A Study of Issues and Considerations of Service Interaction Management in IMS Architecture

HUI-NA CHUA [a], CHOR-MIN TAN [a], YUXIN HO [b]
[a] Malaysian Research Centre, British Telecommunications Group,
Kuala Lumpur, Malaysia
[b] Faculty of Engineering and Computer Science,
University of Nottingham Malaysia Campus, Malaysia
hui.chua@bt.com, chormin.tan@bt.com, kecy7hyu@exmail.nottingham.edu.my

Abstract: - Though IMS (IP Multimedia subsystem) is aimed to provide an open architecture environment for rapid service creation, it does not necessarily solve all the problems of service interactions and service provisioning. Service Brokering function as currently being studied by the 3GPP [1], is aimed to manage service capabilities interaction between any type of IMS application servers. However, the Service Broker definition in standards does not specify precisely the mechanism of how it achieves the service interaction management. Due to the definition is still not concrete, the Service Broker function is currently implemented in a proprietary manner. In this paper, we examine the evolution of Service Broker functionality proposed in standards and evaluate the existing Service Broker approaches that are proprietarily implemented. From architectural and interaction management aspects, we discuss the issues and considerations of Service Broker function.

Key-Words: - IMS, Service Broker, SOA, orchestration, Service Interaction and invocation

1 Introduction

As the demand of wireless networks surges high, these technologies are needed to be integrated with their complementary characteristics in order to provide the best user connection access regardless of the location and time [2]. IMS is then worked on by 3GPP to sustain high Quality-of-Service (QoS) for multimedia support and interoperability between networks.

One of the objectives of IMS architecture is to create horizontal architecture where different common service capabilities can be invoked, combined and quickly deployed made available to the commercial market. However, though IMS is aimed to provide this open architecture service creation architecture environment, it does not necessarily solve all the problems (further described below) of service interactions and service provisioning.

Service Broker as defined in a 3GPP study report [1] is a functional component that manages service capabilities interaction between any types of IMS application servers. Based on the 3GPP report [1], Service Broker should provide dynamic service interactions and orchestration during runtime in an IMS environment by composing modular service capabilities to create and implement new integrated services. This provision is aimed to enable rapid deployment as it can combine existing common service capabilities from different application servers to deliver new integrated services.

Providing the ability of dynamically integrate service capabilities requires mechanisms to manage and control the conflicts of interactions that potentially may occur between the service capabilities. However, at the time of writing, the definition in [1] does not specify precisely the mechanism of Service Broker function in achieving the interaction management, for instance, the way how it should coordinate multiple invocations of service capabilities between application servers owned by different service providers, and also the way how it should handle the incompatibilities between the invocations. Due to its definition is still not concrete in the standards, presently the Service Broker function is mostly implemented in proprietary manner (either commercial or academic).

For the clarity of this paper in seeding the context in relation to Service Broker, we provide a high level look at the definitions of the individual components that will be used throughout the rest of this paper based on standard specifications [3] [4] [5]:

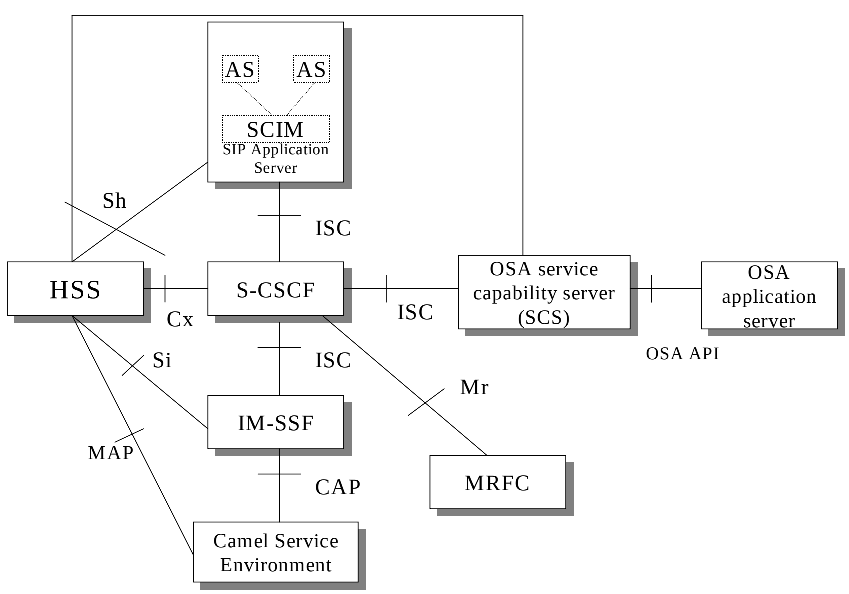
- **Service Capabilities:** Self-contained functionalities that are needed to realise services and can be reused across different application servers. Features offered by service capabilities are accessible via the standardised application interface.
- **Applications:** software components providing services to users by utilising service capability features.
- **Application Interface:** standardised Interface used by applications to access service capability features.
- **Services:** a service is the user experience provided by one or more applications.

