# PROTOCOLS AND ARCHITECTURES ON THE PLATFORM FOR THE DISTRIBUTED APPLICATIONS

Fikret Kasumagic
*e-mail:* fkasumagic@cra.ba
Communications Regulatory Agency of B&H
Vilsonovo setaliste 10, 71000 Sarajevo
Bosnia and Herzegovina

Vlado Glavinic
*e-mail: vlado.glavinic@fer.hr*
Faculty of Electrical Engineering and Computing,
University of Zagreb, Unska 3, 10000 Zagreb
Croatia

*Key words: Distributed application, Platform, Network architecture, Middleware…*

**ABSTRACT**

**The paper outlines the distributed application platform, which exists in the frame of distributed processing environments and implements different kinds of technologies for realisation these applications. On the intermediary platform, between network and application layers, it needs to select right technology. Web-based approach with XML protocol and middleware approach with CORBA architecture were presented. After analyse of several approaches, the way for technology selection was given.**

## I. INTRODUCTION

In the future information society, customers will require tailored ubiquitous services, guaranteed qualities, low costs, and a timely reaction to their changing requirements. In this context the most important issues are information highways and universal personal communications. These approaches address the issue of universal connectivity and global access to information, regardless of the user's current location, or technical or network capabilities. Thus the vision for the future information society is *"information at any time, at any place, in any form, and according to personal preferences"*. [1]

Until recently, the development of infrastructures to support applications in fields such as distributed systems, network and service management, telecommunications, mobile communications, security and databases was performed to a large degree independently of each other. However, the increasing demand for new innovative services combined with the high expectations and requirements of potential service providers, service users and network providers, has resulted in the need to integrate these approaches. Telecommunications and distributed computing will merge into an open distributed processing environment and numerous mixed-media services will be offered and used via network in electronic markets and virtual malls.

## II. DISTRIBUTED PROCESSING ENVIROMENTS (DPE)

Regarding that all new services and applications are distributed and object oriented and whereas that entirely communications are layered, it is very important to ensure such as enviroment that can realise seemless communications and qualitative exchanges of information between end user on the network. That could be possible with building a large distributed processing enviroment(DPE). This DPE is the infrastructure that provide the execution enviroment, including distribution transparencies, for distributed applications in a system. The DPE provides distribution transparent view to its users.The users of DPE are the applications and services designers and developers.

The DPE consists of a collection of nodes that are interconected.The DPE must to support execution all services and applications, including also real time services and applications. Implementation of the DPE have to support applications as developed on RM-ODP (Reference Model- Open DistributedProcessing) concepts and methodology.

In this specific *platform* we find one new part of communications model, which is called *middleware*.

Middleware can be defined as a set of software components that support interconnection of distributed systems and services Middleware are software products that can be positioned between or underneath applications and on top of system software, computing systems and communication network and services.[2] In most cases, middleware is hiding the heterogeneity of the underlying systems, e.g., programming languages, operating systems, computing services and network protocols. From the system developer's point of view, middleware provides a homogeneous view of a well structured, heterogeneous distributed enviroment, where applications and services interoperate through standardized open interfaces.We can see that DPE is an instance of middleware.

DPE has own architecture from the components which are categorized in (Figure1):
DPE Kernel
DPE Object services
DPE Support Tools

