Middleware in Telecommunications

Mehdi Khorasani

Lucent Technologies

Abstract: *Middleware and Application Programmer Interfaces (APIs) has a successful history in the computing industry. Recently, this approach has been taken by the telecommunications industry to address some major architectural issues. This paper will study various aspects of Middleware modelling in telecommunications with the focus on the performance.*

1. Introduction

A new wave is under way; a true IT and telephony convergence is happening in 3rd Generation Mobile Networks. Operators are facing a new era, where individually they will no longer be in the position of controlling and owning the 3G Networks. The complexity of the network infrastructure is increasing, as it will include fixed, 2nd generation mobile and IP network elements as well as satellite components.

Middleware & API modelling is being adopted by the telecommunications industry to address these issues. This will allow them to define open network APIs and to modularise the complex architecture of the new telecommunication systems.

2. Middleware in Telecommunications

Commercial Drivers: Through commercial and regulatory pressure, the model for delivery of communication services is moving towards that of a Service Provider Architecture. Regulatory bodies are asking network operators to open up their networks to third party service providers. Consequently, the number of service providers is increasing rapidly.

Technical Drivers: *Middleware & APIs* allow segregation of the service domain from the underlying network domain and provides a flexible solution to network and service design. Some major architectural characterisations of the current network evolution are:

- **Modularisation:** The introduction of the Intelligent Network (IN) architecture for the fixed networks was the first step towards a layered approach. Before that, a fixed network was a bunch, although hierarchical, of switches, where advanced voice services had to be built directly onto each single switch. The IN architecture pulled the intelligence (i.e. telephony services) out of the core network and modularised a fixed network by layering that into the Service Control Point (SCP) and Service Switching Point (SSP). Logically, the IN was the first telecom middleware approach.
- **New Service Delivery Model:** A change in the network centric service delivery is that the communication services are moving ever outward from the core network. Some examples are SCP based services (IN services), IN services at the periphery (CTI and CSTA based services), IP based Call Centres, etc. A change in the edge of service delivery is that enterprise and personal functionality requirements are pressing ever inward the core network. Some examples are switch based CTI extensions, TAPI, PBX based services, etc. The new service delivery model needs to be implemented on open network APIs.
- **Functionality move:** In a traditional fixed network, the conventional thinking has been that the switching functionality has always been an inseparable part of the core network. Today, these once solid definitions are not valid any more. The introduction of soft-switches altered this way of thinking. A soft-switch offers the switching functionality and can reside in any layer, preferably in a middleware layer. This technology is used for

transferring voice over packet switching. To generalise the idea, functions are unravelling themselves from layers and are moving across the layers.

- **Evolution to 3rd Generation Mobile Networks (3GMN):** This topic is explained in more detail in Section 3.6.

Model and Characteristics: *Middleware* in networking refers to a layer of software sitting between the network and applications, providing services such as identification, authentication, authorisation, soft-switch, certification, directories, security, etc.

The layer itself, as a part of a complex telecommunication network, is vastly distributed, time critical (real time) and needs to be 99.999% (Five Nines principle) reliable and available.

This modelling is based on the open network APIs. The Parlay Group, founded in 1998, is an open, multi-vendor forum organized to create open, technology independent APIs, which enable IT companies, ASPs, ISV's, Internet Companies, E-Business Companies, software creators, service bureaus, and large and small enterprises as well as network providers, network equipment vendors and application suppliers to develop applications across multiple networks. Furthermore, the Group promotes the use of Parlay APIs and ultimate standardization. Current members of the Group are AT&T, BT, Cegetel, Cisco Systems, IBM, Ericsson, Lucent Technologies, Microsoft, Nortel Networks, Siemens and Ulticom.

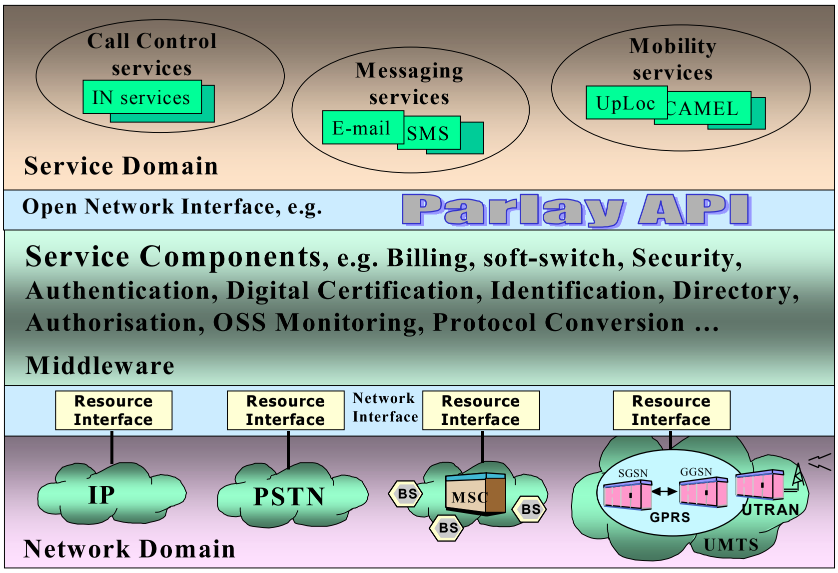

Figure 1 - Middleware and APIs in Communication networks

Performance Study: When operators open up their networks to third party application and service providers, the security and integrity of their networks are their major concerns. Application and service providers are also concerned about the availability, performance and resilience of their services when accessing network capabilities using open network interfaces. Underlying network elements can belong to many different operators. Security, integrity and performance are major factors in designing a middleware product.

