

Network Working Group
Request for Comments: 2552
Category: Informational

M. Blinov
M. Bessonov
C. Clissmann
Teltec UCD-CS
Ireland
April 1999

Architecture for Information Brokerage in the ACTS Project GAIA

Status of this Memo

This memo provides information for the Internet community. It does not specify an Internet standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright (C) The Internet Society (1999). All Rights Reserved.

Abstract

This memo introduces a domain and supplier independent generic architecture for information brokerage, designed as part of the ACTS project GAIA (Generic Architecture for Information Availability).

1. Introduction

Today a huge number of goods and services are offered on the electronic market by a large, and ever-increasing, number of suppliers. However, there is still no efficient way for a customer to find a product or information, he/she is interested in and a supplier that can provide that product. Customers and suppliers already can not deal with the quantity of available information by themselves. The high heterogeneity of existing protocols, formats, and underlying networks also limits development of the electronic market.

This results in a demand for brokerage systems that can work as intermediary entities between customers and content suppliers. Brokerage systems assist a customer during the trading process and hide the heterogeneity and distribution of information from the customer. The design of domain and supplier independent generic architecture for such brokerage systems is an objective of the project GAIA (Generic Architecture for Information Availability). GAIA received part funding from the EU ACTS programme for Research and Technological Development. The GAIA brokerage system allows a customer to

- search for a particular "product" (information, content or services) that he/she is interested in
- locate the product, i.e. find supplier(s) from whom the product is available
- order the product from the supplier
- receive delivery of the product by digital means

All these actions are carried out by the broker in response to requests from the customer. Broker services are accessible to the customer through the unified user interface. The customer system does not have to support all the protocols involved in the trading process.

Full specification of the GAIA Architecture is available in the GAIA Standard [1]. The GAIA Standard includes a description of the GAIA Reference Model, GAIA Functional Architecture, GAIA Standard Profiles, and specification of the GAIA interfaces.

This memo does not aim to include the whole text of the GAIA Standard, but to present the basic ideas and concepts of this standard.

The structure of this memo follows the structure of the GAIA Standard:

1. The GAIA Reference Model provides a common basis for the description and specification of brokerage systems, including the GAIA system.
2. The GAIA Functional Architecture defines functional elements of the GAIA Broker, their roles and relationships.
3. The GAIA Brokerage System Interfaces describes internal and external interfaces of the GAIA brokerage system.
4. The GAIA Standard Profiles specifies mandatory and optional profiles to which brokerage systems may conform.

2. The GAIA Reference Model

The Generic Architecture for Information Availability (GAIA) Reference Model outlines the operations and actors involved in finding, ordering, and delivering physical and digital objects and services ("Products") in a global brokered distributed information environment. It provides an overall view of the GAIA environment, and illustrates the respective roles of and relationships between its

components. Further work on standards and frameworks for individual components of the GAIA environment uses the model and terminology provided by the Reference Model.

The GAIA environment is a collection of actors and functions that are combined to support a procedure for information and services discovery, order, and delivery. The actors play roles in the procedure, including initiation and execution of the Actions which are combined to make up the overall transaction. The GAIA architecture provides a standardised and widely applicable framework for the provision and implementation of the brokered search and retrieve applications in a large-scale networked environment.

2.1. GAIA Roles

The GAIA model considers three principal roles that can be played by the GAIA actors. These are the Customer, the Broker and the Supplier. These Roles are shown in Figure 1 below. It also considers a further class of active entities who play supporting roles in the Actions. This latter class is known as GAIA "Helpers" and includes, for example, authentication and payment. The actors are organisations and individuals in the supply chain. Every GAIA actor plays at least one role at any given time.

2.1.1. The Customer

The aim of the Customer is to obtain some Products or information about some Products. The Customer role initiates the GAIA transaction by requesting one or more GAIA Actions, and receives the results of the transaction. The Customer may deal with actors playing either of the other two roles: the Broker or the Supplier. These actors may themselves play the role of the Customer while requesting further services from other Brokers.

2.1.2. The Broker

The Broker provides brokerage services to the Customer and the Supplier. It responds to requests from the Customer to provide Products, or information about Products. The Products that the Broker supplies to the Customer may originate from one or more Suppliers and/or Brokers. The Broker's primary role is to act as a collector and collator of information from a number of different Suppliers, and to supply this information to the Customer, thus obviating the need for the Customer to deal with a variety of Suppliers. A Broker can also be considered to act on behalf of a Supplier, distributing information about the Products available. The actor playing the role of the Broker may play the role of a Supplier

to a Customer or other Broker at the same time. The Broker may play the role of a Customer while interacting with another Broker or with a Supplier.

2.1.3. The Supplier

The Supplier is the source of the Product supplied to the Customer. The Supplier provides the Broker with information about the Product that it can supply. The Supplier may supply its Product directly to the Customer, or to the Broker for forwarding to the Customer. An actor playing the role of a Supplier may also play the role of a Broker. A Supplier may deal with a large number of Brokers and Customers over a number of GAIA transactions.

2.1.4. Helpers

A Helper is an application layer entity playing a supporting role in a GAIA transaction. Helpers provide some service needed in the supply chain, but outside the core functionality of the Broker. Examples include a global directory service, payment service, or authentication service.

The authentication Helper is concerned with facilitating the authentication of one actor to another.

The payment Helper is concerned with supporting a mechanism for payment to one actor by another.

In any given GAIA transaction, there will be one or more Customers (usually one), one or more Brokers, and one or more Suppliers. A description of the Product sought by the Customer is provided by the Customer to the Broker. The Broker may involve other Brokers in the search for the Product. When a Supplier of the Product is discovered by the Broker, this information is included in the response of the Broker to the Customer. During the course of the Action, it may be necessary to call upon the services of one or more Helpers.