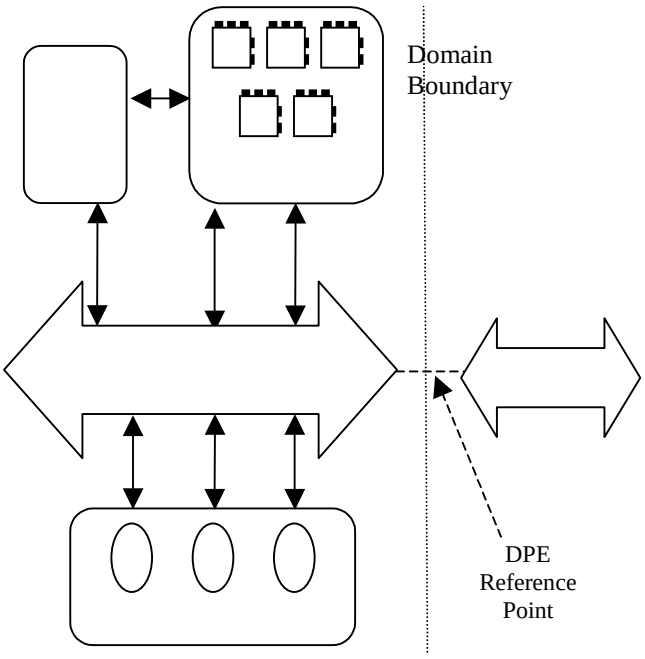


Figure 1. DPE Architecture

*DPE kernel* for object life cycle control and inter-object communication. Object life cycle control includes capabilities to create and delete objects during run time. Inter-object communication provides the mechanisms to support the invocation of operations provided by operational interfaces of remote objects. DPE kernel provides basic and technology independent functions that represent the capabilities of most operating systems, i.e. the ability to execute applications and the abbility to support the communication of applications with each other.
The DPE architecture provides the middleware for interaction between engineering computational objects on remote *DPE nodes*. It also provides tools for diagnosis and configuration for use by network administrators. A DPE node is controlled by DPE kernel. There can be a DPE *refernce point* between any two DPE nodes. Reference point, defined as a set of related specification, is positioned between two business administratives domains.

## MODEL OF DISTRIBUTED APPLICATIONS ARCHITECTURE

The realisation new distributed applications in one open intercommunication systems request a special model of architecture. It defines an architectural model consisting of configurable, re-usable building blocks together with the necessary technological glue to enable the evolution of tomorrow's information systems. This architecture introduces the areas of work paving the way for a future distributed environment in the form of a model consisting of three horizontal planes and one vertical plane (Figure 2.). T*he Network Resource Infrastructure* enabling quality-based data transport, the *Middleware Platform* comprising computing capabilities and middleware services to access and control the infrastructure and to execute various *Distributed Applications* which reside in the top horizontal plane. All three horizontal planes are supported by the vertical plane *System Engineering & Service Design*. [1]This division allows technical solutions in each of the planes to evolve independently, while making appropriate abstractions used to relate the planes to each other. It supports distribution in an inter-domain environment consisting of complex organisational and technological domains. Methodologically, the object-oriented paradigm for distributed applications and systems is the predominant foundation. Object-orientation enables the efficient development, installation, provision, operation and maintenance of information applications and services. It facilitates their design and development from re-usable components and is a prerequisite for desired properties like customisability, adaptability, scalability, re-usability, and autonomy. This kind of architecture allows the development of an infrastructure for the information society from an all-encompassing perspective: from network technology integration to distributed application platforms, and from mobility aspects to system engineering and performance evaluation.

A prerequisite for the development of tomorrow's open service market is a web of interconnected wired as well as wireless networks. That means we need to have a *Communication Platform* on the ground. The goal of the *Network Resource Infrastructure* is the integration of these heterogeneous networks and end-system via high-speed networks as well as via new hardware and software architectures. A strong impact on these efforts has the concepts developed in the Internet community. An integrated management approach based on current and evolving approaches from the ISO/ITU will provide an infrastructure for the management of the networks and resources as well as for co-operative interdomain management.

The *Middleware Platform* is based on an object-oriented architecture supporting the vision of ubiquitous computing and telesensation (computer vision, computer graphics, virtual reality and telecommunications) allowing secure access to existing and future multimedia services.

The *Distributed Applications platform* supports the emerging aspects of user mobility including terminal, personal and session mobility as well as the personalization and interworking of the multimedia

applications and services located in the third plane on the top of the model. The evolution of this Architecture exists on the basis of Open Distributed Processing (ODP) and Telecommunications Information Networking Architecture Consortium (TINA-C). The more practical approach of the Object Management Group's (OMG) Common Object Request Broker Architecture (CORBA) will be used as a main source for the platform development. Telecommunication-specific issues such as subscription, accounting and configuration facilities as well as service creation and composition aspects of the open services market are taken into consideration. Again, an integrated management approach guided by the evolution in the areas of Telecommunications Management Networks (TMN), Internet Management, and Intelligent Networks (IN) will provide an infrastructure for the management of users, resources, and services.