The study on performance was focused on two issues: 1- the impact of adding security functions to a middleware prototype. 2- the impact of design and implementation related issues. Due to the size limitation, this paper only concentrates on the second study.

Test Goal: To study the impact of some major design & implementation issues on the performance.

Test Bed: In the absence of a pure middleware product, the study has been focused on a middleware-like product, an Intelligent Network platform (INP), implemented on Alpha servers 4100. The INP is not a pure middleware product as: 1- it, in addition to offering some middleware functionality, also provides SCP functionality. 2- it currently only supports protocol interfaces (i.e. INAP CS1 & CS2) not open network interfaces (e.g. Parlay APIs). A much better candidate can be an implementation of the Parlay Specification, a Parlay Gateway. However, the INP is a good case study as it includes many common features of a middleware approach such as having: a big database (ObjectStore), transaction capabilities, overloads control mechanism and a distributed object oriented layered design. The system is capable of handling 3000 IN service calls (e.g. basic or advanced number translation) per second. The following tools have been used in the test environment: a 'Martinet' simulator, configured to generate ~600 calls per second, Unix commands (vmstat, ps, etc to measure the time and resource occupancy), 'Collect' (to collect and analyse the system and application data), Sniff+ (to locate areas for code optimisation) and a SS7 product to link the simulator to the system under test.

Test Criteria: An IN call normally should be set-up within 5-6 seconds. This time is divided between the SSP IN resources, middleware and the application/service layers. The response time requirement for the INP is 250 msec (only for transactions & simple number translation). This is the response time that 95% of calls within a 15-minute period must meet. Measurements are from receipt of the last bit of the InitialDetectionPoint INAP operation to the transmission of the last bit of the Connect (or equivalent INAP operation).

Test analysis: A detailed analysis of the test is again the subject of another paper. This paper will only highlight the identified factors, which have a significant impact of the performance, i.e.:

- **Database cache:** Each application that is a client of an ObjectStore database server has a cache, which for performance reasons is large enough to hold all call records the client needs to process. The client cache, in the Digital Tru64 Unix implementation, is in the memory mapped file format, which maps pages of cache into a file on the file system. To prevent any loos of data in the case of a crash, the (Unix) update command executes a sync system call every 30 seconds to write all unwritten system buffers, including cache files, to disk. Because of the huge size of the cache, this causes 'spikes' in CPU occupancy, hence impacting the performance of the call processing.
- **Queues:** Inter process communications have been implemented using Unix Queues. As many calls are handled simultaneously, processing requests are queued at processes' input queues. A bottleneck in the design of 'queues', e.g. allowing multiple processes writing into a single queue, will drop the performance of the system severely and even worst it may cause overload.
- **Complex configuration:** In the first version of the system, Unix environment variables were widely used in the configuration of each component. In the second version, configuration files replaced most of these variables. In the later version, the installation scripts produced most of the configuration files (a semi automated approach). On a few occasions, a mis-configuration has caused serious performance drops and even overload. The study shows that such a complex system needs an automated intelligent configuration mechanism to be able to predict inconsistencies intelligently.
- **Hardware and Operating System configuration:** The hardware configuration includes, amongst others, multi-processors Alpha servers 4100, Digital Memory Channel, ObjectBroker and ObjectStore. A huge effort has been made to tune the hardware configuration and the operating system (kernel parameters, etc) to achieve the maximum

performance, but still the system designers are sceptical on the performance of the Memory Channel and the processors' inter-working schedule.

- **Overload:** The behaviour of systems at their thresholds has always been a very interesting subject for research. The study shows that the performance drops exponentially as the INP approaches its maximum boundaries. An overload control mechanism has been designed to protect the system from overload. The system was recently involved in an overload incident. One recommendation is to add a reasonable level of intelligence to future overload control systems to enable them to detect automatically and dynamically how closely they are modelling the external world.

Future: This segment highlights some 3G Mobile Network characteristics, which identifies the potential for middleware modelling in the new architecture:

Increased complexity in the network layer: 3rd generation systems are more than just cellular. The aim is to provide a wide range of services including telephony, text messaging, voice messaging, paging and broadband ISDN capability. The migration of current networks to 3rd generation will involve the integration of several systems such as fixed, cellular, wireless local loop, cordless and satellite systems.

IT and Telephony Convergence: A true IT and telephony convergence is happening in the 3G architecture. The IT world is rich in applications and telephony systems are mature in network centric services. This will open a new door to future communication services.

No 'one' owner: The current market, which is mainly vertically separated between operators, may change. Operators may no longer be classified as either fixed or mobile, but may offer a variety of different access technologies. The market may not be defined in terms of whole networks, but in terms of access operators, switching operators, transit operators, service operators and data management operators.

3. Summary

- *Middleware* is a layer of software residing between hardware/network and applications/services, providing a set of common and re-usable functions.
- These functions are accessible through APIs.
- This layered approach allows us to:
  - reduce the hardware and network dependencies.
  - abstract the hardware and network complexity from application developers and service providers.
  - segregate the application/service domain from the underlying hardware/network domain.
  - re-use basic common functions.
  - standardise open interfaces, hence to bring more players to the scene.
  - satisfy regulatory requirements, consequently to increase competitiveness.
  - rapidly increase new applications/services.
  - reduce the 'time to market' and the 'cost'.
  - link the Intelligence in the core networks and the Intelligence at the periphery.
  - close in and finally converge the IT and telephony worlds in the 3rd generation networks.