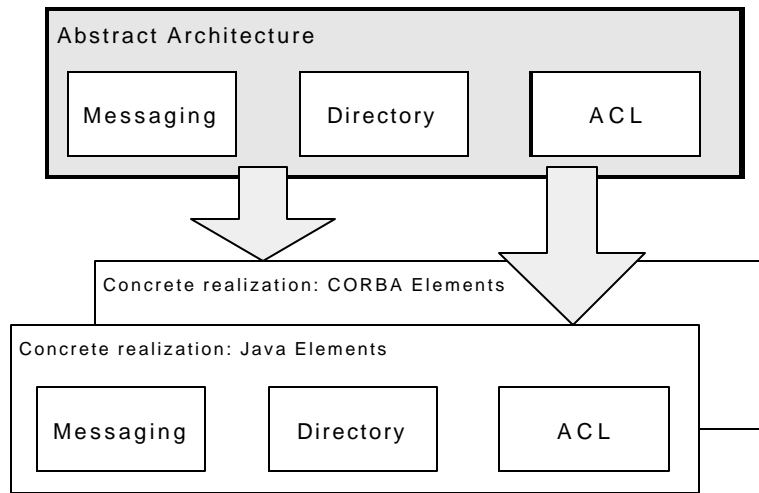


FIPA (Foundation for Intelligent Physical Agents)

Outline

- **FIPA Abstract Architecture**
- **Agents and services**
- **Directory services**
- **Agent Messages**
 - Messages from one agent to another*
 - Messages validity and encryption*
- **Agent Platforms and their services**
 - Agent Management System*
 - Directory Facilitator*
 - Agent Communication Channel*
- **Examples**
 - Ping-Agent*
 - Planner Agent*

FIPA Abstract Architecture



The FIPA abstract architecture:

- cannot be directly implemented. Instead, it forms the basis for concrete architectural specifications.
- does not prohibit the introduction of elements useful to make a good agent system. It merely sets out the minimum required elements.
- also requires the concrete architecture to include mechanisms for agent registration and agent discovery and inter-agent message transfer.

Agents and Services

Agents

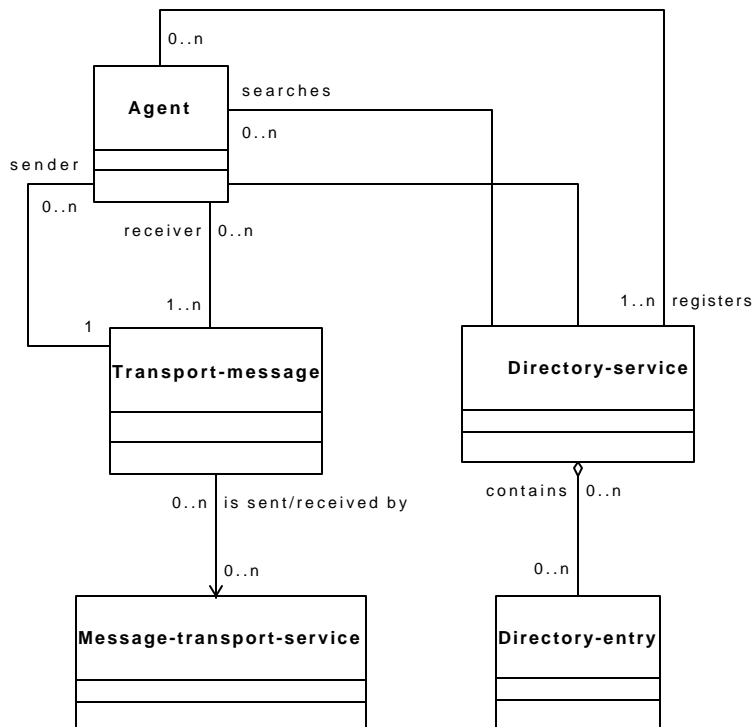
communicate by exchanging messages which represent speech acts, and which are encoded in an **agent-communication-language**.

Services

The Abstract Architecture defines two support services:

- **directory-services**
- **message-transport-services.**

Agents - services relations



Directory Services

Introduction

The basic role of the **directory-service** function is to provide a location where **agents** register **directory-entries**.