The vertical *System Engineering & Service Design* plane provides mechanisms, tools, and protocols for assuring desired qualities for all entities residing in the

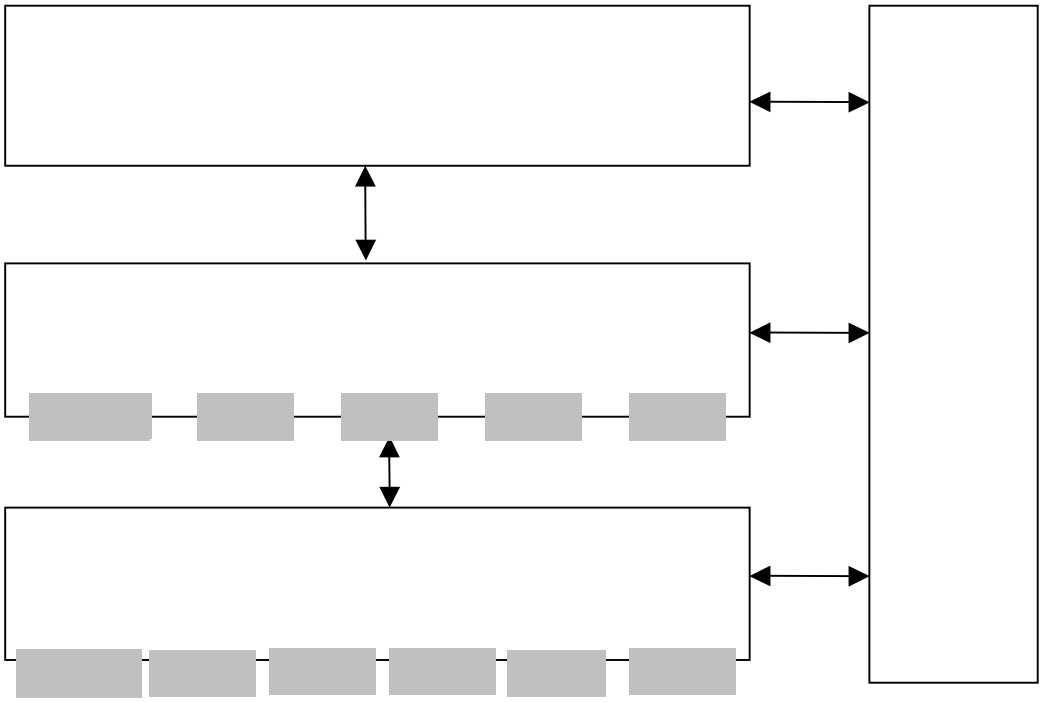

Figure 2. Model of distributed applications architecture

three horizontal planes. It supports the harmonisation of the different models and techniques, which are developed by the various in the information society and which have a strong impact on the implementation of appropriate software components. Additionally the provision of a set of consistent methods and too supporting the complaint development of protocols, services, and applications is a task of this plane. This effort includes both functional and performance aspects and is influenced by the results by ODP and TINA-C, by object-oriented analysis and design methodology, and ISO/ITU activities in the area of formal description techniques and testing methodologies.

## III. INTERMEDIARY PLATFORM

The convergence of the Web and objects technology have led to new software applications and development methods that require unique infrastructure to run effectively [3] Two promising approaches provided by vendors towards the object Web are Microsoft's active server pages (ASP) and Java/ (CORBA). [4]

The ASP technology offers a set of built-in server objects logically related to each other. The HTTP protocol is still used for communications between the client and the server.

This emerging *"object Web"* assumes universal connectivity, broadly distributed environments, and cross platform interoperation of both infrastructure and applications. It represents a next wave of computing and may have a greater impact on the industry then any previous paradigm. The XML definition provides specifications for XML documents and XML document type definition. XML is a key technology for increasing the web's support of complex applications because it gives a richer representation of web information then HTML

On the other hand CORBA allows the creation of server

objects that can be invoked directly by client programs. In this case the communication protocol is the "Internet interoperable object requested broker protocol" (IIOP), provided by the CORBA architecture. *CORBA* currently provide many services including naming, security, transaction and persistence.
A general description of the services, applications and implementation process has been analytically presented. Based on the experience gained from several implementations of Internet applications, an experimental work built upon the new Java/ CORBA architecture was described in contrast to the widely used ASP technology. The Java/ CORBA is recommended for *highly demanded applications*. (regarding own characteristics). The ASP approach is recommended for *small-medium applications* and it facilitates the development process. Furthermore the ASP approach is platform dependent.