In this paper, we will first discuss the evolution of Service Broker functionality and its association with SCIM in standards in section 2. For section 3, the architecture requirements of 3GPP service broker will be elaborated. In section 4, we will identify and compare the existing Service Broker approaches that are presently implemented in a proprietary manner. From architectural and service capability interaction procedural aspects, we discuss the issues and considerations of Service Broker function in the later section.

2 Service Broker and SCIM: The Evolution in Standardization

Service Capability Interaction Manager (SCIM) was proposed in the 3GPP TS23.002 [6] in year 2001 for the purpose of performing "*interaction management*", but nothing is concrete in term of detailed structure and functionality. In [6], it is stated that the SCIM components are "*represented by the 'dotted boxes' inside the SIP Application Server*", and the internal structure of Application Server is beyond the scope of the 3GPP. In other words, the SCIM is a term without 3GPP standardised requirements. However, there exists an Implementation Agreement in Multi-Service Forum for SCIM/Service Broker function for interoperability purposes. Its procedures for 'interaction management' remained unspecified though.

As a consequence, the SCIM in IMS application layer was under-defined and it became the magic box that presumably would answer all the unsolved service interaction questions - which left to be proprietary implementation at present.



HSS: Home Subscriber Server
S-CSCF: Serving Call Session control Function
AS: Application Server
Sh: Reference between AS and HSS
Cx: Reference point between CSCF and HSS
CAP: CAMEL application part
SCIM: Service Capability Interaction Manager
MAP: Mobile application part
MRFC: Media Resource Functional Controller
Si: Reference point between a HSS and IM-SSF
Mr: Reference point between a CSCF and a MRFC
OSA SCS: Parlay-based OSA framework application server to provide 3rd party access to IMS
ISC: IP Multimedia Service Control. Reference point between an application server and CSCF
IM-SSF: IP Multimedia Service Switching Function (for CAMEL network features)

Fig 1. The origin of the SCIM in the architecture [8]

In the 3GPP TS 23.218 [7], it defines SCIM as '*a specialized type of SIP Application Server, the service capability interaction manager (SCIM) which performs the role of interaction management between other application servers*" [7]. Figure 1 is taken from [8], shows the origin of the SCIM in the architecture.

However, in recent year the Service Broker was proposed in the 3GPP R8 as a study item, which is aimed to "*manage the interactions among multiple Application Servers*" [1]. It enables the "*applications to reside in any type of IMS Application Servers including an IMS-SSF, SIP AS, OSA SCS or other (e.g. OMA enabler) or any combination of the above*". Clearly, the intention was to further study the SCIM-like functions via Service Broker.

3 The Study of Service Broker in 3GPP

The service broker functions, as defined in 3GPP TR 23.810 [1], "are to provide an end user a coherent and consistent IP multimedia service experience when multiple IP multimedia applications are invoked in a session". This provision pertains the identification of services invoked, identifying the application sets' order and how to resolve the service capabilities interactions during the session.

The prerequisites [1] of service broker architecture are as below:

- introducing minimal impact towards IMS core network and AS
- enable flexible capability for potential interaction of new applications
- interacts with AS efficiently, without any redundant interaction
- manage interactions of service capabilities between IMS applications, enablers, and other non-IMS applications even for those deployed over different types of application servers
- support the integration with existing Intelligent Network services, enabling the integration of these services with newly defined IMS services as well
- facilitates the service integration across heterogeneous networks, SIP and non-SIP applications and multiple providers
- permits user service personalization and control

3.1 Architecture Reference Model of Service Broker

The architecture reference model for Service Broker is depicted in Figure 2 and 3, showing how service broker is connected to other IMS functional entities [1].