2.2. GAIA Actions

Each GAIA transaction is made up of one or more Actions. These Actions are requests by the Customer to the Broker or the Supplier to carry out some operation and to return a response. Four Actions are defined:

- Search
- Locate
- Order
- Deliver

These Actions are shown in Figure 1.

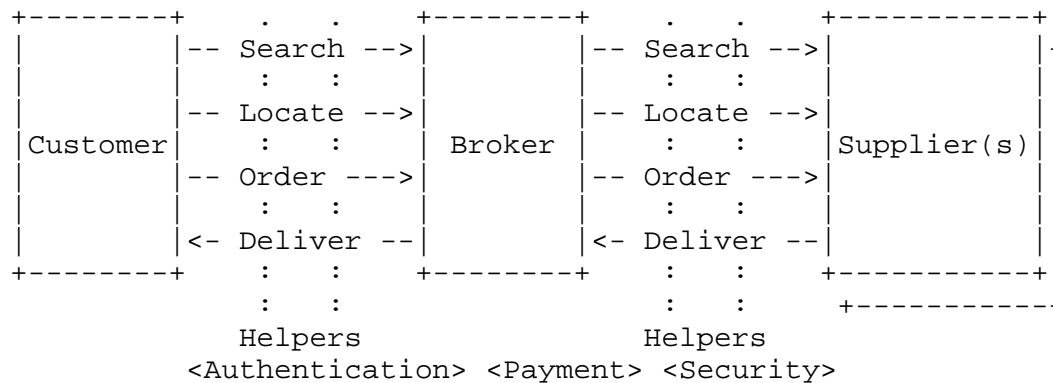


Figure 1 GAIA Roles and Actions

2.2.1. Search

The Search Action is carried out when the Customer asks the Broker to find some information on its behalf. To do this, the Customer provides the Broker with some description of the Product it requires. On the basis of this description, the Broker carries out a search on behalf of the Customer and returns the result. The result of a Search Action is a set of unique identifiers referencing the Products matching the description provided by the Customer.

2.2.2. Locate

The Locate Action is carried out when the Customer asks the Broker to provide it with information regarding the location and source of some Product. To allow the Broker to do this, the Customer provides an unambiguous identification of the Product, which may be the result of a Search Action. The Broker returns information to the Customer about a source or sources for the Product. These data include the Terms of Availability information such as available methods of delivery, time of delivery, costs, etc. However, this information can not be considered final since some special terms and conditions may apply, e.g. discounts for some categories of Customers. The final version of the Terms of Availability is established during the negotiation phase of the Order Action.

2.2.3. Order

The Order Action is carried out when the Customer asks the Broker to obtain a Product on its behalf, or asks the Supplier to sell the Product directly to the Customer. To enable an Order, the Customer provides the Broker/Supplier with Product source information, which

may be a result of a Locate Action. The Order Action consists of a negotiation phase and (possibly) a purchase phase. During the negotiation phase the Customer obtains the quotation that contains the final version of the Terms of Availability for the (batch of) Products he is considering purchasing. If the Customer finds these conditions satisfactory, he commits to the purchase. Alternatively, if the Broker or Supplier supports telepresence services for the human interaction with the Supplier or Broker representatives, these may be used during the negotiations.

2.2.4. Deliver

The Deliver Action is carried out when the Broker provides the Customer with some requested Product. The Product may be information, some physical object, or metadata. The Deliver Action may be in response to an Order Action, a Search Action, or a Locate Action.

While the Actions presented in this section may logically be taken to form an integrated sequence, this is not necessarily the case. Actions may take place independently, rather than as a part of a four-Action whole. For example, Order and Deliver Actions may occur on the basis of information obtained by the Customer using some other mechanism than GAIA Search and Locate Actions.

2.3. GAIA Helper Events

During any of the GAIA Actions outlined above, it may be necessary to carry out some supporting activity. These activities are called GAIA Helper events. They include, for example, authentication and payment. The Helper entities are involved in the GAIA events to provide services, additional to the GAIA Actions, to the GAIA actors.

Authentication

In order to verify the identity of one GAIA actor to another, an authentication exchange may need to take place. This may occur during any of the GAIA Actions. The manner or method of authentication is outside the scope of this document.

Payment

It may be necessary for payment to take place during a GAIA transaction. In this situation, one GAIA actor pays one or more other GAIA actors. The manner or method of payment is outside the scope of this document.

Security

As part of any GAIA Action, it may be necessary to carry out some security operations, such as encryption of data, verification of source and content integrity of Product, or digital signature of some data entity or entities. The particular security services and mechanisms which may be required, or the manner in which they may be provided, is outside the scope of this document.

3. The GAIA Functional Architecture

3.1. The Concept

The GAIA Functional Architecture decomposes the overall functionality of the GAIA Broker into a number of components and describes the roles and relationships of these components, and the manner in which they interoperate.

To work in a heterogeneous environment the GAIA Functional Architecture introduces three levels of abstract elements of the Broker: the Kernel, Functional Unit Managers (FUMs), and Functional Units (FUs) (see Figure 2).

