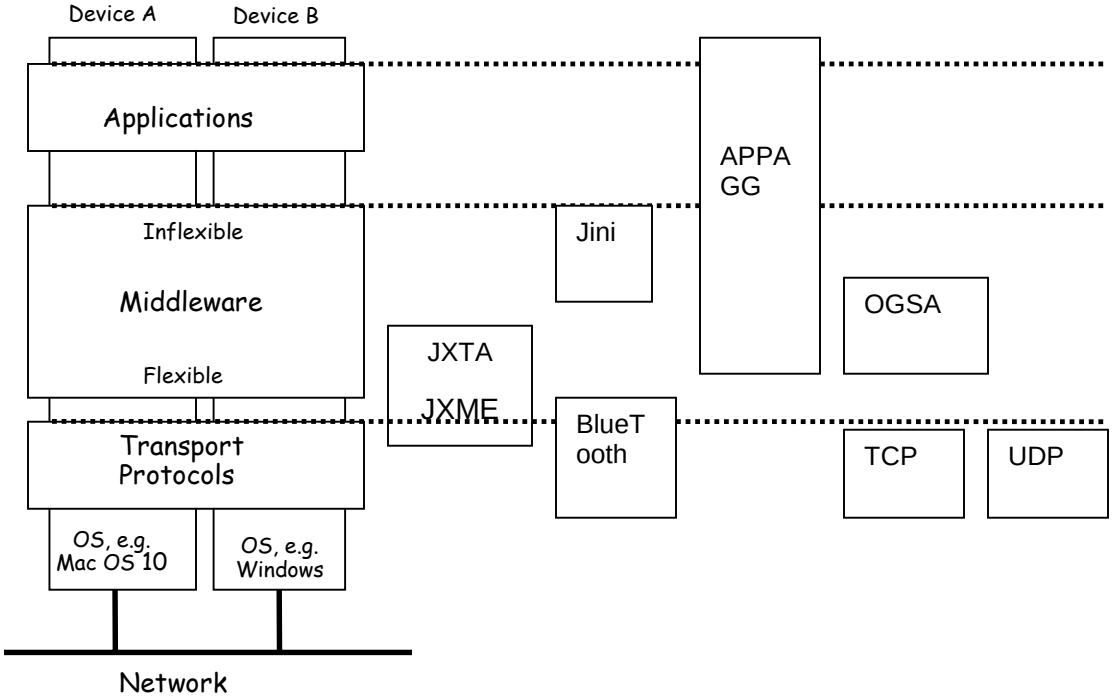

Figure 1: The Appliance Aggregation Stack.

Certain architecture e.g. Jini are at a too high a level on the stack. Jini relies on Java and is not unrealistically implementable on small devices. Other systems at this level include Sony HAVi and Microsoft UPnP. Typically these systems provide support for remote control and coordination, but they do not couple the devices tightly together. Instead they provide functionality similar to more traditional distributed systems (remote object invocation, naming, etc.).

In the rest of this document we relate this goal to existing technology that already claim to implement such behaviour or implement subsets of this behaviour. In the next section, we examine existing protocols and systems for communicating between appliances. We then relate this topic to the current state of the art in the field of peer-to-peer computing and then to Grid

Computing. Finally, we offer some user scenarios to give some examples of user interactions and how they might use the aggregation of such appliances.

3. Existing Protocols and Systems for Connecting Appliances

3.1 Bluetooth

Bluetooth uses short-range radio technology aimed at simplifying communications between devices and the Internet and also aims to simplify data synchronization. Products with Bluetooth technology must be qualified and pass interoperability testing by the Bluetooth Special Interest Group prior to release. The Bluetooth 1.0 specification consists of two documents: the Foundation Core, which provides design specifications, and the Foundation Profile, which provides interoperability guidelines. This specification contains the information necessary to ensure that diverse devices supporting the Bluetooth wireless technology can communicate with each other worldwide. Bluetooth's founding members include Ericsson, IBM, Intel, Nokia and Toshiba.
Unlike many other wireless standards, the Bluetooth wireless specification includes both link layer and application layer definitions for product developers which supports data, voice, and content-centric applications. Radios that comply with the Bluetooth wireless specification operate in the unlicensed, 2.4 GHz radio spectrum ensuring communication compatibility worldwide. These radios use a spread spectrum, frequency hopping, full-duplex signal at up to 1600 hops/sec. The signal hops among 79 frequencies at 1 MHz intervals to give a high degree of interference immunity. Up to seven simultaneous connections can be established and maintained.

3.2 HAVi

Sony HAVi is short for Home Audio Video interoperability. It is a vendor-neutral audio-video standard aimed specifically at the home entertainment environment. HAVi allows different home entertainment and communication devices (such as VCRs, televisions, stereos, security systems, video monitors) to be networked together and controlled from one primary device, such as a PC or television.