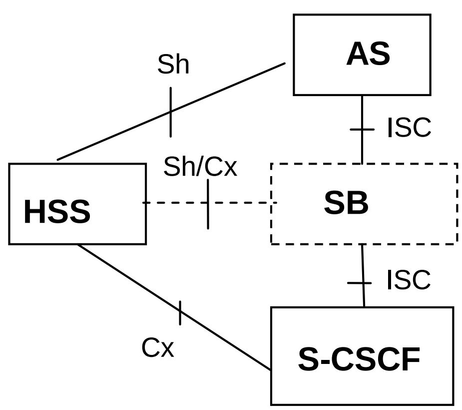


Fig 2. Reference Architecture **[1]**

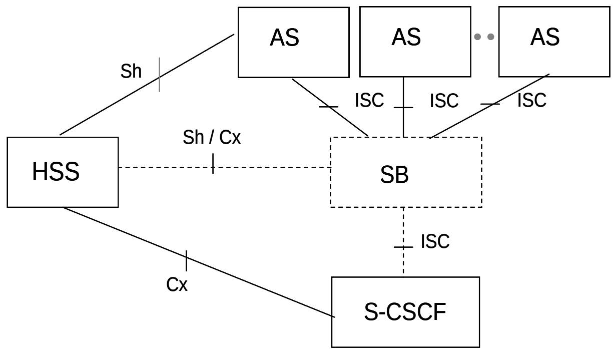


Fig 3. 3GPP Release 8 Service Broker Architecture, Centralized Service Broker for a single S-CSCF [1]

Similar to SCIM, the service broker is illustrated in the dotted box for both diagrams illustrated above. This indicates that it can be either an external independent entity, or embedded in AS or the S-CSCF. The interface between HSS and SB, also indicated by dotted line, is optional for SB to download the interaction logic from HSS.

ISC interface is used to between Application Servers, Service Broker and S-CSCF. The ISC interface is enhanced for carrying IMS service invocation history that is listed based on the Service ID and Service Effect (what service has been invoked, and the result of invocation, whether it is denied or screened). The Sh and Cx interface linked to the HSS is for downloading the interaction logic, expressed as a set of rules based on service invocation history.

The reference architecture shown can be implemented to various types of service brokers like centralized, distributed or hybrid service broker. For release 8 architecture as illustrated in Figure 3, it implements centralized service broker architecture, which will be further discussed in the next section.

3.2 Alternatives Architecture of Service Broker

Progress on the Service Broker has been slow as documented in [1] as part of the 3GPP R8. At the time of writing this paper, the content of [1] is still considered very high level and nothing promises that it will lead to anything solid in R8 timeframe, especially the specification phase. Different Service Broker architecture alternatives, i.e. centralised, distributed and hybrid service brokering functions architectures, are considered in [9]. Figures 4, 5, and 6 are taken directly from [9] showing the respective proposed Service Broker (SB) architecture alternatives.

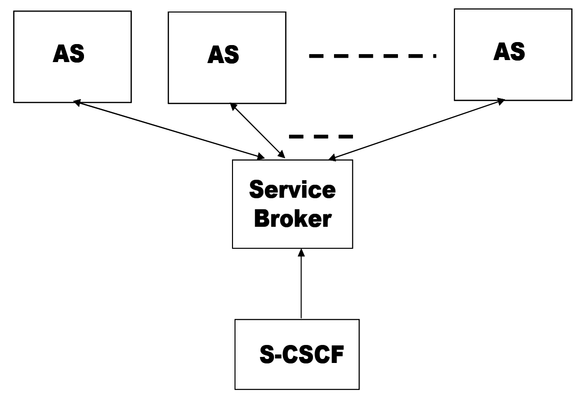


Fig 4. Centralised Service Broker [9]

In Fig. 4 , the ASs offering integrated service are unaware of the existence of service broker in this architecture. S-CSCF treats service broker as an ASs that supports ISC interface. The service broker functions can be either located outside S-CSCF or embedded in it.

As indicated in [10], the service broker implementing this architecture can be defined in either service layer or session control layer. The former one lacks interoperability support between ASs and requires supplementary security and trust control functionalities; whereas for the latter case, problems might arise through the perspective of network scalability, new services flexibility and congestion.

For architecture as shown in Fig. 5, each AS is equipped by one Service Broker, that may be either located independently or embedded in AS. The service broker is treated by S-CSCF as AS, where S-CSCF relays the messages among service brokers until all application servers finish their functions.

In order for these service brokers to cooperate coherently, standards protocols and procedures are needed.

For distributed service brokering functions, the service broker can either be embedded in application servers in the service layer, or embedded in session control layer in the session control layer. For the first architecture stated, it is not favorable as the autonomy and independence of services are not ensured. It requires update in feature integration management functionality over each application server once a new service is introduced. For the latter architecture, the service capability interaction management decisions are made by S-CSCFs that is has service broker functionality [10].