## WEB- BASED TECHNOLOGY

Web- based enterprise management (WBEM) specifies of group of technologies, which provide web access to both managed data and managed elements. [5] Technologies that are likely to be utilised for WBEM are: the Common Information Model (CIM), the extensible mark up language (XML) and HTTP. Figure 3. illustrates how these technologies might be integrated.

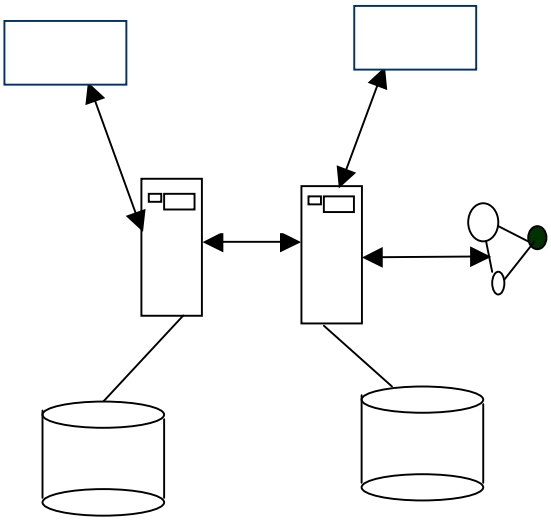

Figure3. Interaction of WBEM enabling technologies

## DISTRIBUTED OBJECT TECHNOLOGY ( CORBA AND DCOM)

The rapid growth of the web combined with developments in high speed and multimedia networks has brought distributed processing into the fore. To simplify the management of software based components, two similar distributed objects models have emerged-CORBA
( Common Object Request Broker Architecture) and DCOM (Distributed Component Object Model ).

Both DCOM and CORBA are based upon the client - server paradigm. To request a service, a client invokes a method implemented by remote object, which acts as the server in the client-server model. The service provided by the server is encapsulated as an object and the interface of an object is described in an Interface Definition Language (IDL). In both DCOM and CORBA, the interaction between a client process and an object server are implemented as object-oriented RPC ( Remote Procedure Call)-style communications.

## IV. THE SELECTION OF PLATFORM TECHNOLOGY

Regarding to the customer services it is possible to make network architecture in several functional layers. Figure 4 shows the Unitary Network's Architecture, which consist of three functional layers offering transport, basic, and co-operative services. [6] The transport service layer consists of standard TCP/IP Internet protocol. The basic service layer includes information resources such as e-mail, file transfer and Web access. The middleware co-operative service layer facilitates the development and development of applications among administrations.

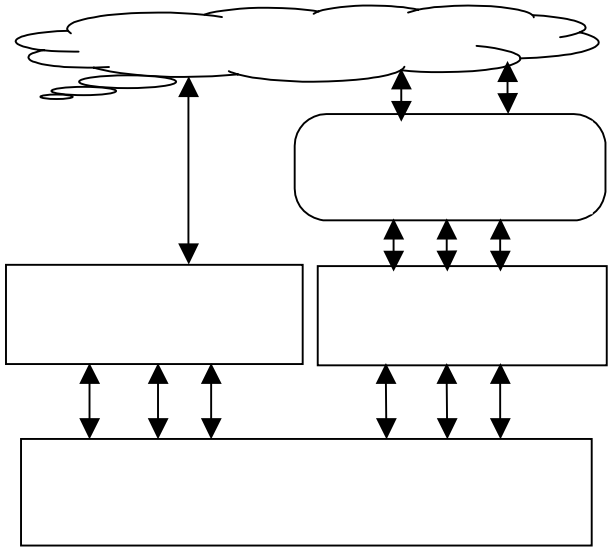

Figure 4. Unitary network architecture

There are different technologies, which are existing on the platform for distributed applications

A *traditional approach* describes data and application services as procedural calls and adopts consolidated technologies such as file transfer, wide spread proprietary protocols, and transaction processing monitors
A *Web -based approach* describes and exports data through XML documents. The exchange uses standard Internet technologies and protocols, mostly HTTP.