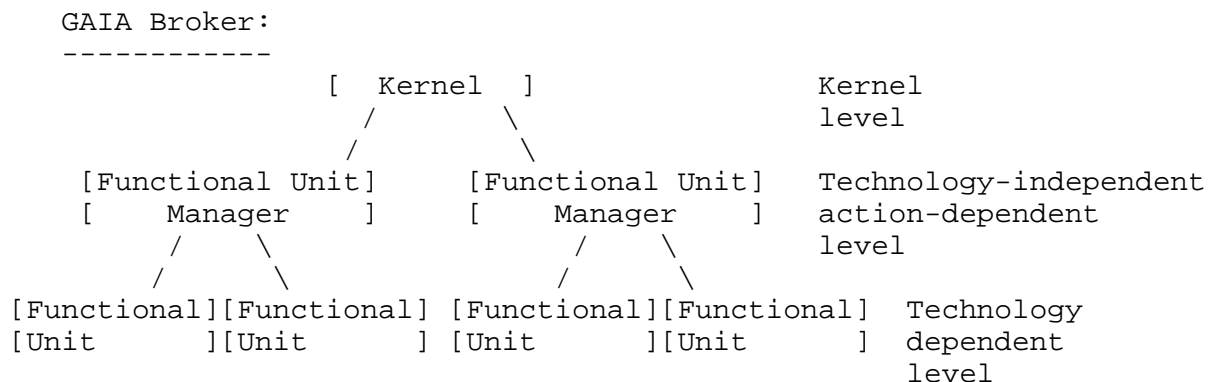


Figure 2 Levels of the architecture

Functional Units are the technology dependent parts of the architecture. They perform required transactions in terms of a particular protocol. All FUs are covered by a technology independent interface. FUs are grouped according to the trading action they participate in, e.g. search FUs or locate FUs. Each group of FUs is governed by the corresponding Functional Unit Manager.

Functional Unit Managers contain technology independent functions for particular actions. To use a particular technology an FUM uses the services of attached FUs. There may be several FUs associated with an FUM, allowing the FUM to operate in different technology contexts.

There is one FUM in the system for every area of functionality, e.g. search, locate, and order. The Kernel is responsible for managing the activity of different FUMs (corresponding to different actions) and synchronising events between them.

The GAIA Functional Architecture establishes relationships between the existing technologies (standards and protocols) that are combined in the GAIA Standard, in the context of a brokerage system. It is to be expected that new technologies will evolve which will be viable alternatives to those selected. The abstract and modular nature of the Functional Architecture allows the replacement of one technology with a new one without disruption to the rest of the brokerage system.

3.2. Functional Units

The brokerage system provides a number of services to its users. These services are supported by the functions of the brokerage system. These include, for example,

- searching
- ordering
- payment

Each of these functions can be provided by a number of different candidate technologies. However, the operations that are required to be carried out remain the same. Regardless of the selected technologies, the functional requirements do not change. The required operations are described in terms of abstract primitives, which can be mapped to the protocol instructions of the technology selected to support the function. A mapping component, called a Functional Unit (FU), is defined for each candidate technology, and converts calls to abstract primitives into protocol instructions. The FU acts as an adaptor between its particular technology and the rest of the brokerage system.

Functional Units are defined for each candidate technology that can be used to fulfil a particular functional need of the brokerage system. A Functional Unit accepts abstract primitive invocations, and maps them to calls to the particular technology to which it is dedicated. The results of these calls are translated into the corresponding abstract primitives and returned by the FU, as shown in Figure 3.

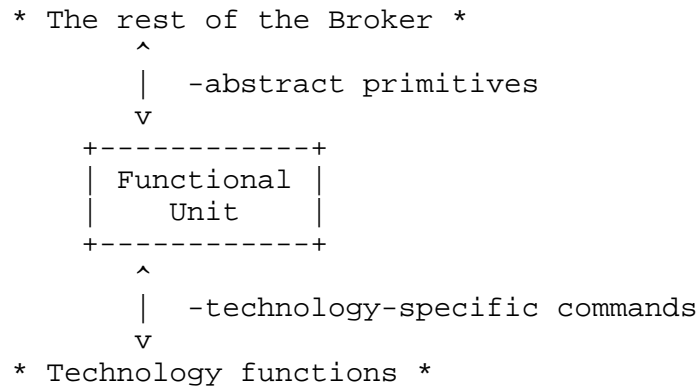


Figure 3 GAIA Functional Unit

3.3. Functional Unit Managers

As noted above, a number of different candidate technologies can be used to fulfil a particular functional requirement of the brokerage system. Depending on the details of the GAIA transaction (underlying network, Customer system capabilities, etc.), different technologies may be more useful during different transactions. As a result, each candidate technology has its own Functional Unit, which is invoked when that particular technology is required.

A number of different Functional Units can exist which fulfil the same functional requirement of the brokerage system. To select the most appropriate FU (and technology), the brokerage system needs to know which is the most useful at any particular time; in general this is the technology supported by the target Supplier system. This is the responsibility of the Functional Unit Manager, or FUM. Each function of the brokerage system has a single FUM, which is invoked using abstract primitives by the Broker Kernel. This FUM selects the most appropriate of the candidate technologies, and calls the corresponding FU (see Figure 4).

The interface between the FUM and the corresponding FUs is defined for every FUM in an open, platform independent, and programming language independent manner. These interfaces do not depend on any particular technology. It allows for configuring the set of technologies supported by the Broker, by attaching different subsets of FUs. If a new technology is to be supported by a Broker, a new FU implementing this technology can be created according to the specification of the interface, and attached to the corresponding FUM.

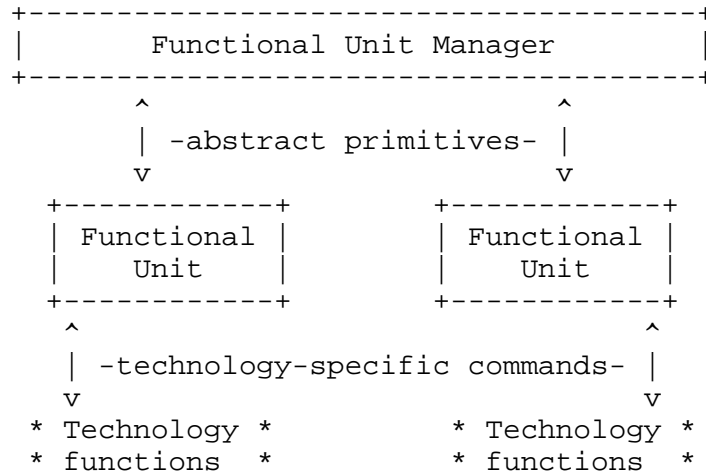


Figure 4 Functional Unit Manager

3.4. The Kernel

The Kernel of the brokerage system acts as a bus for the transmission of abstract primitives between FUMs. Each FUM imports a set of abstract primitives representing those services which the FUM expects to receive from some other part of the system. The services that the FUM is prepared to provide to other elements of the brokerage system are presented in the form of exported abstract primitives. All these abstract primitives are imported from, and exported to, the Kernel (see Figure 5).

