Trusted Third Party Services in COSM

In the near future we can expect to use global communication networks at negligible communication costs - therefore it is becoming increasingly profitable for commercial network service providers to achieve online availability. The services provided may be implemented as software application modules with operational interfaces that require calling clients to follow a service-specific protocol. Some of these network services may adhere to common access standards (as already well-known from the ISO RDA-standardisation effort for remote database access), others are too individualistic or volatile for a reasonable standardisation. If application level standards are too specific or restrictive, service providers lack significant features which distinguishes them from competitors. Moreover, the process of standardisation itself often hinders a profitable service offer as an "early mover" on an EM. After having established a service access standard, potential competitors have to conform to agreed interface types. There is thus a recognised trade-off between standardisation on the one side, and both time-to-market and competitiveness on the other.

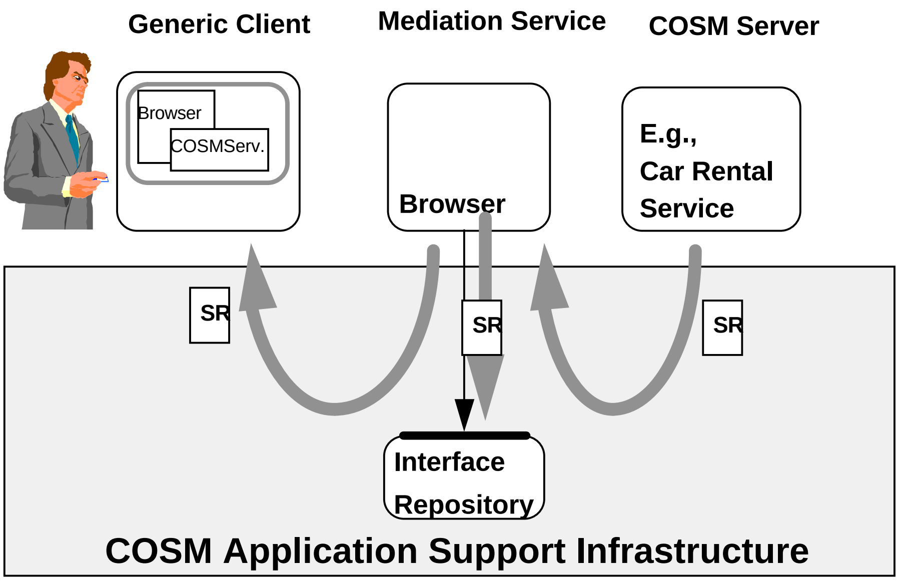

Figure 1: Dynamic Service Access in COSM

In contrast to such standardised services there are so called *unclassified* services, which lack any standardised interface and provide a functionality that is either not specified, or only informally by a text readable for human users - as known from interactive World-Wide-Web (WWW) servers. However, a WWW-based service can be offered immediately after a distinct demand for specific information is recognized and - without any restrictive standardisation - it may provide an individual flavour, increasing its competitiveness compared to other providers. Yet, there is no opportunity of recourse for users who are confronted with a significant difference in service functionality announced (e.g., by hypertext) and the actual service semantics perceived. Due to fraud or misinterpretation this may result in inacceptably high transaction costs - especially in a commercial service environment. Therefore, trusted third party services - as presented further down - play a crucial role on a market for unclassified servies.

Architectural Requirements

Platforms exist to support client/server communication in a generic way: The OMG's Common Object Request Broker Architecture (CORBA) [1] defines *Object Services* which prescribe standardised and functionally dedicated service implementations. In the case of such *classified services*, each implementation must conform to the object service interface definition. On the other hand, independent application level services may be offered via the Object Request Broker which are accessible through an individual, non-standardised interface and which therefore require a dedicated, application-specific client component to set up a type-safe and coherent distributed application. Also compare [2].