HAVi uses the IEEE 1394 as the interconnection medium and allows products from different vendors to comply with one another based on defined connection and communication protocols and APIs. One of the key features of HAVi is its ability to easily add new devices to the network. When a new device is installed, the system will configure itself to accommodate it. Other services provided by the distributed application system include: addressing scheme and message transfer, lookup for discovering resources, posting and receiving local or remote events, streaming and controlling isochronous data streams.

This industry standard has been jointly developed by Grundig AG, Hitachi Ltd., Matsushita Electric Industrial Co. (Panasonic), Royal Philips Electronics, Sharp Corporation, Sony Corporation, Thomson Multimedia and Toshiba Corporation.

3.2.1 3.3. IEEE1394 Standard – ‘Firewire’

The IEEE 1394 is a fast external bus standard that supports data transfer rates of up to 400Mbps (in 1394a) and 800Mbps (in 1394b). Products supporting this standard go under different names, depending on the company. Apple, which originally developed the technology, uses the trademarked name of FireWire. Other companies use other names, such as i.link and Lynx, to describe their products. A single 1394 port can be used to connect up 63 external devices. In addition to its high speed, it also supports isochronous data (i.e. delivering data at a guaranteed rate). This makes it ideal for devices that need to transfer high levels of data in real-time, such as

video devices. Like USB, it supports both Plug-and-Play and hot plugging, and also provides power to peripheral devices.

There are two levels of interface in IEEE 1394, one for the backplane bus within the computer and another for the point-to-point interface between device and computer on the serial cable and a bridge that connects the two environments. The backplane supports 12.5, 25, or 50 Mbps and the cable interface supports 100, 200, or 400 Mbps. Each of these interfaces can handle any of the possible data rates and change from one to another as needed. The serial bus functions as though devices were in slots within the computer sharing a common memory space. A 64-bit device address allows a great deal of flexibility in configuring devices in chains and trees from a single socket.

The 1394 standard requires that the device is within 4.5 meters of the bus socket. Up to 16 devices can be connected in a single chain, each with the 4.5 meter maximum (before signal attenuation begins to occur) so theoretically you could have a device as far away as 72 meters from the computer.

Other approaches for to connecting consumer devices include:

- Universal Serial Bus (USB) which provides the same "hot plug" capability as the 1394. Its less expensive but data transfer is limited to 12 Mbps.
- Small Computer System Interface offers a high data transfer rate but requires address preassignment and a device terminator on the last device in a chain.

3.3 Plug and Play (UPnP)

Universal Plug and Play (UPnP) is a standard that uses Internet and Web protocols to enable devices (e.g. PCs, peripherals, intelligent appliances, and wireless devices) to be plugged into a network and automatically know about each other. With UPnP, when a user plugs a device into the network, the device will configure itself, acquire a TCP/IP address, and use a discovery protocol based on the HTTP to announce its presence on the network to other devices.

For example consider the following: you could use UPnP to send a picture taken from a digital camera and send it directly to a printer. This could be achieved by the camera issuing a "discover" request after it had taken a picture asking if there were any printers on the network. The printer would identify itself and send its location in the form of a universal resource locator (URL). The camera and printer use XML to establish a common language or "protocol negotiation" to talk to each other and determine capabilities. Once a common language was established, the camera would control the printer and print the photograph you selected.

Microsoft is one of 29 companies sponsoring UPnP and their hope is that UPnP will make it as easy to plug a device or appliance into a home or small business data network as it is to plug a lamp into an electrical outlet. It is also described as "seamless proximity networking" that provides "standardization on the wire rather than in the devices," using existing Internet standards.

3.4 Rendezvous

Rendezvous is a networking technology by Apple that enables an automatic creation of an ad-hoc network of computers and devices and discovering the services available on them. It is intended for enabling sharing of files, content, printers, and other devices. For example, it enables discovery, network integration, setup and administration of router, webcam, printer, and laptop.

Rendezvous is a collection of technologies. At the core of rendezvous, there are link-local addressing, multicast DNS, and DNS service discovery. On top of the core technologies, there are utilities, such as for chatting, sharing data, distributed games, and simply access and control of remote devices, such as home stereo, television, and media server.

Rendezvous works in the following way. When a new device is added to a network without the support of DHCP, Rendezvous configures it using link-local addressing. Link-local addressing is based on randomly selecting an IP address from a predefined range and assigning it o the new device. Then it is verified throughout the network if this address is used by any other device. This process is repeated until an unused address is found.

For name services, Rendezvous uses a variation of the DNS, called multicast DNS-Service Discovery (mDNS-SD). Compared to traditional DNS, mDNS-SD is inherently distributed (rather than based on central DNS servers) and automatically maintained (rather than supported by IT professionals). Every device advertises its service by sending notification message which describes the type of the service, name o the service, IP/port address, and additional information if applicable. This information is maintained by each device in its own lightweight DNS server.