Other **agents** can search the **directory-entries** to find agents with which they wish to interact.

The **directory-entry** is a **key-value-tuple** consisting of at least the following two key-value-pairs:

- **Agent-name** = A globally unique name for the agent
- **Locator** = One or more transport-descriptors that describe the transport-type and the transport-specific-address to use to communicate with that agent

the **directory-entry** may contain other **descriptive attributes**:

- the services offered by the agent,
- cost associated with using the agent,
- restrictions on using the agent,
- etc.

Starting an Agent

TRANSPORTS

Each Agent binds itself to one or more **transports**.

In some implementations it will use a **message-transport-service**;

Or

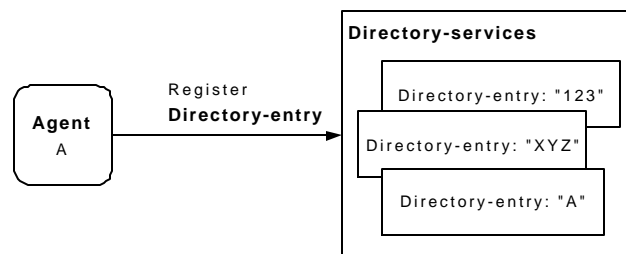
it could contact an ORB, register with an RMI registry, establish itself as a listener on a message queue.

DIRECTORY

The agent registers with the **directory-service** passing to it a **directory-entry**

The **directory-entry** includes:

- the agent's name,
- its transport addressing information,
- optional attributes that describe the service.

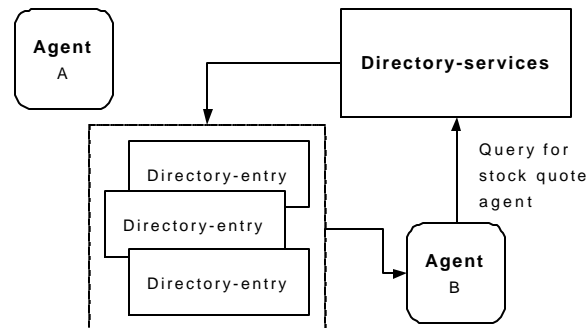


For example, a stock service might advertise itself in abstract terms as {agent-service, "com.dowjones.stockticker"} and {ontology, org.fipa.ontology.stockquote}.

Finding an Agent

INTRODUCTION

Agents can use the **directory-service** to locate other agents with which to communicate.



EXAMPLE

if agent B is seeking stock quotes, it may search for an agent that advertises use of the stockquote ontology.

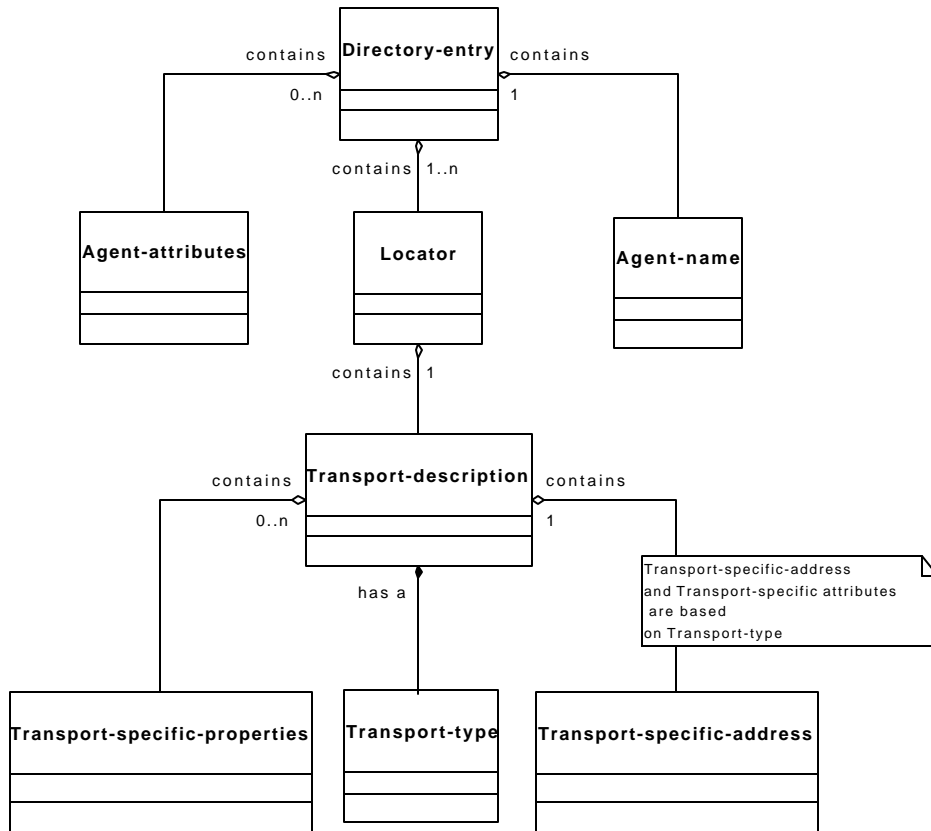
Technically, this would involve searching for a directory-entry that includes the key-value-pair:

`{ontology, {com, dowjones, stockquote}}`.

If it succeeds it will retrieve the **directory-entry** for agent A. It might also retrieve other **directory-entries** for agents that support that ontology.

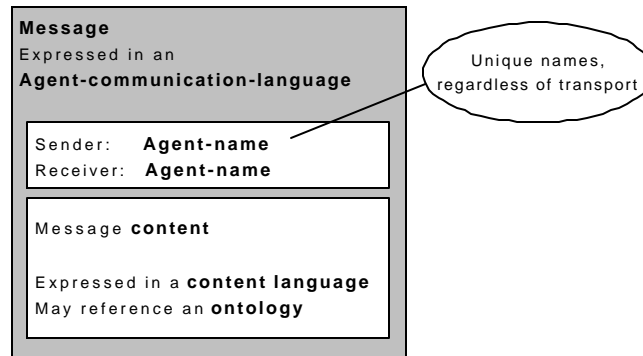
Agent B can then examine the returned **directory-entries** to determine which agent best suits its needs.

DIRECTORY ENTRY RELATIONSHIPS



Agent messages

Message Structure



The structure of a **message** is a **key-value-tuple** and is written in an **agent-communication-language**, such as FIPA ACL.

The **content** of the **message** is expressed in a **content-language**, such as KIF or SL.

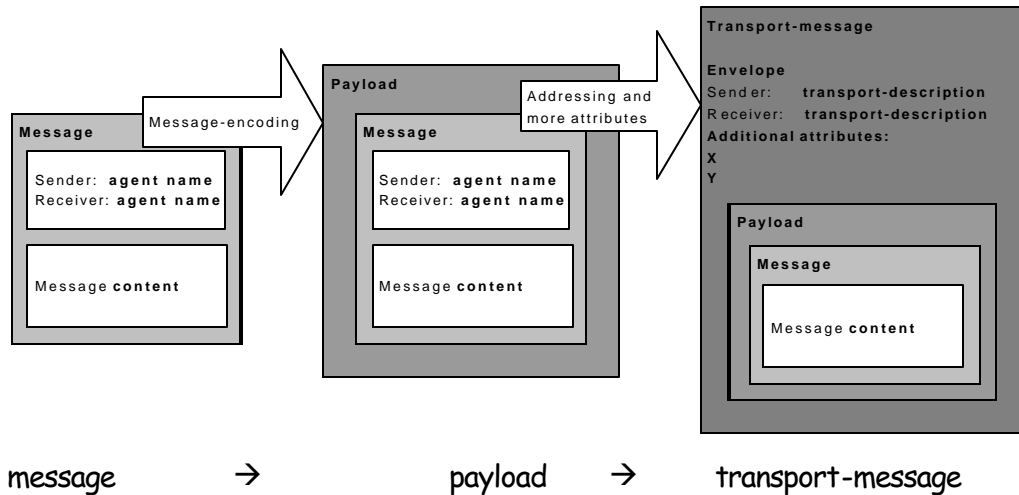
The **content-language** may reference an **ontology**, which grounds the concepts being discussed in the **content**.

The FIPA Agent Communication Language (ACL) is based on speech act theory: **messages are actions**, or communicative acts, as they are intended to perform some action by virtue of being sent.