In the setting of an EM background, we consider the following requirements as essential for a successful support infrastructure for unclassified services:

- Service providers must not be hindered by an inceased time-to-market due to the standardisation process to make network services available.
- Servers cannot be expected to be accessible via third-party infrastructure providers - like BTX, or CompuServe.
- Users accessing such services can not be overtaxed with keeping control of as many client components as there are services they interact with. Service discovery and the description of their semantics should take place uniformly. Similarly, users cannot be overtaxed with prohibitively high access costs imposed by infrastructure providers.
- Transaction costs of service discovery and utilisation must be minimised in order not to keep potential providers from "going online".
- Existing services shall be embedded by newly offered value-added services, which provide additional functionality based on the first. Access of such of value-added services may take place transparently to the user.

In a service market satisfying the requirements for perfect competition any constellation of value chains can emerge - driven by user demand and the incentive to meet it profitably.

The COSM Architecture

The COSM/TRADE (Common Open Service Market / TRADing Environment) infrastructure allows providers to offer a newly created service without previous standardisation [3], [4]. After a distinct time of maturity and coherence achievement among competing service providers, an a-posteriori standardisation may take place based on a uniform reference implementation which a user may refer to when a contractual obligation is not fulfilled.

The architectural model of COSM comprises the following main components (figure 1):

- The *Generic Client* (GC) supports users in service discovery, service access and in the inspection of service descriptions during run-time. It creates a service-specific user interface which allows the user to enter data and to execute remote procedure calls at the server's site by hitting corresponding button controls in the user interface. Information about structure and layout of the user interface is provided by
- the *Service Representation* (SR) which can be considered as a run-time data object that is transferable between heterogenous computer systems. The SR contains several descriptional components (describing the operational service interface, the GC's user interface layout, the interrelation between user interface and remote procedure invocation, human-understandable descriptions of service semantics, billing information on the charge of procedure invocations, a finite automaton-based description of the order of service invocation, etc.. The SR can be extended by single service providers or groups of them to meet application-specific requirements. However, any coherence

between an SR's specification and the service implementation depends on the service provider's honesty.

- Finally, service providers (server) implement dedicated functionalities and are accessible on-line in the COSM network. They supply SRs to their potential users.

When binding to a server, a GC receives the SR and generates the corresponding user interface. The user is able to inspect the information provided by the SR now. If this information does not specify the kind of service the user is looking for a binding can be released by effecting an *unbind* call to server.

Trusted Third Party Services in COSM

The rest of this contribution focuses on the possibility of enforcing contracts in the COSM context - i.e. in the context of user interaction with COSM services. In COSM, a client/server interaction is considered as a contractual relationship, where servers fulfil a promised service whilst clients charge for service utilization. By interacting with the remote server after binding has taken place, the user implicitly accepts the server's "terms of trade". In the current COSM development, the following two issues are being considered:

1. Since an anonymous interaction between client and server is provided for, funding must also be effected on an anonymous basis.
2. Both parties, client and server, require means to enforce their title to fulfillment if the other party breaches the contract.

Accordingly, two trusted third-party services (TTS) are required to support client/server interaction in COSM: a *bank service* and a *notary service*. The decision to involve one or both of these services may depend on each single binding, therefore the user should be able to call in a TTS on demand at binding time.

The bank server's funding principle is based on electronic currency mechanisms described by [5]. Here, the payor (the client) receives an electronic coin from the bank server and transfers it in an encrypted form to the payee (the server) who is able to verify the validity of the coin.

The notary service may be involved at distinct levels:

- First, it records the identity of each party, still allowing the parties to communicate anonymously.
- Second, the service representation sent from server to client at binding time is recorded. The notary service assures the identity of both versions - one supplied by the server and the other received by the client.
- Finally, remote operation invocations can be logged by the notary service in order to prevent maliciously modified parameter values by senders or receivers.

If both, client and server, intend to involve the notary service at the same level, a binding takes place immediately. If they demand different levels - e.g. the server requires level 2 (SR recording) while the client only demands level 1 (identity recording) - the client's binding request is rejected by the communication run-time system with an indication of the server's level requirement.