Apple made rendezvous source code public, with the support for computers and devices running UNIX, Linux, and Windows operating systems. Rendezvous is in the process of standardization through IETF, coordinated by the ZeroConf networking group at IETF. Apple also signed up a number of companies supporting it, including printer companies (e.g., HP, Xerox, Lexmark, Epson, and Canon), storage and data base companies (e.g. Chapparal and Sybase), and content delivery (e.g., TiVo).

We envision Appagg as a layer built on top of and using Rendezvous as one of the potential implementations of appliance aggregation at the networking level. We would extend rendezvous with the ways of creating trust boundaries, implementing new ways of sharing data, applications and services.

4. Existing P2P Technologies

It could be argued that appliance aggregation is addressed by some existing peer to peer systems [23]. This argument, however, is restricted since there is no existing infrastructure that tackles the intranet/internet capabilities required by uses of the appliance aggregation framework. In this section, we present some existing technologies, both protocols and middleware that may or may not be investigated further within this context. We first look at current popular P2P applications, including CPU and file sharing to explore the possible synergy between these fields. We then consider advances in the P2P middleware arena that could possibly address some of the issues for appliance aggregation.

4.1 Current P2P Applications

There are many types of applications that use P2P technology. Examples are file sharing e.g. Gnutella, CPU sharing e.g. companies such as United Devices and Entropia and University based projects such as SETI@Home and instant messaging systems, such as Jabber. The relevance of any of these to appliance aggregation is limited but below we outlined some of the ideas that could be lent from such applications in the design of the appliance aggregation framework. We consider file sharing as an example.

Most file sharing applications base their algorithms (either directly or indirectly) on decentralized distributed search made popular through the Gnutella protocol [14]. Such a searching capability might be needed for appliance aggregation but this would be a small sub section of the whole framework. Further, other approaches (e.g. AAFS) have been suggested to implement distributed file system needed. To explore the usefulness of current file sharing technology, we must first address the likelihood of the number of appliance that could be connected. If we consider one household, then in the future, if DVD's, phones, Hi-FI's, household appliances, rings, PDAs etc can all contribute to this network then we could be looking at 10's or even 100's of devices connected together in this fashion. If each had some storage facility, then it would be possible to

distribute the files across all the devices. If this is true then some kind of distributed search might be appropriate since it is unwise to coordinate this centrally.

It is important to note here that hiding details as to the location of files from the user, adds location transparency but this might be confusing to the user if files seem to disappear/reappear depending upon which devices are currently available. Therefore, files need to be replicated or appropriate error messages supplied to the user when a device is not available. Decentralized file sharing protocols therefore might be appropriate here since file duplication is easily implemented by such systems. Whatever the implementation, however, such fault tolerance should be available within ensembles with lessons learned from these decentralized approaches.

Another possibility that could be fed into appliance aggregation is small world networks, coming from Stanley Milgram paper in 1967, where he performed a social networking experiment within the United States and found that it only took, on average, 5.5. 'social hops' for messages to traverse through the population of 200 million people. Recently, many file sharing applications have borrowed from this idea and implemented the small-world approach within their applications. This so-called centralized-decentralized approach significantly reduces the number of hops within the network and is implemented within various derivatives of Gnutella e.g. Limewire[15] and a derivative using reflector node concepts, such as KaZaA[16]. You could imagine that this could be relevant in the future for appliance aggregation since such networks could potentially span companies or even towns, cities etc., and would therefore consist of a vast number of devices. Cooltown illustrates this concept in their on-line video [17], within a fusion of appliances, users, desktops, and supercomputer devices interoperating within a wirelessly connected town.

4.2 P2P Middleware

Current P2P applications and protocols are limited to their specific domain. Recently, there has been effort into generalizing these concepts into a set of middleware that allows the application developers to abstract themselves away from the underlying mechanisms of P2P computing. One example is JXTA [18], which was started by Sun Microsystems but is now an open source project. Such an architecture could be useful within the context of appliance aggregation to give a virtual network overlay for the various underlying devices. Briefly, the key JXTA concepts are summarized below.

Project JXTA defines a set of protocols that can be used to construct decentralized peer-to-peer (P2P) applications (but also supports centralized and brokered architectures also). A Peer in JXTA is any networked device that implements one or more of the JXTA protocols. Peers could be appliances sensors, phones, PDAs, PCs, servers and even supercomputers. The JXTA protocols define the way peers discovery and communicate with each other and are specified using XML message formats. Such protocols are therefore programming-language independent and current bindings include Java, C and Python. Each peer operates independently and asynchronously from all other peers, and is uniquely identified by a Peer ID. A peer group is a collection of cooperating peers providing a common set of services. Groups also form a hierarchical parent-child relationship, in which each group has single parent. Groups are analogous to ensembles in appliance aggregation.