The specification consists of a set of message types and the description of their pragmatics, that is the effects on the mental attitudes of the sender and receiver agents.

The messages also contain the sender and receiver names, expressed as **agent-names** (globally unique name identifiers for an agent)

Message Transport



When a **message** is sent it is transformed into a **payload**, and included in a **transport-message**.

The **payload** is encoded using the **encoding-representation** appropriate for the transport.

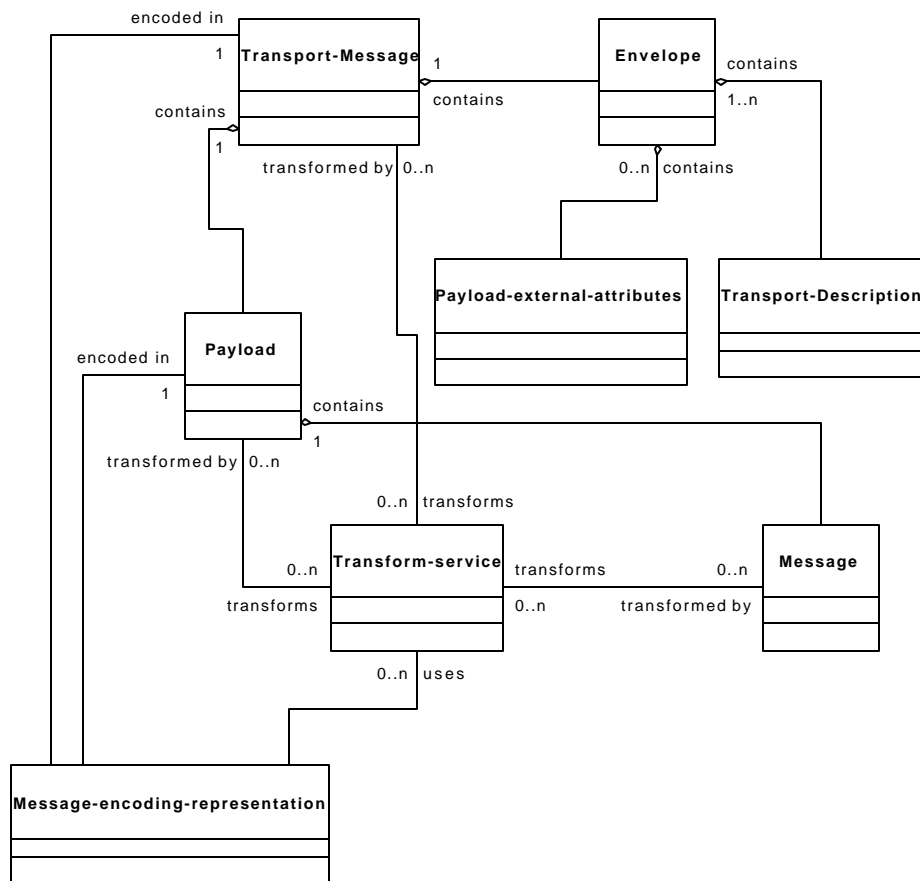
The **transport-message** itself is the **payload** plus the **envelope**.

The **envelope** includes the sender and receiver **transport-descriptions**.

The **transport-descriptions** contain the information about how to send the message (via what transport, to what address, with details about how to utilize the transport).

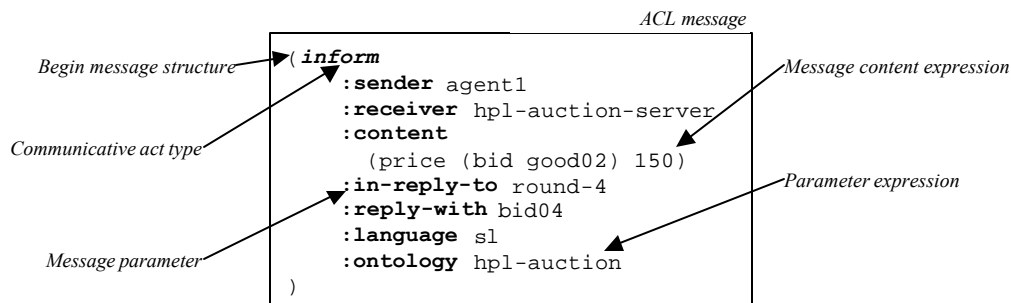
The **envelope** can also contain additional information, such as the **encoding-representation**, data related security, and other realization specific data that needs be visible to the **transport** or recipient.