A *Middleware -based approach* uses an appropriate interface definition language to specify a collection of business objects and components and deploys co-operative gateways as object/component servers such as CORBA, COM+

On the basis of practically experiences it is clear that applicable technology depends on purpose of use. And type of co-operation. When co-operation exist only of simple *data* exchange, an approach based on *XML* and standard Web technologies seems more suitable.

If co-operation consist of *services*, a *middleware*-based approach is more effective.

Currently, most projects require simple data exchange, and a risk associated with a middleware -based approach is sometimes too high - for example, the failure of some prototype efforts.

However, the increasing popularity of the co-operation paradigm and business process reengineering macro processes has pushed co-operation to evolve from simple data exchange toward the invocation of complex application services.

Eventually the current middleware and Web technologies will probably merge converging toward a standard suite protocol. This protocol suite is very interesting and very often was called as Functional profile. Hence Functional profiles are defined as OSI layer standards' subset specially designed to support particular user requirements [7]

The intermediary platform for distributed applications exist on the border between forth and haigher layers and it seems as the extra zone, one gray zone, in which usage of protocols and architecture were mixed. This is very importanat for efficency of network and quality of interoperability between end users.

Graphic version of this situation is shown on Figure 5.

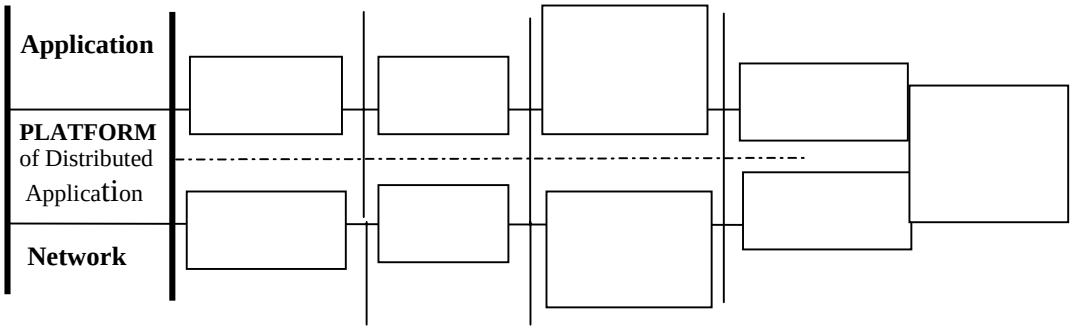


Figure 5. Presentation of Distributed application platform

It is clear that simpler requests use the protocols (web based tecnology-XML) whilst complex purposes use appropriate middleware architecture ( CORBA,DCOM). In some mixed cases this two technologies were replaced with protocol suits, i.e. Functional profiles. Regarding to the importance of this components in the frame of communication Reference Model,it could be conclude that this platform or zone make the acquisition or the some kind of extension of communication Reference Model. It seems as a new virtual layer which work by distributed applications.

## VI. CONCLUSION

The customer requests for quality application realisation need to have good selection of protocols or architectures on the distributed appluication platform. In this article it was described, which kind of platform technologies and which way for approach selection we can use in these cases. It was shown that selection depends of various kind of co-operation between network and applications layers

## ACKNOWLEDGMENTS

This work has been partly carried out within the project 036033 Architectural Elements for Regional Information Infrastructures jointly founded by the Ministry of Science and Technology of the Republic of Croatia and Istrian Country.

## REFERENCES:

1. Perspectives and Y-The Solution, GMD FOKUS, Research Institute for Open Communication System, Berlin, 11/2000
2. Draft new ITU-T REC.Z.600 - Distributed Processing Environment Architecture, COM 10-28-E, Study Group 10-Contribution 28, July 2000
3. C. McFall, An object infrastructure for Internet Middleware, IBM on Component Broker, IEEE Internet Computing, March-April 1998
4. A. Michalas et.al., A comparasion of multimedia application development platforms toward the object web, Computer Standards & Interfaces 22(2000)13-26
5. A. King, R. Hunt, Protocols and architecture for managing TCP/IP network infrastructures, Computer Communications 23(2000) 1558-1572
6. M. Mecella, C. Batini, Enabling Italian E-Government through a Cooperative Architecture, IEEE Computer, February 2001
7. J. Gadre, C. Roher, C. Summers, S. Symington, A COS Study of OSI Interoperability, Computer Standards & Interfaces 9 (1988/89) 217-237