There are six JXTA protocols: the *Peer Resolver Protocol* (PRP) is the mechanism by which a peer can send a query to one or more peers, and receive a response (or multiple responses) to the query. The *Peer Discovery Protocol* (PDP) is the mechanism by which a peer can advertise its own resources, and discover the resources from other peers (peer groups, services, pipes and additional peers). The *Peer Information Protocol* (PIP) is the mechanism by which a peer may obtain status information about other peers, such as state, uptime, traffic load, capabilities. The *Pipe Binding Protocol* (PBP) is used to connect pipes between peers. The *Endpoint Routing Protocol* (ERP) is used to route JXTA Messages. Finally, the *Rendezvous Protocol* (RVP) is the mechanism by which peers can subscribe or be a subscriber to a propagation service. Within a Peer Group, peers can be rendezvous peers, or peers that are listening to rendezvous peers. The Rendezvous Protocol allows a Peer to send messages to all the listeners of the service. The RVP

is used by the Peer Resolver Protocol and by the Pipe Binding Protocol in order to propagate messages.

The key concepts within JXTA that could be investigated and learned from are the idea of the *virtual network overlay*, which abstracts the JXTA network from the underlying devices, network protocols or programming languages. Using this overlay, peers are not required to have direct point-to-point network connections between themselves and they can discover each other on the network to form transient or persistent relationships called peer groups. Typically, a peer group is a collection of cooperating peers providing a common set of services, similar to ensembles.

For example, messages in JXTA are transmitted by the use of pipes. Pipes are virtual and a pipe's endpoint can be bound to one or more peer endpoints. The actual network transport used for the delivery of JXTA messages is bound at run-time (late binding). Current bindings include TCP/IP, HTTP and Bluetooth and hence provide a level abstract above the underlying protocols that actually deliver the messages. This approach is insensitive to underlying changes within the structure of the particular network you use and could simplify the development process.

JXME is a lightweight JXTA implementation for mobile devices that could be used to run on appliances. Its goals are as follows:

1. Be *interoperable* with JXTA on desktops and workstations.
2. Provide a *p2p infrastructure* for small devices.
3. Be *simple* and easy to use by developers.
4. Be *small* enough to be used with Cell phones and PDAs.
5. Provide a good *user experience*.
6. Be *CLDC-1.0* and *MIDP-1.0* compliant.

5. Grid Computing

In this section, we consider Grid computing architectures and other distributed architectures within the context of appliance aggregation.

Recently, there has been significant interest in the field of Grid computing. Viewing the Grid as an infrastructure to support "Virtual Organizations" with a single sign-on mechanism is a significant and important step. Further, the recent convergence of Grid Computing and Web Services in the form of the Open Grid Services Architecture (OGSA) [19] has given rise to an enormous drive in this direction by both industrial [20] and academic projects, such as Globus [21]. OGSA uses the notion of a service, which is defined as a "network-enabled entity that provides some capability" [19]. The service paradigm is similar in concept to method calls or sub-routines but is much more coarse grained and therefore makes it a much better abstraction for Computational Grids. The recent development of Grid Services using OGSA is an important step in this direction. A computational Grid environment is typically composed of a number of heterogeneous resources, which may be owned and managed by different administrators. Each computing resource may offer one or more services and each service could be a single application or a collection of applications.

Appliance aggregation frameworks can lend ideas from Grid Computing and could interact within other groups, such as the "Relation of OGSA/Globus and Peer2Peer (OGSAP2P)" P2P research group [22] that intends to migrate OGSA into the P2P world by solving many of the problems that exist in this direction.

6. Jini

Jini is one of a large number of distributed systems architectures, including industry-pervasive systems such as CORBA and DCOM. It is distinguished by being based on Java, and deriving many features purely from this Java basis. There are other Java frameworks from Sun which would appear to overlap Jini, such as Enterprise Java Beans (EJBs). However, whereas EJB's make it easier to build business logic servers, Jini could be used to distribute these services in a *network plug and play'* manner.

In a running Jini system, there are three main players. There is a *service*, such as a printer, a supercomputer running a software service etc. There is a *client* which would like to make use of this service. Thirdly, there is a *lookup service* (service locator) which acts as a broker/trader/locator between services and clients. There is an additional component, and that is a *network* connecting all three of these, and this network will generally be running TCP/IP. (Note that the Jini specification is fairly independent of network protocol, but the only current implementation is on TCP/IP)

Code is moved around between these three pieces, and this is done by *marshalling* the objects. This involves *serializing* the objects in such a way that they can be moved around the network and later reconstituted (*deserialized*) by using included information about the class files as well as instance data.