The Kernel is also responsible for synchronisation of different actions within a transaction and for maintaining a common context between actions.

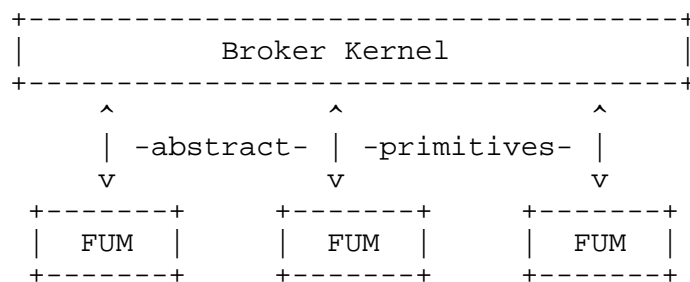


Figure 5 Broker Kernel

3.5. Description of FUMs

The core activities of the brokerage system include:

1. searching for Products that fit a user description
2. sourcing Products the identification of which is known
3. allowing users to order Products
4. delivering information in item format
5. delivering information as a continuous media stream
6. providing a user interface to the brokerage services
7. alerting users as to the availability of information
8. interacting with external directory services
9. authentication of other actors
10. payment operations

Each of these activities is carried out by the corresponding FUM as described below and shown in Figure 6.

Search FUM

The Search FUM accepts requests to carry out a search for Products that fit a particular user description. It returns lists of identifiers of Products that fit the description.

Locate FUM

The Locate FUM accepts Product identifiers and discovers where they may be obtained. It returns lists of Suppliers and locations for the Product.

Order FUM

The Order FUM manages negotiations between a Customer and a Supplier in order that agreement may be reached on the terms of availability of a particular Product or group of Products. Following the negotiation phase, the Order FUM accepts purchase commitments from the Customer and forwards them to the Supplier. It returns a notification of the status of the Order Action.

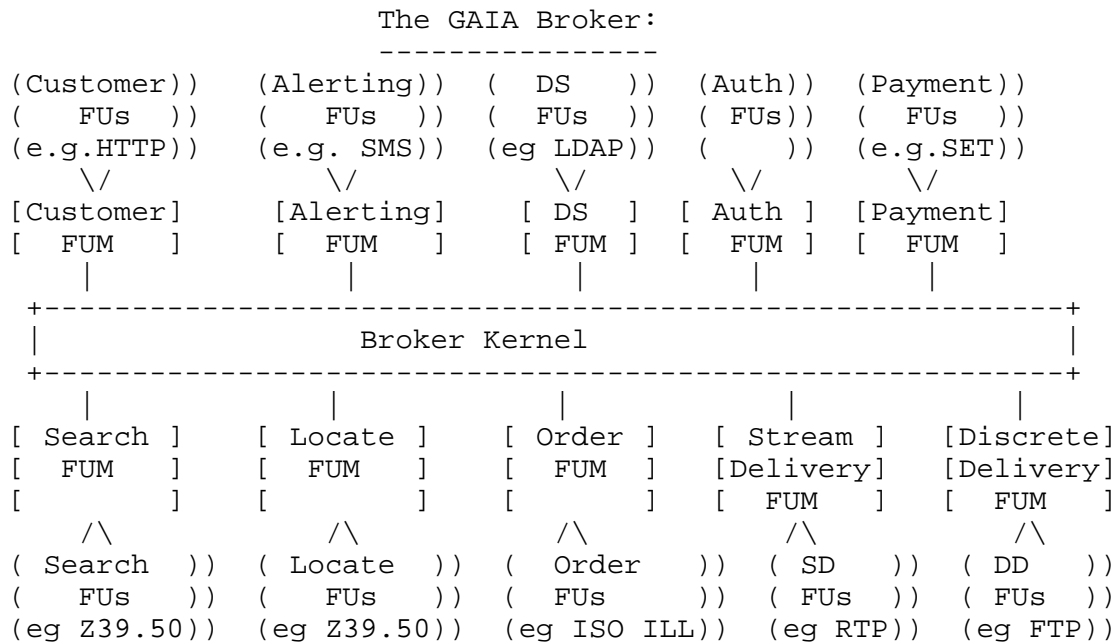


Figure 6 GAIA Functional Architecture

Discrete Delivery FUM

The Discrete Delivery FUM manages the delivery of discrete items to the Customer.

Stream Delivery FUM

The Stream Delivery FUM manages the delivery of real-time multimedia data streams to the Customer.

Customer FUM

The Customer FUM provides an interface to support the Customer's systems interaction with the brokerage system.

Alerting FUM

The Alerting FUM notifies Customers about changes that may interest them.

Directory Services FUM

The Directory Services FUM provides an interface between an external directory service and the brokerage system.

Authentication FUM

The Authentication FUM provides a mechanism that allows a user to prove his identity to the brokerage system.

Payment FUM

The Payment FUM provides a mechanism for payment from one actor to another.

4. GAIA Brokerage System Interfaces

This Chapter describes the internal and external interfaces of the GAIA brokerage system.

4.1. Internal Interfaces

The definition of communication between functional components within the GAIA Broker is based on the OMG CORBA model [2]. Interfaces between components are defined in the IDL language specified by OMG. Interface calls are passed between components by the Object Request Broker (ORB).

