

VU Research Portal

Improving Solution Architecting Practices

Poort, E.R.

2012

document version

Publisher's PDF, also known as Version of record

[Link to publication in VU Research Portal](#)

citation for published version (APA)

Poort, E. R. (2012). *Improving Solution Architecting Practices*. [PhD-Thesis - Research and graduation internal, Vrije Universiteit Amsterdam].

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal

Take down policy

If you believe that this document breaches copyright please contact us providing details, and we will remove access to the work immediately and investigate your claim.

E-mail address:

vuresearchportal.ub@vu.nl

5

Case Study: Successful Architecture for Short Message Service Center

In Part I of this thesis, we examined how non-functional requirements be handled to improve the success of IT solutions and the projects delivering them (research question RQ-1). In Part II, we will now turn our attention to research question RQ-2: establishing a solution architecting approach to improve an IT service provider's success. This chapter starts off Part II with a small case study. It presents and analyzes the key architectural decisions in the design of a successful Short Message Service Center as part of a GSM network.

5.1 Introduction and Requirements

In the early nineties, a Short Message Service Center¹ was developed according to the specifications for text messaging embedded in the GSM standard [ETSI, 1995]. This chapter looks back at the conceptual design phase of the realization project. The chapter is a practitioner's report, analyzing the key architectural decisions and distinguishing factors that contributed to the system's success.

The SMSC's key requirements are listed below, according to the categorization presented in §2.3.2: first the primary (functional) requirements, and then the supplementary requirements, divided in secondary functional requirements and quality attribute requirements.

¹This system was developed and commercially deployed by CMG, later LogicaCMG Telecoms and now Acision. In order to protect the commercial interests of the manufacturer, the descriptions have been left at a reasonably high level of abstraction, and data are mostly not quantified.

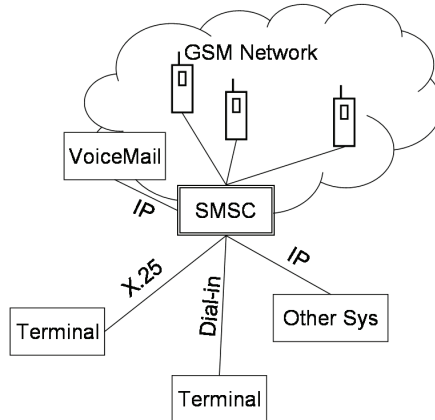


Figure 5.1: SMSC context.

Primary functional requirements

Fig. 5.1 shows the SMSC system in its primary context. The main purpose of the system is [PF1:] to pass messages between mobile telephones in a GSM network, and from and to other systems [PF2:] outside of the GSM network. Messages that cannot be immediately delivered are [PF3:] temporarily stored in the system.

Supplementary requirements

According to the classification in §2.3.2, supplementary requirements consist of three categories: Secondary Functional (SF) requirements, Quality Attribute (QA) requirements and Delivery requirements. For the purposes of this chapter, we only need to discuss the key SF and QA requirements.

The major secondary functional requirements were that [SF1:] a record of every message that has passed through the system is kept for billing purposes, and [SF2:] there is an interface to monitor and operate the system.

The major quality requirements set by the customers centered around [QA1:] *performance* of message throughput, [QA2:] *availability* of the messaging service and [QA3:] *reliability* of message storage. [QA4:] *Timeliness* in responses to external systems was critical. With a view to productizing of the solution, the manufacturer added requirements for [QA5:] *extensibility* and [QA6:] *scalability* of the solution.

5.2 Key Architectural Design Decisions

In order to fulfill the requirements set out above, the architects made some design choices that distinguished the system from other similar systems in three major aspects: *platform choice*, *storage strategy* and *interprocess communication*.

5.2.1 Platform choice

The main choice to be made with respect to the platform for the SMSC was between a traditional “telecom switch” platform and an IT platform. Even though the telecom switch platforms were better rated in terms of performance [QA1], availability [QA2] and reliability [QA3], IT platforms were deemed superior in terms of extensibility at a reasonable cost [QA5].

At the time of the design of the SMSC, the most popular platforms for these kinds of medium-high performance requirements were Unix environments. The development team, however, also had ample experience with OpenVMS platforms. It was felt that the OpenVMS platform would better be able to fulfill the timing requirements [QA4].

5.2.2 Storage strategy

The performance of the system [QA1] was important and was perceived to become more important later on [QA6]. For this reason, it was decided to use a system where messages were stored in memory and on disk in parallel. The permanent message store mechanism is based on proprietary OpenVMS file I/O. If a more conventional storage strategy would have been used, such as an RDBMS, the added resource usage needed to perform the more complex file operations would have made it harder to fulfill the performance requirements [QA1]. Thus, the chosen storage strategy provided a better fit with the non-functional requirements.

5.2.3 Interprocess communication

A process architecture over multiple nodes was necessary to fulfill the performance and scalability requirements [QA1,QA6], resulting in a need for transparent communication between processes (IPC) running on different hardware units. It was felt that using the commercial-off-the-shelf IPC products available at the time would cause problems fulfilling the performance and flexibility [QA5] requirements. The team decided to develop a mean-and-lean transparent IPC itself. The resulting utility was christened VIQ (Virtual Interprocess Queue).

5.3 Conclusions and Discussion

In the years following delivery of the system to the first customers, demand for short message services grew spectacularly. In the race to keep up with this growing demand, performance and reliability turned out to be the main deciding factors. The product quickly became the world's leading SMS product in terms of number of subscribers being serviced.

The major lesson we learned from this success story is to *beware of fashion in solution architecture*. In the SMSC case, key architectural choices deviated from the prevailing “fashion” at that time, because analysis indicated that the more popular practices were not the best choices to fulfill the key requirements of performance, timeliness and reliability. The deviations turned out to be the key distinguishing factors in the architecture, that led to a success story.

Practicing architects in our experience are often under pressure from managers and customers to follow trends and fashions in system design. This phenomenon can partly be attributed to personal risk management behavior: it is hard to blame a manager for making a wrong decision if many others made the same wrong decision. We frequently encounter the term “best practice” to rationalize decisions that follow trends and fashions, often without a clear trade-off analysis as to why these practices are best for that particular situation. For this reason, we prefer the term “best fit practice” to the ubiquitous “best practice”.

Methods like the Cost Benefit Analysis Method [Kazman et al., 2002] can help architects to present the benefits of their choices in an objective way. This can be especially helpful when arguing choices that go against prevailing “fashion”. It should, however, be kept in mind that the previously mentioned “career risk management” argument for following trends and fashions is not necessarily invalid, and risk management related quality attributes can rightfully show up in architecture evaluations [Clements et al., 2002].

This case presents an example of the benefits of an environment where architects can argue their choices and priorities in an objective manner, and select practices that best fit those priorities, rather than follow fashion. In this case, such an environment led to a solution that was very successful in terms of the business goals identified in §1.3: particularly consistency in delivery and customer satisfaction. In subsequent chapters, we will explore a number of factors that can contribute to such an environment, leading up to the establishment of a successful solution architecting approach.