The scenario of using Jini services is as follows: first, the client uses the lookup server to find the service(s) it wishes to use. The lookup service then returns information to the client (in the form of a Java Proxy) which allows the client to contact the service directly. Thereafter, the client and service exchange information directly and the lookup server is no longer required.
Jini, in concept is very similar to appliance aggregation but lives at a higher level in the stack given in figure 1. Jini lives above the networking level to provide coordination of resources.

7. Taxonomy of appliances and ensembles

7.1 Introduction

The following analysis of ensembles and appliances will help understand:

- The different appliance characteristics, and how they can be classified.
- The different types of ensembles that can be formed based on different scenarios, and environments on which these appliances interact.
- The implications and requirements for the different areas related to appliance aggregation (e.g. networking, security, collaboration, etc),

This section is organized in the following sections - Appliance classification, ensemble classification, ensemble scenarios, and summary.

7.2 Appliance classification

The following classification is based on –

(i) How appliance are connected
(ii) How appliances interact
(iii) Appliance ownership/usage
(iv) And how is appliance access controlled

7.2.1 Connection

Basically this differentiates between appliances that are connected only to an ensemble, and those that also participate in a regular client-server network simultaneously.

- *Stand-alone*. This type of appliance does not have dedicated connections to any other node or appliance. When this appliance is initialized it stands-by until it can be aggregated into an ensemble.
- *Networked*. This type of appliance is connected to a network that is not part of the local ensemble, but it is enabled to participate and be connected to an ensemble when requested. This type of appliance will be like a gateway node (i.e. connected to two networks at some point in time).

7.2.2 Interaction

In this classification, appliances are divided based on the type of interaction that takes place among appliances.

- *Information source (Output)*. Information goes out from the appliance to other appliance or appliances. The destination appliance will be based on the type of information this appliance provides and to whom it is addressed. The information provided could be for public consumption (i.e. Broadcast), for a specific group (i.e. Multicast), or for a particular appliance (i.e. point-to-point).

  Example: As you arrive the movie theater schedules are sent to your palm
  Example: As you arrive to your campus you are warned about high-priority events.
- *Information sink (Input)*. This appliance collects information from its surroundings. Appliance information is collected, as they get closer to the Information sink.

  Example: Badge checkers in a campus
- *Combination of both cases above (Input/Output)*.
- *Appliance that can receive requests from other appliances (Service provider)*.
- *Appliance that can engage in a complicated interaction/workflow with other appliances (Workflow/planning)*.

7.2.3 Ownership/Usage

Identifying the ownership of the appliance is always necessary to be able to validate that the information from that appliance comes from the right party, the interaction with that appliance can be tracked for repudiation purposes, and that in general reputation can be part of the model. As proposed in the ownership model from Appliance Aggregation Architecture (A3) paper [26], we have two types of ownership:

- *User*.
- *Organization/Environmental*.

A3 specifies 5 ways on how appliances can be used - (i) for a ***User,*** these could be *Own* (Owner uses the appliance) and *Borrow* (A user borrows another user's appliance); and (ii) for an ***Organization*** these could be *Share* (parallel usage of appliance), *Control* (Sequential usage), and *Queue* (Requests are queued). Later on, the paper discriminates the appliances owned by an organization and creates a simplified User's model which only deals with two cases: Own and Borrow. This simplifies access control for a particular appliance.

7.2.4 Access control

The way appliances interact can be determine based on how they can access each other's functionality and information. Appliance access control will define the rules on how other appliances can access the appliance's features and information.
Access control to a particular appliance can be based on different criteria like:

- Ownership
- Specific user list
- Role
- Group
- Public/Free
- Type of appliance
-

7.3 Ensemble classification

As described earlier in this paper, an ensemble is defined as "a group of appliances to perform a function greater than their parts". A similar concept called "Smart Spaces" is defined in [26], and as the name implies there is a notion of space or location.
Ensembles will have different requirements depending on different factors like appliances and environment characteristics, location, etc. It is necessary to do an ensemble classification based on these factors to understand such requirements.

We have classified ensembles based on the following:

(i) Physical distance;
(ii) How is ensemble access controlled;
(iii) and Ensemble vulnerabilities.

7.3.1 Physical distance

- *Pure Local*. A pure local ensemble is defined when all the appliances in the ensemble can establish direct communication between them only within a maximum defined distance. Example: All of them use Bluetooth.
- *Local networked*. In a local networked ensemble, all the appliances in the ensemble can establish communication between them only within a maximum defined distance between them, but not all the appliances can establish direct communication between them. This means that some appliances in local can be connected utilizing different communication protocols. For example: My TV has both a HAVi port as well as a Bluetooth port, and it connects to my VCR using HAVi and connects to my Palm using Bluetooth.
- *Distributed*. An ensemble is distributed when one or more appliances are beyond the maximum distance for a local ensemble, and are connected to the ensemble utilizing intermediate/proxy nodes that are part of the local ensemble.
-

7.3.2 Access control

Ensemble formation could be based on factors like authentication and access control. Once an appliance joins an ensemble it will be guaranteed that it can access the ensemble resources. If we look at all the different scenarios exposed in section 3, we can see that there are three main types of ensembles:

- *Individual access*.
- *Group/Membership*.