The advantage of this approach is that the specifications of the interfaces are platform and programming language independent. These interfaces can be implemented using different programming languages on different platforms. All necessary conversions during interface invocations are transparently performed by an ORB. The CORBA model also allows installing different functional components of the GAIA Broker on different computers connected by a network. Interface calls will be transferred over the network by an ORB transparently for the application.

The specification of the interfaces between the Kernel and FUMs and between each FUM and corresponding FUs is presented in the GAIA Standard [1].

4.2. External protocols

The GAIA Broker can use existing protocols to communicate with other actors. For example, it can use HTTP for interactions with Customers, Z39.50 for search, etc. As described in the GAIA Functional Architecture, support for particular technologies is provided by FUs. A set of supported protocols can be extended by attaching the corresponding new FUs to a Broker. The GAIA Broker can support several protocols for each action. The FUMs will select the most appropriate protocol for a transaction. The more protocols supported by the Broker, the better service it can provide to

Customers and Suppliers.

The GAIA Standard does not limit the set of protocols supported by the Broker. However, for the purpose of interoperability, it specifies several GAIA profiles. These profiles define a common subset of protocols (and a common range of protocol parameters) that Brokers are encouraged to support in order to make communication between GAIA Brokers, and with GAIA-aware Suppliers and Customers, possible.

Existing protocols are not the only way to contact the GAIA Broker. The GAIA interfaces have been designed as a generalisation of existing interfaces and protocols, so they provide more functionality than any particular protocol. To give access to the full functionality of the GAIA Broker, the GAIA Standard allows users (Customers and other Brokers) to directly use the CORBA-defined Customer interface of the GAIA Broker (interface between the Customer FUM and FUs) as shown in Figure 7. In this case, the Customer system gets access to the Customer interface of the Broker using the service of an underlying ORB, and can request operations by calling the corresponding methods of the interface. The Customer interface of the GAIA Broker is specified in the GAIA Standard [1].

Where Customer and Supplier systems are not CORBA-aware, they can communicate with a GAIA Broker using existing protocols. If, however, they can use the service of an ORB, they are encouraged to communicate with a Broker by connecting to its Customer interface. This method allows for avoiding convergence between a particular protocol and the GAIA interface. The former method makes interactions with all existing types of Customers and Suppliers possible using existing and widespread protocols. The later method has been designed to achieve maximum functionality by using native GAIA methods for communication with Customers and Suppliers.

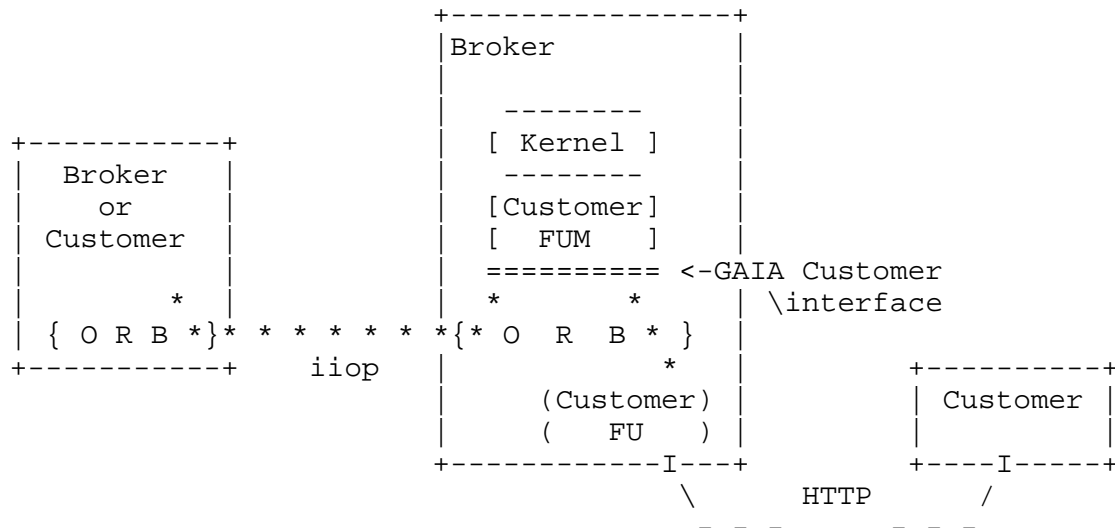


Figure 7 External protocols and the GAIA Customer interface

5. GAIA Standard Profiles

The GAIA Standard defines a number of profiles, which a Broker may support in order to achieve interoperability with other GAIA actors (Customers, Suppliers and other Brokers). The complexity of the profile chosen by a Broker depends on the level and type of service which the Broker wishes to deliver in a GAIA-conformant manner. The higher the level of service that a Broker provides to a Customer, and the greater the length of the supply chain which the Broker wishes to support, the more advanced the profile and/or the greater the number of extension modules the Broker must support.

5.1. Supply Chains

The GAIA profile definition approach is based on the possible types of supply chain that a brokerage system can be a part of.

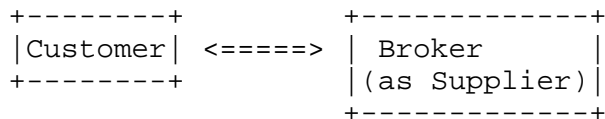
The operations of a brokerage system can be broken into three categories:

- interactions with the Customer
- interactions with other Brokers
- interactions with Suppliers

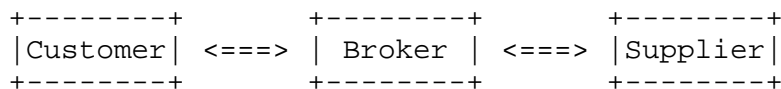
The first and last of these occur at the two ends of a supply chain, while interbroker operations take place at other points in the chain. The supply chain may take a number of different forms:

- a minimal chain, where the Customer and the Broker are the ends of the chain and there are no intervening links. In this case, the Broker plays the role of Supplier to the Customer.
- a three-piece chain, where the Broker deals with the Customer and the Supplier but not with any other Broker.
- a longer chain, with one or more interbroker operations.