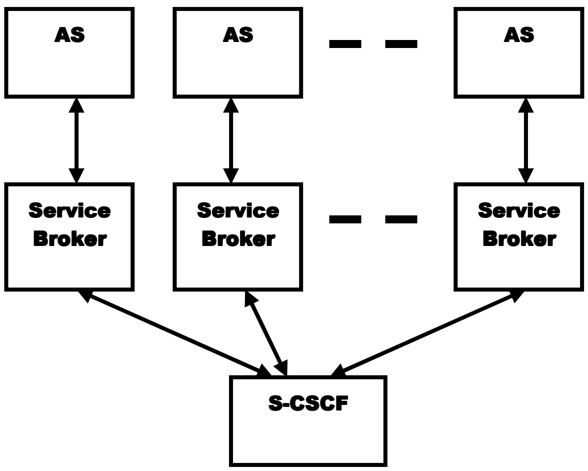


Fig 5. Distributed Service Broker [9]

Besides the architecture alternatives, the Service Broker ISC (IMS Service Control interface) improvements are also suggested in [1]. The improvements proposed are listed below:

- improvement when re-targeting Request URIs
- improvement when handling AS generated error responses
- enhanced service triggering conditions
- indication of specific service executed by the AS with multiple services supplied

This effort indirectly indicate that the 3GPP does not have concrete definition on Service Broker at this stage and would need to devote substantial effort in order to further realise the concept.

As the definition of SCIM is not well defined and we have described the evolution of SCIM to a possible Service Broker functionality from the 3GPP standards perspective, from this point of this paper we will primarily discuss in more generic term which is Service Broker functionality rather than making any imprecise claim of a SCIM. Having said that, it should be clarified that the initial phase of Service Broker study item is in the process of being concluded in 3GPP, that 3GPP has no immediate intention to standardise the Service Broker functionality in its IMS architecture at this stage. A second phase of the study item has been recommended to focus more on service interactions between multiple users, multiple service providers, and multiple sessions.

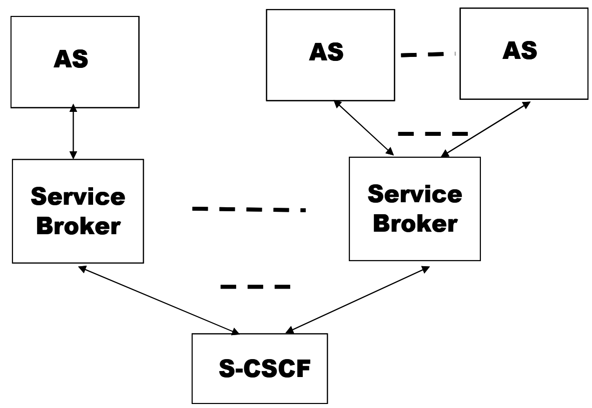


Fig 6. Hybrid Service Broker [9]

4 Service Broker Implementation Issues in IMS

There are numerous solutions [11] [12] [13] [14] [15] [16] [17] [18] [19] [20] for service capability interaction, such as avoidance methods and detection and resolution methods. Nevertheless, each different solutions proposed brought diverse drawbacks, arising the need of a functional entity that is in charge of detecting and resolving service capability interactions; hence the introduction of Service Broker.

As proposed by the 3GPP [1], Service Broker represents an intermediate entity of building blocks between the CSCF control layer and the service layer that manages the interactions between service capabilities. In this section, we discuss implementation issues of Service Broker function from an architectural aspect.

In order to manage service interaction, Service Broker needs to individually invoke each service on an application server - this implies the Service Broker needs to have sufficient knowledge of:

i) All the services to be invoked;

ii) Interface between the Service Broker and application servers to provide the identification of individual services;

iii) Interaction relationships between services to be invoked;

iv) Service provider and network operator's policy or business rules;

v) Subscriber preferences.

This is a complicated task to achieve - as it requires specifications that address individual services, while currently the 3GPP standards only manage to address interactions between application servers and the network. It also requires a standardised way to classify each service into different Service Equivalent Classes [1] to facilitate a standardised way to manage the order of service invocation dynamically to avoid run-time conflicts.

Today, most of the proprietary implementations are using the term SCIM to represent the functionality they have developed to achieve Service Broker role. We have primarily identified three different types of proprietary approaches (either vendor proprietary or academic) similar to the Service Broker function. The three types of the approaches are:

i) Internal approach as part of an application (denoted as SB-AS), this approach is comparable to the 'distributed' case in [9] which one SB serves an application (AS);

ii) Internal approach as part of S-CSCF (denoted as SB-SC);

iii) External approach as stand-alone server (denoted as SB-EX), comparable to the 'centralised' and 'hybrid' cases in [9] which one SB serves multiple applications (AS).