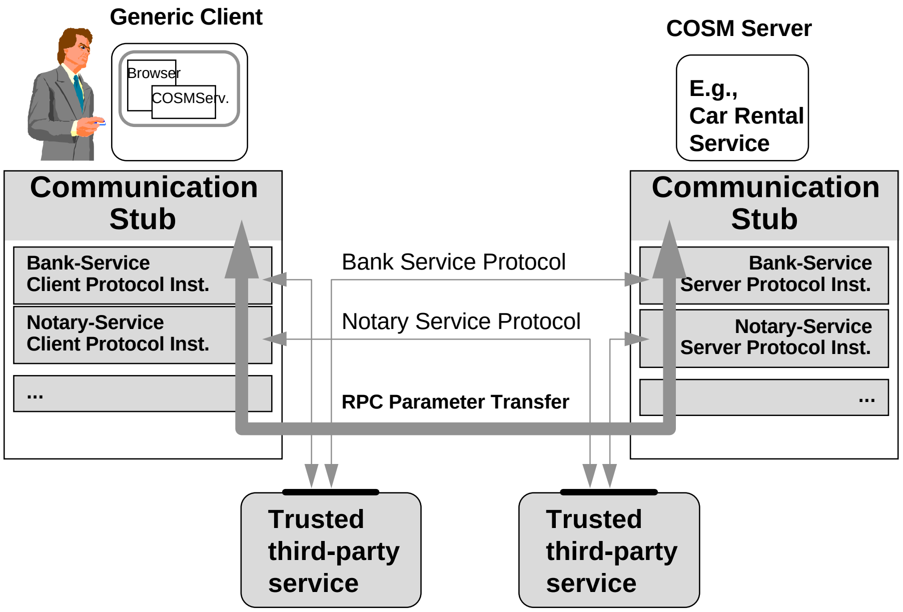

Figure 2: Involving Trusted Third Party Services in a COSM Client/Server Binding

The Technical Viewpoint

In COSM, communication is based on remote procedure calls which are, in turn, implemented on top of OMGs ORB Dynamic Invocation Interface (DII) for two reasons: First, static, stub-based RPC implementations do not provide an adequate means to support dynamically typed operation calls, as they are regularly effected by the Generic Client. Second, additional values can be added to the parameter list when a client application (the GC) locally invokes DII run-time calls. At this interface, several additional protocol instances may intercept the parameter list and extend it by protocol-specific additional ones. In the case of the notary service, the client and server DII run-time systems are extended by protocol instances communicating with the notary service transparently for the application. These protocol instances exchange information like session keys among themselves by dynamically adding these values to the parameter list. At the receiving site, the corresponding protocol instance takes these values out of the parameter list and passes the remaining values over to the application (figure 2). Since different protocol instances may be involved from binding to binding, protocol instances themselves can be dynamically instantiated and released.

State of the Project

Development focuses on the specification of the bank and notary service protocol. A current student project aims at the creation of a heterogeneous set of COSM services which human users can interact with on the basis of generic clients. COSM is being developed on IBM RS/6000 AIX workstations and will be ported to other platforms in the course of this year.

The current COSM prototype does not aim to meet real-world demands as far as completeness is concerned, yet a micro-cosmos is provided that serves as a testbed for implementing general principles and components of electronic markets.

[1] The Common Object Request Broker: Architecture and Specification, OMG Document No. 91.12.1, 1991

[2] ISO/IEC JTC1 SC21 WG7: Trader, Working Document N7047, 1992

[3] M. Merz: "Cooperation Support for an Open Service Market", IFIP / GI 'International Conference on Open Distributed Processing' (ICODP'93), Berlin, August 1993

[4] M. Merz, K. Müller, W. Lamersdorf: "Service Trading and Mediation in Distributed Computing Systems", Proc. IEEE International Conference on Distributed Computing Systems, Los Alamitos 1994

[5] G. Medvinsky, B. C. Neuman: Electronic Currency for the Internet, in: Electronic Markets, October 93, pp 23-24