Minimal Supply Chain:



3-piece Supply Chain:



Longer Supply Chain:

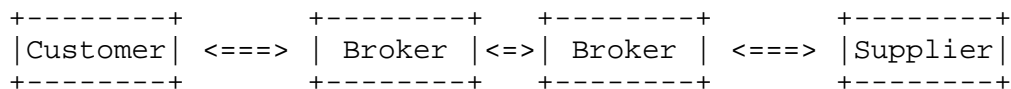


Figure 8 Supply Chains

5.1.1. Minimal Supply Chains

As discussed in the GAIA Reference Model, a GAIA transaction is composed of a number of actions, such as search, order, and delivery. Each transaction is initiated by the Customer who makes a request to the Broker. In the event that the Broker is able to fulfil the request, the transaction involves no other actors.

In this simple case, the GAIA transaction involves the Customer and the Broker. The only protocol which needs to be standardised is that between the Customer and the Broker. This is specified in the GAIA Standard Minimal profile below.

5.1.2. Longer Supply Chains

In the event that the Broker is not able to fulfil a request, the action may be propagated on to other Brokers, with the original Broker playing the Customer role. Each of these Brokers may in turn propagate the request if they cannot fulfil it.

Eventually, if the action is successful, a Supplier will be found who can fulfil the request. The supply chain is thus made up a single Customer, one or more Suppliers, and one or more Brokers.

In order to propagate an action from one Broker to another, a standardised communication protocol must be defined for broker-broker interaction. This is specified in the Basic profile, below. This profile is based on CORBA.

Supplier and Brokers, however, are not obliged to support the Basic profile of the GAIA Standard. They may instead use another, more traditional, protocol such as Z39.50 for discovery, or ISO ILL for ordering. The Extension Modules to the GAIA Standard specify the profiles to be used for various brokerage functions.

5.2. Introduction to the GAIA Standard Profiles and Modules

The profiles specified are

- The Minimal profile, which is the very least to which a GAIA Broker must conform
- The Basic Profile, which allows inter-broker communication
- A number of Extension Modules, which allow the Broker to provide various services, and to interoperate with Suppliers, Brokers and Customers using protocols specified in the modules
- A set of Interface Modules, that defines which particular Functional Unit CORBA interfaces are supported by the Broker

Each Broker must conform at least to the Minimal profile to provide a web-based user interface. In addition, to take part in inter-broker communications, the Basic profile is recommended. For interaction with non-CORBA-aware entities, and for the use of advanced services, there are other modules of the standard to which the Broker may conform. These are denoted "Extension Modules", and they characterise the protocols and standards in a particular area of functionality. A Broker can choose an appropriate set of Extension Modules to conform to according to the functionality it wishes to achieve.

The GAIA Standard specifies all interfaces between FUM and FUs for the GAIA Broker. However, it would be too much to require every Broker to implement all of them. The GAIA Standard decomposes all interfaces into a number of Interface Modules. A Broker can choose a subset of Interface Modules that are more important in its area of operation, and implement interfaces defined in these modules. These interfaces are important only inside the broker system and do not play any role in communication with other GAIA actors. However, a declaration of supported interfaces is important for the administrator to find the areas in which the functionality of the Broker can be extended by attaching GAIA-conformant FUs.

5.3. Minimal Profile

The minimum functionality that a Broker must support will allow it to provide services to the Customer as a part of a minimal chain. In this case, what is required of the Broker is simply a user interface for the Customer. Any further operations take place within the Broker, and so do not come within the scope of the standard.

The Minimal profile requires the Broker to implement a user interface based on the HTTP 1.1 protocol, defined in RFC 2068 [3], and HTML 2.0, defined in RFC 1866 [4]. It means that a Customer should be able to access the basic functionality of the GAIA Broker by using a HTTP 1.1 and HTML 2.0 conformant web-browser.

It should be possible for Customers to locate a GAIA Broker. Thus a GAIA Broker should be registered in a Directory Service using a schema specified in the GAIA Standard [1].

Minimal Profile	
Customer	HTTP 1.1 (server), HTML 2.0

5.4. Basic Profile

While the minimal functionality is sufficient to allow a Broker to function, an important aspect of any GAIA Broker functionality is dealing with other Brokers. The goal of the Basic profile is to achieve federation between Brokers. Every GAIA Broker can use the service of other GAIA Brokers in order to fulfil a request of a Customer. That Broker in turn can use the service of the third GAIA Broker. So every request can be chained by several Brokers. This extends the abilities of every GAIA action (Search, Locate, Order, etc.). Chained transactions are particularly important in the discovery phase of a transaction, where a Broker unable to fulfil a particular information requirement passes on the search to another Broker.

The Basic profile requires the Broker to implement the GAIA Customer interface defined in terms of CORBA. This interface is described in more detail in Section 4.2 above. The Basic profile also requires the Broker to implement interface requestor procedures, i.e. to be able to connect to the Customer interfaces of other Brokers. The ORB used by the Broker should be conformant to the CORBA 2.0 specification [2] and use IIOP protocol for inter-ORB communications [2].

A full specification of the GAIA Customer interface is presented in the GAIA Standard [1].

A GAIA Broker should be able to find other Brokers and Suppliers. It should also allow other participants to find it. Thus a GAIA Broker should support a directory service. The Basic profile includes a directory access protocol for this purpose. The actual choice of protocol is not standardised, because the choice does not influence the success of the Broker's inter-operation with other Brokers. The directory schema, which should be used, is specified in the GAIA Standard.