Figures in 7, 8 and 9 show the different approaches respectively, and Table 1 provides a broad comparison between these different Service Broker approaches.

4.1 Internal Approach: as part of AS

This approach [21] resides the Service Broker (SB) function within an Application Server, which enables the application server to invoke specific service logics based on the nature of the request.

4.2 Internal Approach: as part of S-CSCF

In this approach [22], network equipment manufacturers (NEMS) provide Service Broker (SB) functionality as part of S-CSCF.

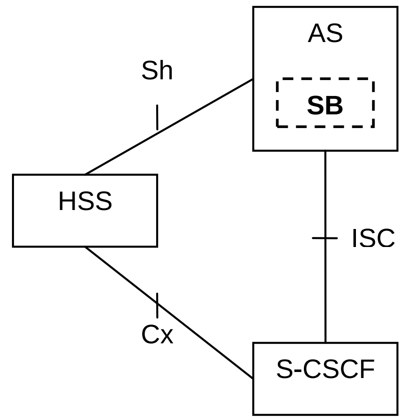


Fig 7. External approach as part of AS

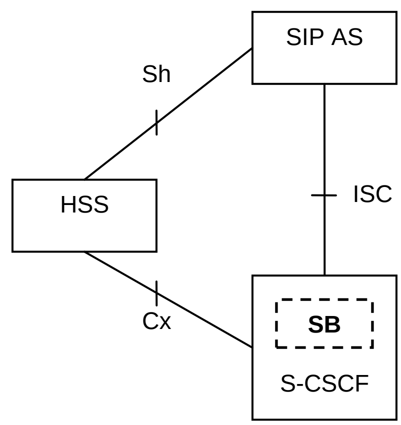


Fig 8. Internal approach as part of S-CSCF

4.3 External Approach: as stand-alone server

In these approaches [23] [24] [25] [26] [27], the Service Broker resides in between AS, S-CSCF and HSS. Most of the IMS deployments today are mostly hybrid environments where some service components are IMS compliant and the rest are distributed among legacy systems.

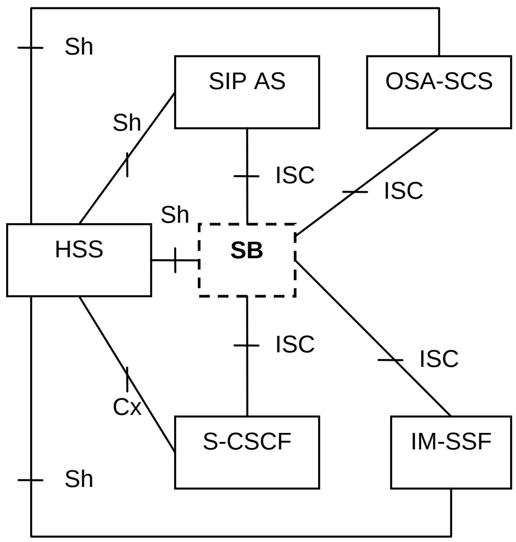


Fig 9. External approach as stand-alone server

4.4 Comparisons between Different Approaches

Table 1 shows the comparisons of different approaches to Service Broker functionality. By considering the factors of rapid service creation and deployment in IMS, it is essential to have the Service Broker function that supports vendor and technology independence, and requires minimum effort and time for service interaction logics change in the IMS implementation.

Consequently, from the rationale of having 'clean' architecture component functionality, a Service Broker should preferably be serving and focussing on the services it is assigned to coordinate.

	Internal as part of AS (SB-AS)	**Internal as part of S-CSCF (SB-SC)**	**External as stand-alone server (SB-EX)**
Architecture aspect	Coordination logics for service interaction and protocols and business rules will need to be built tightly in AS	Heavily overload S-CSCF server	Provides clean functions distinction between S-CSCF, SB and AS. Scalability in the network will need to be efficiently handled for new services.
Deployment speed to service interaction logics change	Requires change to AS logics	Requires change in S-CSCF – requires NEM involvement, maybe slowest in coping with service interaction logics change	May requires only update to service interaction rules
Effort required to service interaction logics change	Service redeployment is required in AS.	Highest among all approaches - required a particular NEM to change	Maybe as low as changing service execution rules
Technology dependency	Tightly coupled with the technology of application implemented in the AS	Minimum dependency	Minimum dependency
Vendor dependency	Minimum dependency	Service provider will need to tie up with a particular NEM and consequently restricting vendor-independence	High dependency without standardised specification. Interface and protocols between entities and Service Broker will be NEM implementation specific.

Table 1. Service Broker Functionality Approaches Comparisons

If not all services are resided at the same application, the Service Broker may use an extension of its protocol to call the remote servers - this would be compatible with the SB-EX approach, i.e., "hybrid" case in [9].

