Abstract. Software architects struggle to choose an adequate architectural style for multi-tenant software systems. Bad choices result in poor performance, low scalability, limited flexibility, and obstruct software evolution. We present a comparison of 12 Multi-Tenant Architecture (MTA) patterns that supports architects in choosing the most suitable architectural pattern, using 17 assessment criteria. Both patterns and criteria were evaluated by domain experts. Five architecture assessment rules of thumb are presented in the paper, aimed at making fast and efficient design decisions. The comparison provides architects with an effective method for selecting the applicable multi-tenant architecture pattern, saving them effort, time, and mitigating the effects of making wrong decisions.

Keywords: Multi-tenancy, architecture patterns, quality attributes.

1 Introduction

As a consequence of the current shift of on-premises software to the cloud [4], software architects find themselves facing numerous new challenges related to the adequacy of architectures for cloud software. A commonly used technique in architecting for Software-as-a-Service (SaaS) is the use of the concept of multi-tenancy, which is defined for this research as “a property of a system where multiple varying customers and their end-users share the system’s services, applications, databases, or hardware resources, with the aim of lowering costs” [11].

Multi-tenancy can bring about many benefits. By serving the software service from a centrally hosted location, clients are relieved from the responsibility of purchasing and maintaining expensive in-house servers. The total cost of ownership decreases, giving the SaaS provider access to new potential customers that previously could not afford the expenses [2]. In addition, the utilization rate of hardware in a multi-tenant environment is higher than in a single-tenant environment [12]. Furthermore, when multiple customers share application and data instances, the total number of running instances will be lower than in a single-tenant environment, catering the same number of customers. A low number of instances is beneficial for maintenance [9] and is beneficial for application development [1].

However, multiple barriers withhold service providers from massively switching to multi-tenant environments. The challenges for multi-tenancy adoption include
performance, scalability, security, and the re-engineering of current software applications. Selecting the appropriate multi-tenant architecture is a complex problem due to the existence of numerous alternative architectural patterns. Benefits and barriers of multi-tenancy are identified and described in literature, but the aspect of choosing an appropriate multi-tenant architecture based on software vendors’ preferences has received little attention in literature. Finding the most suitable multi-tenant architecture is crucial; it expresses a fundamental structural organization schema for a provider’s software system. However, choosing the appropriate architecture is a wicked problem. Accounting for all the challenges and benefits complicates the decision process considerably.

This paper presents a comparison of different Multi-Tenant Architecture (MTA) patterns, based on the mixed method research approach used within this study (Section 2). The twelve different MTA patterns are shown in section 3, together the MTA comparison matrix in section 4. We conclude with a discussion on the comparison, together with threats to validity present and future work in section 5, focussing on the importance of evaluating more effective methods in architectural decision making.

2 Research Approach

The main research question of this research is formulated as follows:

RQ. How can a SaaS provider be supported in the decision process of choosing an applicable multi-tenant architecture?

Three sub questions are answered in order to develop a decision model that solves the main research question. The decision model consists of three fundamental elements, which need to be identified. The first element is a set of multi-tenant architectures to choose from. Hence, the first sub question is defined as follows:

SQ1. What distinctive layers in multi-tenant architectures can be defined? — Using a Structured Literature Research (SLR), the distinctive layers in multi-tenant architectures are identified in SQ1. For more details on the search query, criteria, strategy and construction of trail searches, please see [11]. Instead of searching directly for multi-tenant architectures and documenting them, a different approach is taken. First, different layers on which multi-tenancy can be applied are identified. Then, generic multi-tenant architectures are identified, based on these layers. The list of identified architectures is evaluated by domain experts to ensure the list is complete and concise. The expert evaluation is not only essential for checking the correctness of the list, but also to make sure the identified architectures reflect relevant and implementable architectures.

SQ2. What are the relevant decision criteria for choosing an appropriate multi-tenant architecture? — SQ2 aims at identifying the different decision criteria, or architecturally significant requirements, related to multi-tenant architectures.