- *Public limited*.
- *Public*.

7.3.3 Protection

Determining the type of attacks and the degree of exposure for a particular ensemble will depend on different factors like the location of the ensemble, the type of information that can be accessed in the ensemble, the level of service that needs to be provided, among others. These could be classified in the three categories mentioned bellow:

- *Open*. This will be a friendly environment, and access is unprotected.
- *Controlled*. This may not be as friendly environment, but there are mechanisms in place to reduce security risks.
- *Hostile*. There are no means to reduce risk, thus leading to the need of high security measures and mechanisms.

Different mechanisms will be required to ensure that ensembles in different situations can be protected and they can provide the corresponding service.

7.4 Ensemble scenarios

Following are some scenarios on which an ensemble can be created:

7.4.1 Scenario 1. Personal ensembles

These ensembles will mostly happen in a location that is normally used by one person. It would be expected that a lot of appliances that belong to this person normally exist and stay on those locations.

- John's home
- John's office
- John's car
- John’s room

In this scenario the ensemble is considered private and the appliances belong to the same owner. Joining this ensemble is very limited and restricted.

7.4.2 Scenario 2. Organizational ensemble

These ensembles’ appliances belong to an organization/group, and are used by those members of the organization.

- Company’s conference room
- Company’s conference center
- Company’s gym
- Company’s elevator
- Company’s lab
- Private Club
- Universities

- House living room

In these scenarios the ensembles could be considered private for members only. Every member's appliance might be able to join the ensemble of the organization. This does not mean that each member's appliance has access to the other member's appliances.

7.4.3 Scenario 3. Public ensemble for customers only

These ensembles' appliances belong to an organization, but their intended use is for the customers of the organization.

- Hotel's conference rooms
- Hotel's restaurants
- Convention centers
- Movie theater

In these scenarios the ensembles could be considered public for customers only. Every customer's appliance, inside the particular premise, for which he or she has paid, might be able to join that particular ensemble.

7.4.4 Scenario 4. Public ensemble but contained

These ensembles' appliances belong to an organization, and their intended use is for customers that are in-board passengers.

- Subway
- Train
- Airplane
- Bus

In these scenarios the ensembles could be considered public for customers & passengers only. Every customer's appliance that is onboard of the vehicle, and for which he or she has paid a ticket, might be able to join that particular ensemble.

7.4.5 Scenario 5. Public ensemble

These ensembles' appliances may belong to an organization or to a user. Their intended use is for any appliance that gets close enough and wants to participate. This would apply to any type of appliance, and could also be true in any of the other 4 scenarios.

- Restaurant
- Mall
- Supermarket
- Park
- Street

In these scenarios the ensembles are public, the actual appliance's owners will define joining rules. This is the most open case and security risks are high

7.5 Summary

Taxonomy of appliances	**Taxonomy of ensembles**
• Connection • Interaction • Ownership/usage • Access control	• Physical distance • Access control • Protection
Scenarios	
• Personal ensembles • Organizational ensemble • Public ensemble for customers only • Public ensemble but contained • Public ensemble	

8. User Scenarios

8.1 First Scenario

John works in an advertising company and waits for some customers in order to present them his new ideas for their product. He is still working at his office PC when they arrive. The doorman asks their names and who they are visiting and as soon as he inserts the information into the computer, the system informs John for their arrival. He goes to the waiting lounge, welcomes them and brings them to the meeting room. There, when they enter the room, the lights turn on and the coffee maker starts functioning. John brings his watch close to the laptop that exists in the room and his personal desktop with his work appears. As soon as the video-projector turns on, the lights turn down and the presentation begins.

In a question where John is not the qualified person to give the answer, he gives an oral command to the system to connect with Mary who is the one that can inform them responsibly. Mary gives them some basic directions and forwards them the detailed report that they should study. After a while, the presentation ends, the report is saved in everyone's mobile storage media, the video-projector, turns off, the lights turn on and everybody enjoys his coffee before the end of the meeting.

8.2 Second Scenario

While Mr Smith prepares his to-do list in his PDA, he checks the estimate time of his arrival to his office by car. The estimation is possible due to the wireless connection of his PDA and a National Database which updates its data concerning the traffic, through sensors spread in the roads.

Mr Smith finally, prefers to use the metro in order to enjoy a few minutes of morning walking. When he is boarded in the wagon, he decides to read the newspaper. He loads the daily news of Times to his PDA, through the wireless LAN of metro and pays using his credit card. While he is reading, he finds an interesting advertisement and he stores it to his PDA. Following, he chooses an online price comparison with other products of similar specifications.

