3GPP TSG_CN / SMG3 **Tdoc NP-000182**
Plenary Meeting #7, Madrid, Spain
13th – 15th March 2000.

Source: **LM Ericsson AB**

Title: **Background information on OSA Release '99**

Agenda item:

Document for: **INFORMATION**

The last few months a lot of work has been acomplished on the UMTS Open Service Architecture Release 99. During corridor discussions on the UMTS Open Service Architecture for Release 99, it became clear that there could be some discussion about an interface which is presently not included in the OSA Release 99 material. The interface has been identified by the OSA group, but has been given second priority so far.
This contribution aims to provide delegates with some background on this interface, and shows that the interface under discussion is largely the same as an interface already worked out in the OSA Ad-hoc output document (3G TS 29.1xx version 0.3.0 part 1).

Background

Figure 1, taken from the stage 2 description (3G TS 23.127) shows the the Open Service Architecture and its main components. Applications get access to Service interface classes via the Framework.

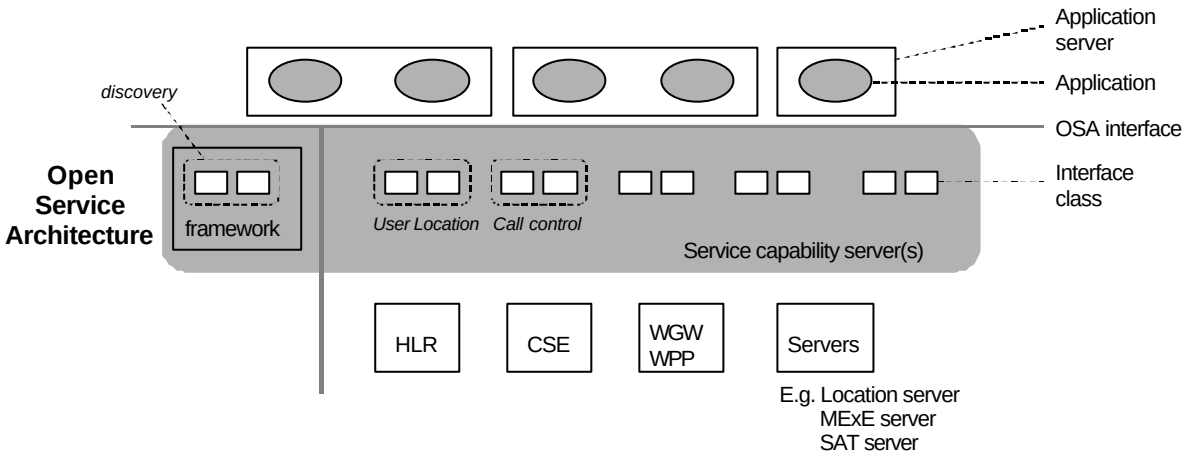

Figure 1 Overview of Open Service Architecture

The Interface between Application and Framework has been completed in the OSA Ad-hoc group, and enables independent service development based on standardised API's. The generic interface between Services and the Framework has not been included in the OSA stage 3 material yet. This interface would allow Services and Framework to be developed and/or operated independently.

This interface has been identified by the OSA groups, but the interface between Applications and Framework was prioritised. Most of the stage 3 work on this interface was done during the last OSA Ad-hoc meeting in Antwerp.

It should however be noted that the work is largely based on the work already done by the Parlay group. In the Parlay group the interface between Framework and Services has already been defined. A large fraction of the interface between Application and Framework is reused between Services and Framework.

Framework Interface Classes

Below an overview of the interface classes relevant for the interface between Application and Framework is listed, as in the table of content of the OSA stage 3 output.

Framework Service Interfaces
The major part of the interface between Services and the Framework will be the same as between Applications and the Framework.
The Services would use all the Framework interfaces defined already for Applications, except the IpServiceDiscovery interface.
Furthermore the Framework interfaces already specified on the Application side will be mirrored to be used on the Service side as well, and thus also do not require additional specification work.

The only additional interface classess to be specified still for 3GPP OSA would be two relative straightforward interface classes. A description from Parlay is attached here to give an impression of the work to be performed still.
Please note that a lot of detailed work has already been done in the Parlay group.

Service Registration: Framework Interface

This API forms part of the interface shown as 5 in Figure 1.
Inherits from the base Parlay interface.

- Before a service can be brokered (discovered, subscribed, accessed, etc.), it has to be registered with the Framework. Services are registered against a particular *service type*. Therefore service types are created first, and then services corresponding to those types are accepted from the Parlay Service Suppliers for registration in the Parlay framework. The framework maintains a repository of service types and registered services.

The Framework service registration-related interfaces are invoked by third party service supplier's administrative applications. They are described below.

Interface Class

<<Interface>> IpFwServiceRegistration
RegisterService(serviceTypeName: in TpServiceTypeName, servicePropertyList: in TpServicePropertyList, serviceID: out TpServiceIDRef) : TpResult announceServiceAvailability(serviceID: in TpServiceID, serviceFactoryRef: in IpServiceRef) : TpResult unregisterService(serviceID: in TpServiceID) : TpResult describeService(serviceID: in TpServiceID, serviceDescription: out TpServiceDescriptionRef) : TpResult

Service Factory Service Interface

Inherits from: IparlayInterface

The IpSvcFactory interface allows the framework to get access to a service manager interface of a Parlay service. It is used during the signServiceAgreement, in order to return a service manager interface reference to the application. Each service has a service manager interface that is the initial point of contact for the Parlay service. E.g., the generic call control service uses the IpCallControlManager interface.

Interface Class