Transport Message Relationships (UML)



Components of an ACL message

STRUCTURE

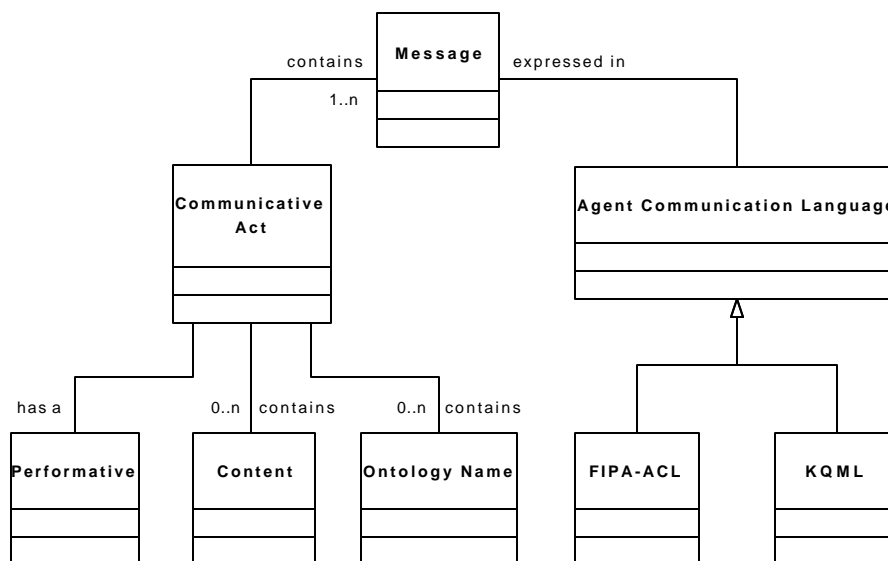


PRE-DEFINED MESSAGE PARAMETERS

Message Parameter:	Meaning:
:sender	Denotes the identity of the sender of the message, i.e. the name of the agent of the communicative act.
:receiver	Denotes the identity of the intended recipient of the message. Note that the recipient may be a single agent name, or a tuple of agent names. This corresponds to the action of multicasting the message. Pragmatically, the semantics of this multicast is that the message is sent to each agent named in the tuple, and that the sender intends each of them to be recipient of the CA encoded in the message. For example, if an agent performs an inform act with a tuple of three agents as receiver, it denotes that the sender intends each of these agent to come to believe the content of the message.
:content	Denotes the content of the message; equivalently denotes the object of the action.
:reply-with	Introduces an expression which will be used by the agent responding to this message to identify the original message. Can be used to follow a conversation thread in a situation where multiple dialogues occur simultaneously. E.g. if agent i sends to agent j a message which contains :reply-with query1, agent j will respond with a message containing :in-reply-to query1.
:in-reply-to	Denotes an expression that references an earlier action to which this message is a reply.

:envelope	Denotes an expression that provides useful information about the message as seen by the message transport service. The content of this parameter is not defined in the specification, but may include time sent, time received, route, etc. The structure of the envelope is a list of keyword value pairs, each of which denotes some aspect of the message service.
:language	Denotes the encoding scheme of the content of the action.
:ontology	Denotes the ontology which is used to give a meaning to the symbols in the content expression.
:reply-by	Denotes a time and/or date expression which indicates a guideline on the latest time by which the sending agent would like a reply.
:protocol	Introduces an identifier which denotes the protocol which the sending agent is employing. The protocol serves to give additional context for the interpretation of the message.
:conversation-id	Introduces an expression which is used to identify an ongoing sequence of communicative acts which together form a conversation. A conversation may be used by an agent to manage its communication strategies and activities. In addition the conversation may provide additional context for the interpretation of the meaning of a message.

Message elements (UML)



Agents Send Messages to Other Agents

Generalities

Each **agent** has an **agent-name**. This **agent-name** is unique and unchangeable.