When he reaches his destination, he walks to work while ordering more info about the advertisement he saw earlier. Arriving to the office, his PDA informs him that, according to the last 10 minutes updated information from the stock house, he has over past the upper bound he has chosen, and he can sell his stocks. So he orders the sale to go ahead and starts his work.

8.3 Third Scenario

Dave is a doctor at a hospital. He starts his day by visiting his patients, a procedure that requires the patient's history, that Dave invokes from the hospital database via his PDA with a wireless connection. He gives the nurses the appropriate instructions, determines the medication while the patient's file is automatically updated for future use. The system searches the medicine stock and sends automatically order to the pharmacy company in case of shortage.

While Dave continues his activity, in another room an emergency occurs. In a few seconds, a call appears on his PDA accompanied with the patient's symptoms and history. Dave, on his way to the patient, he weighs that the patient needs to be transferred to another room, equipped with more sophisticated health machinery, so he sends a message asking the personnel to prepare that room.

As soon as the situation gets under control he gives an order to have some microbiologic examinations and to be informed for the results the minute they are ready. After a while, the results appear on his PDA screen and the doctor can continue with his diagnosis and decide the appropriate actions to treat the patient's health problem.

9. Security Considerations

Appliance Aggregation is about sharing data, code and services and hence the appliance domain has the same security considerations as other technologies and more. Securyt and functionality of an ensemble are directly related – the more ubiquitous and the more functionally rich an ensemble becomes, the more security considerations will need to be effected. A very generic device which configures itself to a multitude of tasks would expect a very secure infrastructure to protect itself and to protect others.

One could simplify the requirements by focusing on appliance aggregations which primarily address personal appliances and appliances in local rather than those on the Web. If we consider only local links, the security is much higher. But as the devices and applications become more compelling, the devices would be connected to the web and thus we would have to consider the internet component.

The following sections detail some of the security considerations:

9.1 Trust and Identity

Establishing trust between appliances and service providers is a basic requirement for the appagg domain. As the most of the interactions would be between the appliances, identity of the appliances is crucial. The authentication and identity is important especially for ah-hoc communication and collaboration - we will face issues of transitive trust and brokered trust in this space. Establishing trust and discovering identity in a self configuring, self organizing network scenario has a lot of security implications.

9.2 Channel Security

Channels in the appliance aggregation space are asymmetric and partially connected; they also could be asynchronous. Establishing secure communication channels under these circumstances while maintaining performance is a key consideration. What makes this even more challenging is the fact that we cannot assume full cryptographic capabilities in the devices – may be we should. The appliances would be used for transactions and would exchange money and so, confidentiality as well as transactional integrity is required.

9.3 Executable code and data integrity

As the appliances depend on mobile executable code for achieving the functionalities, all the security measures required to assure authenticity of code (code signing, authorization based on granular permissions depending on the source and functionality,…) are of importance. The concept of shared state, preference and cache brings another set of integrity considerations. The data integrity also is an important issue in the cases where content aggregation is being performed.

9.4 Content security/privacy

One of the challenges in this space is the DRM. Personal devices and appaggs would be used to play, share and edit multimedia – songs, video and other contents. So this domain needs to consider the implications of content privacy, license management, delegation of rights et al. The content security also applies to domains where regulations are if effect – like the healthcare. Ensembles would need to understand and implement role based access controls and similar paradigms.

9.5 Anonymity/Privacy

Issues like anonymity/privacy of hosted and downloaded content also are relevant here. A user may not want anyone or any service provider to know about his or
her involvement in the system. With a central server, it is difficult to ensure anonymity because the server will typically be able to identify the client, at least by Internet address. By employing a P2P structure in which activities are performed locally, users can avoid having to provide any information about themselves to any-one else. FreeNet is a prime example of how anonymity can be built into a P2P application. It uses a forwarding scheme for messages to ensure that the original requestor of a service cannot be tracked. It increases anonymity by using probabilistic algorithms so that origins cannot be easily tracked by analyzing network traffic. Another idea is that May be the devices themselves would go thru anonymizers and other proxies.

9.6 Protection

Another security issue to be addressed by devices/appliances that are capable of establishing bi-directional communication channels with other devices to form ensembles is on attacks. It is necessary to understand what type of mechanisms exist (if-any) or would exist (if we can predict) to protect devices of this nature. It is very likely that the underlying technologies will consider this and provide solutions to these problems, however, we will need to include support for such a technology, so the authorized devices can join an ensemble, and the authorized users can use the ensemble, yet the unauthorized users/devices can't access or disrupt the operation of an ensemble (immune to DOS attacks).