5 Features of Service Broker: Considerations

In this section, we discuss the possible features of Service Broker from the perspectives of complementing the current flaws in IMS architecture.

5.1 Dynamic Service Interaction

When a user registers to the IMS core network, S-CSCF will assess the user profile from the initial Filter Criteria (iFC) that is stored in an HSS. The iFC determines the service invocation sequence according to their priority set [28]. This mechanism provides simple and efficient service interaction mechanism in a static predefined manner.

However, advanced mechanism is required for more dynamic and intelligent service invocation by taking consideration of contextual criteria such as presence information and location. In view of extending service interaction mechanism to the S-CSCF for advanced interaction, there are problems to be considered:

- It may cause over-loading to the S-CSCF of SIP interactions with application servers, in addition to other tasks that already dedicated to the S-CSCF, which the S-CSCF is overseeing and performing most of the functions in the IMS core network such as user registration & authentication, handles charging process, end-to-end SIP routing and etc.

- It may lead to bad end-to-end latency and worsen the global session setup delay (time to complete a transaction at the client and application server side) for service delivery due to multiple SIP based interactions between the S-CSCF and application servers.

Based on the problems mentioned, it is preferably S-CSCF should not perform advanced service capabilities interactions mechanism. If more advanced mechanisms are required, they should be implemented probably in the Service Broker for more intelligent “interaction management” [6]. In this case, the Service Broker will perform at between the application and service layers, whereas S-CSCF to stay focus at the session core control layer. The role of Service Broker is therefore to select and integrate service capabilities that need to be invoked when an initial request reaches the Service Broker. In this case, the Service Broker may be resided at the application or outside the application, but not as part of S-CSCF.

To achieve this, the Service Broker will require (or gain access to) a predefined service capability interaction model that gives all potential interaction decisions, for instance, in what order and priority that may be invoked at the Service Broker level. The service capability interaction model needs to explicitly describe individual service and the valid interactions between service capabilities according to user profiles. Such a Service Broker function will provide enriched user experience and customisable service experience between IMS users.

5.2 Communication between Basic SIP-based AS and 3GPP SIP AS

One of the predicaments is that IMS employs 3GPP SIP as its core signalling protocol, which some of the 3GPP SIP extensions are not applicable to Internet Engineering Task Force (IETF) defined SIP [29]. This then causes the difficulty for interoperability between IMS SIP AS and basic SIP AS.

The basic SIP protocol as proposed by IETF, stated in [30] is an application layer signaling protocol for the establishment, modification and termination of Internet multimedia session. This protocol is widely utilized for Internet telephony, multimedia distribution and conferences.

However, there are dissimilarities between IETF defined SIP and the 3GPP SIP employed by IMS as 3GPP SIP introduces extensions in three areas: message body types, headers, and messages types. 3GPP SIP extensions can be subdivided into five classes: general, session operation, QoS, Authentication, Authorization, Accounting, and Charging (AAAC) and security, as stated in [29].

A general comparison done between 3GPP SIP and basic SIP is shown in [29]. To list a few, UA-proxy (User Agent-proxy) authentication is optional for basic SIP, but is forbidden in 3GPP SIP. Additionally through the aspect of extension support, P-headers and privacy mechanism are optional for basic SIP, whereas both of it are mandatory for 3GPP SIP. Moreover, some features such as the integration of resource management, event notification, Security

Mechanism Agreement remain optional for basic SIP, but are mandatory for 3GPP SIP.

The interworking between IMS SIP AS and basic SIP AS is essential especially for user to access 3GPP system IMS-based services; since this involved heavy usage of SIP for session negotiation and control. The signalling interworking necessitates traditional SIP elements to understand the 3GPP extended SIP protocol. Moreover, SIP can be extended freely by domain operators based on their need to support certain specific services, making the interworking between different versions of SIP challenging.

5.3 Communication Interface for SIP and Non-SIP AS

One of the main objectives of introducing Service Broker is to manage and coordinate service capabilities interactions between different types of application servers defined in the IMS architecture. The different types of application servers consist of SIP AS, OSA-SCS (for OSA gateway), and IM-SSF (for CAMEL/IN gateway).