Each agent has one or more **transport-descriptions**, which are used by other agents to send a **transport-message**.

Each **transport-description** correlates to a particular form of message **transport**, such as IIOP, SMTP, or HTTP.

A **transport-message** is a message expressed in a format (or encoding) that is appropriate to the **transport** being used.

A set of **transport-descriptions** can be held in a **locator**.

Example

Directory entry for agent "ABC"

Agent-name: ABC

Locator:

Transport -type	Transport-specific- address	Transport- specific- property
HTTP	http://www.whiz.net/abc	(none)
SMTP	Abc@lowcal.whiz.net	(none)

Agent-attributes:

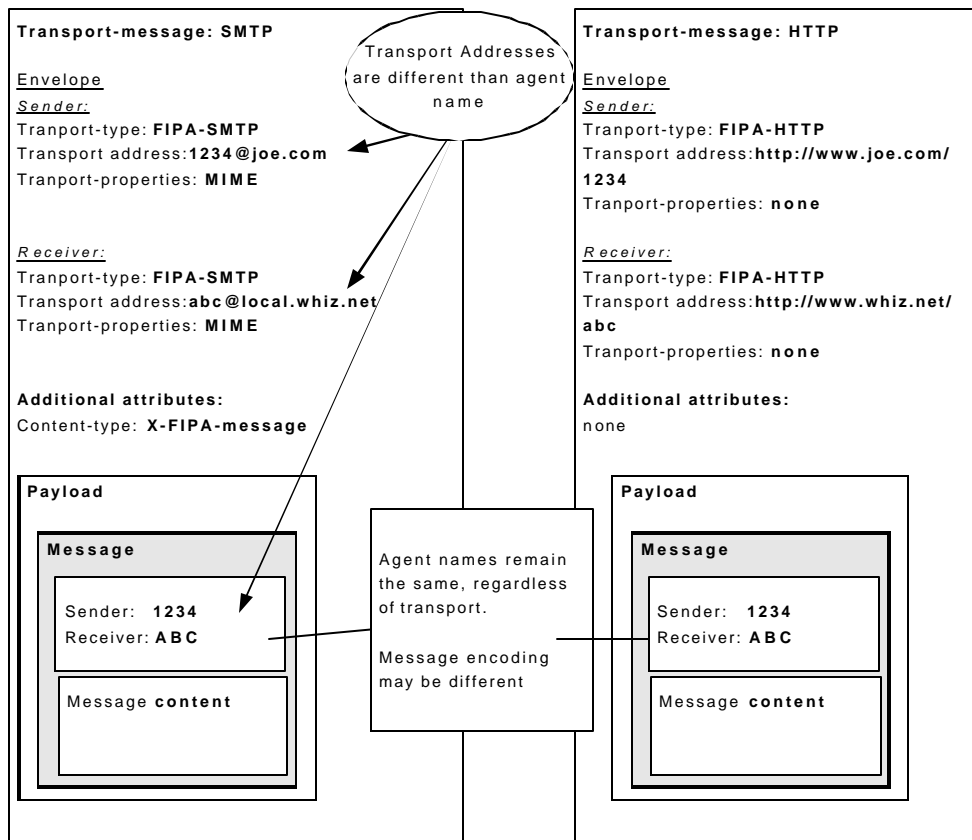
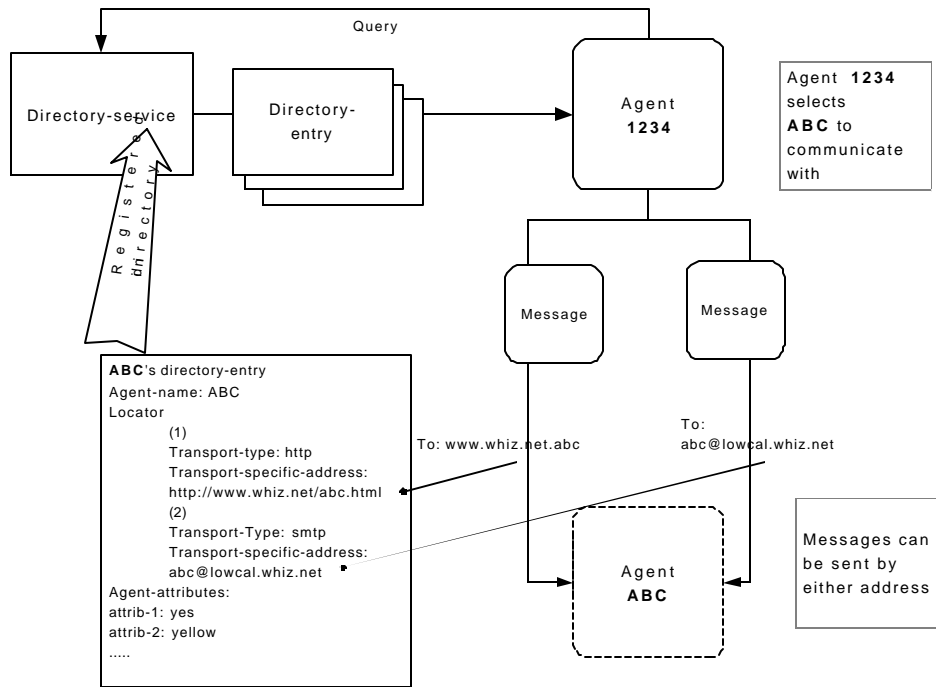
Attrib-1: yes