The Basic profile suggested for a Broker to allow it to interoperate with other GAIA Brokers is as follows.

-----+-----	
Basic Profile	
+-----+-----	
Customer	GAIA Customer interface/IIOP (server)
Search and Locate	GAIA Customer interface/IIOP (client)
(Discovery)	
Order	GAIA Customer interface/IIOP (client)
Directory	Some directory access protocol,
	such as LDAP
-----+-----	

5.5. Extension Modules

In order to allow Brokers to interoperate with other Brokers that do not support the Basic profile, and to allow Brokers to deal with Suppliers and Customers who are not CORBA-aware, as well as to allow delivery of items and data streams via the Broker, other open technologies are suggested as extensions to the Basic and Minimal profiles. These technologies reflect the results of the technology evaluation carried out as part of the project GAIA.

The extra protocols are grouped into Extension Modules. Support of these Extension Modules is optional. A Broker can choose an appropriate set of Extension Modules to conform to according to the functionality it wishes to achieve. There is one Extension Module for each of the functional areas which are not covered by the Basic and Minimal Profiles, and also one Extension Module for each of the existing areas (Customer, Discovery, and Order) to allow the use of protocols other than GAIA abstract primitives.

The following Extension Modules are defined.

- Discovery Extension Module
- Order Extension Module
- Discrete Delivery Extension Module
- Stream Delivery Extension Module
- Security Extension Module
- Payment Extension Module
- Alerting Extension Module
- Customer Discovery Extension Module

5.5.1. Discovery Extension Module

The Discovery Extension Module specifies the technologies to be used in searching for and locating products and services.

This Extension Module requires the Broker to support the client part of the Z39.50 protocol, as defined in [5]. The following subset of the protocol is required:

- Init, Search, and Present services
- GRS-1 record syntax

Z39.50 protocol PDUs should be carried using TCP/IP network protocols.

+-----+-----+-----+-----+-----+-----+	
	Discovery Extension Module
+-----+-----+-----+-----+-----+-----+	
	Searching,
	Locating
	Z39.50 (client)
+-----+-----+-----+-----+-----+-----+	

5.5.2. Order Extension Module

The Order Extension Module specifies the protocols to be used to order products and services from a Supplier.

This Extension Module requires the Broker to support all mandatory services of the client part of the ISO ILL protocol [6]. Basic conformance criteria should be adhered to. ISO ILL protocol PDUs should be carried using TCP/IP network protocols.

- Mapping
RTP payload format for MPEG Audio (MPA), defined in RFC 2250 [13].
- Session control protocol
RTCP, specified in RFC 1889 [12].

This profile provides delivery of high quality audio over networks with non-guaranteed quality of service such as the Internet.

+-----+ Stream Delivery Extension Module +-----+	
Compression	MPEG-2 Audio Layer 3
Data transfer	RTP (client+server)
Mapping	RFC 2250
Session control protocol	RTCP
+-----+	

5.5.5. Security Extension Module

The basic security services required for GAIA are

- Authentication of users, remote servers (both as entity authentication and as bilateral peer-to-peer authentication), senders and receivers in network transactions, as well as the authentication of documents. Authentication is required for three situations: authentication at the user workstation when starting the session, authentication in a local environment (client/server authentication) and authentication in a global, open network (Internet).
- Confidentiality and integrity of all resources transferred over the network or handled locally at application servers and user workstations.
- Control of access to services and resources.
- Non-repudiation of transactions, participants, and sensitive documents.

This module allows a Broker to secure communications with other participants. It provides channel security, authentication, and certificate exchange.

The Security Extension Module specifies the following protocols and algorithms:

- Privacy, integrity, non-repudiation

SSL v3.0 protocol, defined in [14].
 PKCS #7, defined in [15].

- Remote, client/server authentication
 GSS v5, specified in RFC 1508 [16].
- Certification services
 PKIX certification protocol, specified in [17].

-----+-----	
Security Extension Module	
-----+-----	
Privacy, integrity, non-repudiation	SSL v 3.0, PKCS #7
Remote, client/server authentication	GSS v5
Certification services	PKIX certification protocol
-----+-----	

5.5.6. Payment Extension Module

This module allows a Broker to perform electronic payment operations with Customers, Suppliers, and other Brokers. Such operations may take place at any stage during a GAIA transaction, during a Search, Locate, Order, or Deliver Action.

The GAIA Standard does not specify the tariffing or charging model to be used by a Broker; this is considered to be an internal matter. However, when a bill has been agreed, payment must take place in a secure and mutually acceptable manner. The payment procedure specified in the GAIA Standard makes use of the SET specification.

The Payment Extension Module requires a Broker to support SET v1.0 merchant's server and SET certification protocol, specified in [18].

-----+-----	
Payment Extension Module	
-----+-----	
Payment	SET v 1.0 :
	1) CA server for banks
	2) Cardholder wallet
	3) Merchant Server
	4) Payment Gateway server
-----+-----	

5.5.7. Alerting Extension Module

The Alerting Extension Module specifies the protocols to notify Customers about changes that can be interesting for them.

This Extension Module requires the support of the following technologies:

- Internet e-mail, based on SMTP protocol [8], and mail data formats specified in RFC 822 [9]. The Broker should be able to generate and send e-mail messages.
- SMS (Short Message Service), specified in [19].

+-----+-----+-----+-----+-----+-----+		
	Alerting Extension Module	
+-----+-----+-----+-----+-----+-----+		
	Alerting	
	Internet e-mail [SMTP,RFC822] (sender),	
	SMS	
+-----+-----+-----+-----+-----+-----+		

5.5.8. Customer Discovery Extension Module

The Customer Discovery Extension Module allows Z39.50 clients to use the service of the GAIA Broker.