In [1], the ISC (IMS Service Control interface) is used as the interface between S-CSCF and application servers, as shown in Figure 10. However, the ISC is currently very high-level and not specific in how it handles:

i) Different invocation mechanisms for different types of application servers. For instance, SIP-AS is invoked based on SIP protocol standardized mechanism. However, OSA-SCS might be open service interface such as Web Service (WS) interface running on protocols for instance eXtensible Markup Language (XML), Web Service Description Language (WSDL), and Simple Object Access Protocol (SOAP). Below provides a brief overview of the protocols utilized [31]:

- XML:
  Fundamental protocol for WS for defining and describing information
- WSDL:
  Used to describe WS interfaces; defining operations and binds them to protocols
- SOAP:
  Facilitates the message transfer. SOAP messages are generally conveyed using HTTP.

ii) Communication mechanism of different application server protocols such as SIP and non-SIP protocols. For instance, an application that does not comply with IMS SIP extensions may not be able to extract or interpret the 3GPP P-Asserted-Identity header, and therefore the service will not be exploited.

To solve the incompatibility of different SIP and non-SIP application server protocols communication, adaptations may be performed in the Service Broker to provide interface to different types of applications in order to manage service interactions between them. In this case, the Service Broker may adapt SIP messages between SIP and the non-SIP AS. Further, the Service Broker may support business rules for service interactions of an application that is owned by a 3rd party service provider.

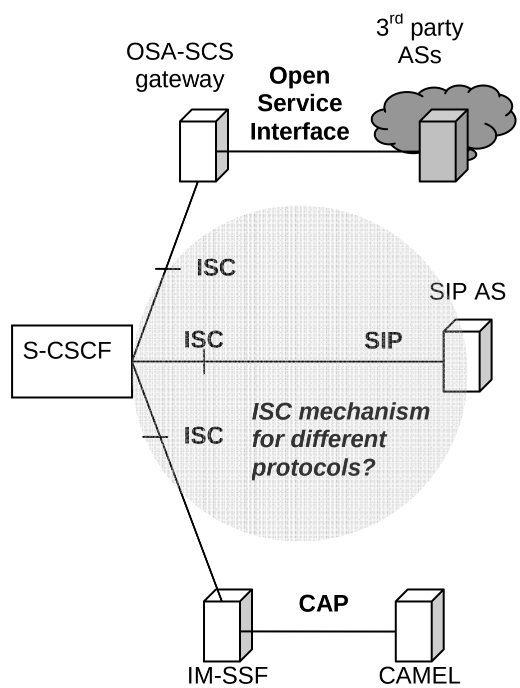


Fig 10. ISC for different application servers – under defined in [1]

5.4 Data Interoperability between ASs

For various types of AS, there exists different system behaviours, for instance, different types of system input and output parameters such as string and character. In order to cooperate between different ASs for integration of service capabilities, such data must be adapted for interoperability.

The responsibility to interoperate data of different parameters might be held by Service Broker as well. However, such mechanism is yet to be specified hence remains as one of the challenges.

6 Conclusions

To realise the full potential of IMS, it demands an open service creation environment where service capabilities can be reused and invoked to form any number of integrated services, via service interaction mechanism – which will enable operators to rapidly deploy a service and minimise their integration efforts. Service Broker is being studied by the 3GPP in [1] to provide service interaction management but its precise working procedure is still unresolved today.

In order to invoke and knit together reusable service capabilities from different application servers to create integrated services, understanding the impact and requirements of Service Broker function in an IMS architecture is also essential. Further investigation regarding the detailed structure and mechanism of Service Broker function is required as it plays a critical role in managing service interactions: without it, the efficient dynamic interactions of different service capabilities will be difficult to achieve. Moreover, service broker should address some of the difficulties faced by IMS architecture, such as the communication between basic SIP AS and 3GPP SIP AS, and between non-SIP AS and SIP AS.

As currently there are mostly proprietary Service Broker approaches implemented due to immature definition by the standards, thorough test on the functionality provided by vendors is necessary to ensure interoperability. The functionality implemented in Service Broker should be able to mediate and complement IMS core network architecture and service interactions between different application servers.

References:

[1] 3GPP, “Study on Architecture Impacts of Service Brokering”, TR 23.810 v1.0.0.

[2] Hui-Min, H., Ying-Hong, W. Bandwidth Management Method for Heterogeneous Wireless Network. *WSEAS TRANSACTIONS on COMMUNICATIONS*, Issue 4, Volume 7, April 2008.

[3] 3GPP, “Service requirements for the Internet Protocol (IP) multimedia core network subsystem”, TS 22.228, Release 8.

[4] 3GPP, “Service Requirement for the Open Services Access (OSA)”, TS 22.127, Release 8.

[5] ETSI DTR-01024, “TISPAN: NGN; NGN generic capabilities and their use to develop services”, 2005-05.

[6] 3GPP, “Network Architecture”, TS 23.002, Release 8, V8.2.0.