Attrib-2: yellow

Language: French, German, English

Preferred negotiation: contract-net

Schema



Possible transport services

- CORBA IIOP used as a simple byte stream.
- Remote method invocation, using Java RMI or a CORBA-based interface.
- SMTP email using MIME encoding.
- XML over HTTP.
- Wireless Access Protocol.
- Microsoft Named Pipes.
- A Java Messaging System (JMS) service provider, such as Fiorano.
- Enterprise message systems such as those from IBM and Tibco.

Providing Message Validity and Encryption

Security options

MESSAGE VALIDITY

messages can be sent in such a way that any modification during transmission is identifiable.

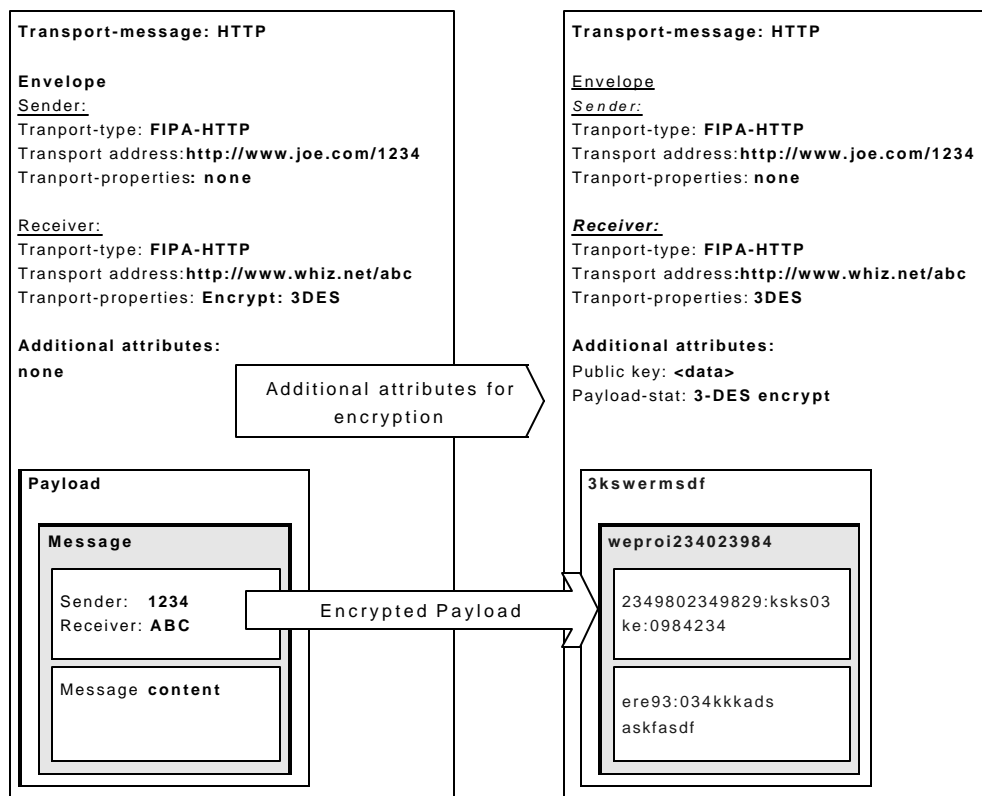
MESSAGE ENCRYPTION.

a message is sent in encrypted form such that non-authorized entities cannot comprehend the message content.