Author Information

Dr Ian Taylor
Department of Computer Science,
Cardiff University, Dept. of Computer Science,
PO Box 916, Cardiff CF24 3XF, UK.
Tel: +44 2920 875032, Fax:029 20 874056
Email: i.j.taylor@cs.cardiff.ac.uk

Ass. Prof. Dimitris Lioupis
Dept. of Computer Engineering and Informatics,
University of Patras, Rio, Patras, Greece.
Tel: +30 2610997750, Fax: +30 2610991909
mobile:+30 6972551122

email: lioupis@cti.gr

Dejan Milojicic,
HP Labs
Page Mill Road, Palo Alto CA, 94304, USA
Tel: +1 650 236 2906, Fax: +1 650 857 7029
Email: dejan@hpl.hp.com

Professor Ashley D. Lloyd
Curtin Business School
GPO Box U1987
Perth
Western Australia 6845
Tel: +61 8 9266 7198, Fax: +61 8 9258 9298
Email: Ashley@curtin.edu.au

Krishna Sankar
Cisco Systems
170, W.Tasman Drive,
San Jose, CA-95134
Email: ksankar@cisco.com
Tel: 408.853.8475

Sergio Mendiola
Oracle Corporation
500 Oracle Parkway
Redwood Shores, CA 94065 USA
Email: sergio.mendiola@oracle.com
Tel: 650.506.5963

Glossary

Recommended but not required.

Intellectual Property Statement

Full Copyright Notice

References

Binford, L. R. (1979) Organization and formation processes: looking at curated technologies. Journal of Anthropological Research 35: 255-273.

1. Foster, I. and Kesselman, C. (1999) Computational Grids, in Foster, I. and Kesselman, C. The Grid: Blueprint for a new computing infrastructure. Morgan Kaufmann: San Francisco.
2. Hughes, T.P. Edison and electric light, in Mackenzie D. And Wajcman J. Eds. (1999) The Social Shaping of Technology. Open University Press: Philadephia.
3. Dagobert D. Runes (editor), The Diary and Sundry Observations of Thomas Alva Edison, Philosophical Library, New York, 1948
4. Ratliff, J.M. (2002) “NTT DoCoMo and Its I-mode Success: Origins and Implications” California Management Review, 44(3), pp.55-71
5. Douglas,M.1982. In the active voice London: Keegan Paul
6. Law, J. and Bijker, W. E. (1997) Postscript: Technology, Stability, and Social Theory, in Law, J. and Bijker, W.E. Eds. Shaping Technology /Building Society: Studies in sociotechnical change. MIT Press: Cambridge Mass.
7. Levitt, T. (1983) “The globalisation of markets,’’ Harvard Business Review, pp. 92-102.
8. Ohmae, K. (1989), ``Managing in a borderless world'', Harvard Business Review, May/June, pp. 152-61.
9. Rugman, A.M. (2001) “The Myth of Global Strategy” International Marketing Review, 18(6), pp. 583-588.
10. Hofstede, G. (1980) Culture's Consequences: International Differences in Work-related Values, Sage, CA.
11. Trompenaars, F. (1993) Riding the Waves of Culture: Understanding Cultural Diversity in Business, Brealey, London.
12. Palen L. and Salzman M. (2002) Welcome to the Wireless World: Problems Using and Understanding Mobile Telephony, in Brown, B., Green, N. and Harper, R. Eds. Wireless World: Social and Interactional Aspects of the Mobile Age. Springer: London.
13. Churchill E. F. and Wakeford, N. (2002) Framing Mobile Collaborations and Mobile Technologies, in Brown, B., Green, N. and Harper, R. Eds. Wireless World: Social and Interactional Aspects of the Mobile Age. Springer: London.
14. Gnutella: The Gnutella Protocol Specification v0.4, Document Revision 1.2, Clip2, http://www.clip2.com, protocols@clip2.com
15. LimeWire: http://www.limewire.com/

16. KaZaA: http://www.kazaa.com/
17. Colltown Video: http://www.cooltown.com/cooltownhome/cooltown-video.asp
18. JXTA: http://www.jxta.org/
19. The Physiology of the Grid: An Open Grid Services Architecture for Distributed Systems Integration. I. Foster, C. Kesselman, J. Nick, S. Tuecke, Open Grid Service Infrastructure WG, Global Grid Forum, June 22, 2002.
20. IBM and Globus Announce Open Grid Services for Commercial Computing, http://www.ibm.com/news/be/en/2002/02/211.html
21. The Globus Project: http://www.globus.org/
22. OGSAP2P Research Group: http://www.ggf.org/4_GP/ogsap2p.htm
23. Milojicic et al, "Peer-to-Peer Computing", HPL Technical Report. http://www.hpl.hp.com/personal/Dejan_Milojicic/p2p_o.pdf.
24. Milojicic et al. “Appliance Aggregation Architecture,” HPL Technical Report. http://www.hpl.hp.com/personal/Dejan_Milojicic/aaa2.pdf.
25. Satyanarayanan, M. 2001, Pervasive Computing: Vision and Challenges” IEEE Personal Communications, August, 2001