[7] 3GPP, “IP Multimedia Session Handling; IM Call Model”, TS 23.218 v8.0.0.

[8] 3GPP, “IP Multimedia Session Handling; IM Call Model”, TS 23.218, Release 8, V8.0.0, section 5.1.

[9] 3GPP, “Study on Architecture Impacts of Service Brokering”, draft TR 23.810, Release 8, V1.0.0, section 5.

[10] Gouya, A., Crespi, N. Detection and Resolution of Feature Interactions in IP Multimedia Subsystem, *International Journal of Network Management*, 2008 (To be published).

[11] Kolberg, M., Magill, E.H. Handling Incompatibilities between Services deployed on IP based Networks, *IEEE Intelligent Networks*, 2001, 360-370.

[12] Griffeth, N.D., Velthuijsen, H. The Negotiating Agents Approach to Runtime Feature Interaction Resolution. *Feature Interaction in Telecommunication Systems, IOS press*, 1994, 217-235.

[13] Harada, D. Fujiwara, H., Ohta, T. Avoidance of Feature Interactions at Run-Time, *IEEE International Conference on Software Engineering Advance*s, 2006, 6-12.

[14] Ohta, T., Harada, Y. Classification, Detection and Resolution of Feature Interactions in Telecommunication Services. *Feature Interaction in Telecommunication Systems, IOS press*, 1994, 60-72.

[15] Chan, K.Y., Bochmann, G.V. Methods for Designing SIP Services in SDL with Fewer Feature Interactions. *Feature Interaction in*

Telecommunications and Software Systems VII, IOS press, 2003, 59-77.
[16] Chentouf, Z., Cherkaoui, S., Khoumsi, A. Implementing online Feature Interaction detection in SIP environment: early results. *IEEE International Conference on Telecommunications*, 2003, 515-521.
[17] Khoumsi, A. Detection and Resolution of Interactions between Services of Telephone Networks. *Feature Interaction in Telecommunications Networks IV, IOS Press*, 1997, 78-92.
[18] Jouve, H., Gall, PL., Coudert, S. An Automatic Off-line Feature Interaction Detection Method by Static Analysis of Specifications. *Feature Interactions in Telecommunications and Software Systems VII. IOS Press*, 2005, 131-146.
[19] Kolberg, M., Magill, E.H. A Pragmatic Approach to Service Interaction Filtering between Call Control Services. *The International Journal of Computer and Telecommunications Networking*, Volume 38, Number 5, 2002, 591-602.
[20] Kolberg, M., Magill, E.H. Managing Feature Interactions between Distributed SIP Call Control Services. *ELSEVIER Computer Networks*, Volume 51, Issue 2, 2007, 536-557.
[21] Chris Buliton, "An introduction to the Service Capability Interaction Manager (SCIM)", Avaya white paper, September 2007.*https://udn.devconnectprogram.com/products/whitepapers/final-wp_scim-092007-lb3700.pdf*
[22] Gouya, A. Crespi, N. Bertin, E., SCIM (Service Capability Interaction Manager) Implementation Issues in IMS Service Architecture, *IEEE International Conference on Communications*, 2006.
[23] Kocan K, Roome W and Anupam V: 'Service Capability Interaction Management in IMS Using the Lucent Service BrokerTM Product', *Bell Labs Technical Journal*, 10, No 4, pp 217—232, 2006.
[24] Leapstone, CCE serviceBROKER™, http://www.leapstone.com/serviceBroker.php.
[25] AEPONA, "Aepona Service Broker", *http://www.aepona.com/products/service_broker/.*
[26] IntelliNet Technologies, Accelero SCIM™, *http ://www.intellinet-tech.com/products/scim/index.php.*
[27] Stratus Technologies, Converged Service Broker (CSB) Solution, *http://www.stratus.com/telecom/service-access.htm*
[28] 3GPP, "IP Multimedia Session Handling; IM Call Model", TS 23.218, Release 8, V8.0.0, section 6.9.
[29] F. G. Marquez, M. G. Rodriguez, T. R. Valladares, T. De Miguel, and L. A. Galindo, Interworking of IP Multimedia Core Networks Between 3GPP and WLAN, *IEEE Wireless Communications*, vol. 12, 2005, pp.58-65.
[30] Libnik, R., Svigeli, A. Performance evaluation of SIP based handover in heterogeneous access networks. *WSEAS Transactions on Communications*, Issue 5, Volume 7, May 2008.
[31] Mahmood, Z. Software Products and Technologies for the Development and Implementation of SOA. *WSEAS Transaction on Computer Research*, Issue 1, Volume 3, January 2008.