Implementation

ENCODING

through **encoding-representations** and the use of additional attributes in the **envelope**.



EXAMPLE

as the payload is transformed, one of the transformations could be to a digitally encrypted set of data, using a public key and preferred encryption algorithm.

Additional parameters are added to the envelope to indicate these characteristics.

Glossary

AGENT

An Agent is the fundamental actor in a domain. It combines one or more service capabilities into a unified and integrated execution model which can include access to external software, human users and communication facilities.

AGENT MANAGEMENT SYSTEM (AMS)

The Agent Management System is an agent which manages the creation, deletion, suspension, resumption, authentication and migration of agents on the agent platform and provides a "white pages" directory service for all agents resident on an agent platform. It stores the mapping between globally unique agent names (or GUID) and local transport addresses used by the platform.

AGENT COMMUNICATION CHANNEL (ACC) ROUTER

The Agent Communication Channel is an agent which uses information provided by the Agent Management System to route messages between agents within the platform and to agents resident on other platforms.

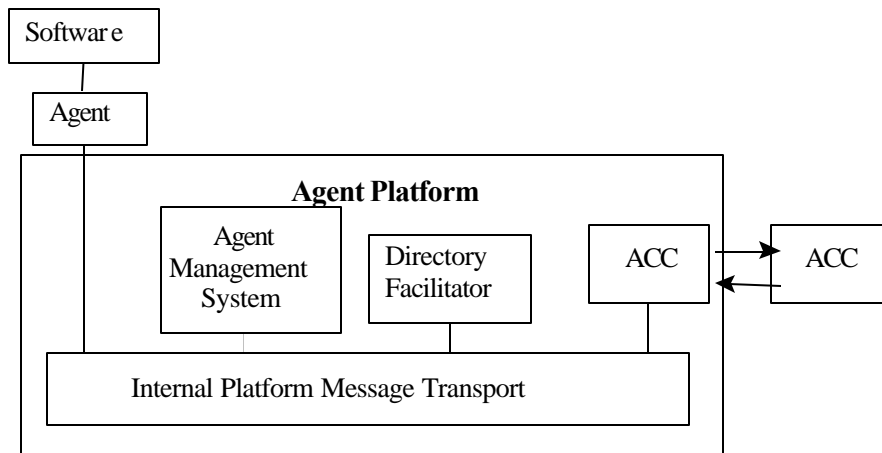
DIRECTORY FACILITATOR (DF)

The Directory facilitator is an agent which provides a "yellow pages" directory service for the agents. It store descriptions of the agents and the services they offer.

AGENT PLATFORM (AP)

An Agent Platform provides an infrastructure in which agents can be deployed. An agent must be registered on a platform in order to interact with other agents on that platform or indeed other platforms. An AP consists of three capability sets ACC, AMS and default Directory Facilitator.

Agent Platforms



Agent Platforms provide the framework within which FIPA Agents exist and operate.

An AP consists of:

- **Directory Facilitator** : The DF provides "yellow pages" services to other agents. The DF is a mandatory, normative agent which is the trusted, benign custodian of the directory within a single domain
- **Agent Management System** : An AMS is a mandatory component of the AP (agent platform). It is an agent which exerts supervisory control over access to and use of the ACC(Agent Communication Channel). Registering an agent with the AMS will result in it automatically being registered with the AP ANS.

The AMS can support the storing, retrieval and forwarding of messages. Only one AMS will exist in a single AP.

An AMS must register with at least one DF.

The ANS (ANS) maintains a directory of logical agent names and their associated transport addresses for an agent platform. The ANS offers "white pages" services to other agents.

- **Agent Communication Channel** : All agents have access to at least one ACC. The ACC is the default communication method that connects all agents within an AP and between AP's.