This Extension Module requires the Broker to support the server part of the Z39.50 protocol, as defined in [5]. The following subset of the protocol is required:

- Init, Search, and Present services
- GRS-1 record syntax

Z39.50 protocol PDUs should be carried using TCP/IP network protocols.

+-----+-----+-----+-----+-----+-----+		
	Discovery Extension Module	
+-----+-----+-----+-----+-----+-----+		
	Searching, Locating	Z39.50 (server)
+-----+-----+-----+-----+-----+-----+		

5.6. Interface Modules

For the purpose of conformance, all interfaces between FUMs and FUs, specified by the GAIA Standard, are grouped into GAIA Interface Modules. These modules are recommended to be supported by a GAIA Broker, but they are not mandatory. A Broker can choose a subset of

Interface Modules that are more important in its area of operation, and implement interfaces defined in these modules.

A full specification of the Functional Unit interfaces is presented in the GAIA Standard [1].

The following table defines Interface Modules and specifies which interfaces have to be supported in each of them.

Interface Module	Interfaces that are required to be supported in this module
Search	Search FU interface
Locate	Locate FU interface
Order	Order FU interface
Discrete Delivery	Discrete Delivery FU interface
Stream Delivery	Stream Delivery FU interface
Customer	Customer FU interface
Alerting	Alerting FU interface
Directory Services	Directory Services FU interface
Authentication	Authentication FU interface
Payment	Payment FU interface

6. Acknowledgement

We wish to express our gratitude to all members of the GAIA Consortium for a very lively discussion and their valuable direct and indirect input in the design process of the GAIA Standard.

7. Security Considerations

Security issues related to the electronic brokerage are discussed in Sections 2.1.4, 2.3 and 5.4.5.

8. References

- [1] GAIA Consortium, Deliverable 0403, "GAIA Standard (Final)", December 1998, see also <<http://www.syspace.co.uk/GAIA/>>.
- [2] Object Management Group, "CORBA 2.0 Specification", July 1996, See <<ftp://ftp.omg.org/pub/docs/formal/97-02-25.pdf>>.
- [3] Fielding, R., Gettys, J., Mogul, J., Frystyk, H. and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", RFC 2068, January 1997.

- [4] Berners-Lee, T. and D. Connolly, "Hypertext Markup Language - 2.0", RFC 1866, November 1995.
- [5] ANSI/NISO Z39.50-1995 or ISO 23950 "Information Retrieval: Application Service Definition and Protocol Specification".
- [6] ISO 10161:1997 "Information and documentation -- Open Systems Interconnection -- Interlibrary Loan Application Protocol Specification".
- [7] Postel, J. and J. Reynolds, "File Transfer Protocol", STD 9, RFC 959, October 1985.
- [8] Postel, J., "Simple Mail Transfer Protocol", STD 10, RFC 821, August 1982.
- [9] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822, August 1982.
- [10] Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.

Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.

Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.

Freed, N., Klensin, J., and J. Postel, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", RFC 2048, November 1996.

Freed, N., and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, November 1996.

- [11] ISO/IEC IS 13818 "Information technology -- Coding of moving pictures and associated audio information"

Part 1: Systems

Part 2: Video

Part 3: Audio

Part 4: Conformance testing

Part 5: Software simulation

- [12] Schulzrinne, H., Casner, S., Frederick, R., and V. Jacobson, "RTP: A Transport Protocol for Real-Time Applications", RFC 1889, January 1996.
- [13] Hoffman, D., Fernando, G., Goyal, V. and M. Civanlar, "RTP Payload Format for MPEG1/MPEG2 Video", RFC 2250, January 1998.
- [14] Freier, A., Karlton, P. and P. Kocher, "The SSL Protocol - Version 3.0", Work in Progress, Transport Layer Security Working Group, November 1996, See <http://home.netscape.com/eng/ssl3/index.html>.
- [15] PKCS #7: Cryptographic Message Syntax Standard. Version 1.5, November 1993.
- [16] Linn, J., "Generic Security Service Application Program Interface", RFC 1508, Geer Zolot Associate, September 1993.
- [17] Public-Key Infrastructure (X.509) IETF Working Group, <http://www.ietf.org/html.charters/pkix-charter.html>, July 98.
- [18] "SET Secure Electronic Transaction Specification", Version 1.0, MasterCard and Visa, May 97.
- [19] Digital Cellular Telecommunications System (Phase 2+): Technical Realization of the Short Message Service (SMS) Point-to-Point (PP) (GSM 3.40). Version 5.2.0. European Telecommunications Standards Institute. May 1996.

9. Authors' Addresses

Mikhail Blinov
Computer Science Department
University College Dublin
Belfield, Dublin 4, Ireland

Phone: +353 1-706-2488
Fax: +353 1-269-7262
EMail: mch@net-cs.ucd.ie

Mikhail Bessonov
Computer Science Department
University College Dublin
Belfield, Dublin 4, Ireland

Phone: +353 1-706-2488
Fax: +353 1-269-7262
EMail: mikeb@net-cs.ucd.ie

Ciaran Clissmann
Computer Science Department
University College Dublin
Belfield, Dublin 4, Ireland

Phone: +353 1-706-2488
Fax: +353 1-269-7262
EMail: ciaranc@net-cs.ucd.ie

10. Full Copyright Statement

Copyright (C) The Internet Society (1999). All Rights Reserved.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this paragraph are included on all such copies and derivative works. However, this document itself may not be modified in any way, such as by removing the copyright notice or references to the Internet Society or other Internet organizations, except as needed for the purpose of developing Internet standards in which case the procedures for copyrights defined in the Internet Standards process must be followed, or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by the Internet Society or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and THE INTERNET SOCIETY AND THE INTERNET ENGINEERING TASK FORCE DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

