Sommerville's chapter on cloud-based software was a brief overview of the applications of the cloud. Cloud computing is a topic with seemingly infinite paths of potential, so it is expected that Sommerville omitted some information that many would find useful, to keep the chapter within a reasonable size and learnability. Unfortunately, one of the services offered in the cloud was only covered briefly, for reasons of low maturity at the time of publishing. Therefore, this analysis will focus on function-as-a-service (FaaS) or serverless computing in the cloud. Particularly, this analysis will elaborate on the current state and challenges of the model, citing sources that have been published within the past ten months. This limitation was made to account for Sommerville's claim that serverless computing is not sufficiently mature. The expectation from the sources chosen is that they will include information that was not available when Sommerville's textbook was published, presenting new, significant insights into the realm of serverless.

According to Sommerville, serverless computing stands out due to low maintenance. It helps developers work better by avoiding the need to manage servers, allowing them to develop efficient systems with little configuration. Further, it helps product managers work better by operating on a pay-per-use model, cutting costs of hardware and inefficient use of time and resources. In short, Sommerville implies that serverless computing is a strong contender for future computing solutions as the model matures. Most other sources show that this is the case, but there are still challenges that need to be addressed before they can become widespread to the level of Platform as a Service (PaaS) or Infrastructure as a Service (IaaS).

A study on the state of serverless applications was conducted to determine a clear conclusion to the best use cases for serverless computing [1]. In addition to supporting Sommerville's claims of faster development and lower costs, the study determined that scalability was a primary role in the adoption of serverless systems and that some of the primary use cases were implementing APIs and stream/async processing. These applications benefit strongly from the serverless' pay-per-use model due to their sporadic and bursty traffic patterns. Furthermore, with the growing use of software in general, APIs will continue to grow, and using serverless functions to execute them, FaaS will grow in popularity alongside APIs.

One of the main concerns with serverless computing, which Sommerville made no mention of, is security. In an age of endless cyber-attacks and privacy concerns, security should be one of the primary factors when considering innovations in computing. The security flaws of serverless functions theoretically wipe out any potential cost mitigation that switching to a serverless system would achieve [2]. Since serverless is primarily used with APIs, and APIs are the communication lines between applications, there is distributed risk of breaching wherever serverless functions are used. Fortunately, the challenges are known, and there are proposed solutions on the table [2].

Serverless computing can turn a development project into a simple task, but it is important to consider the implications of using such simple execution models. As FaaS picks up in popularity among integration engineers, solutions to common problems such as security must be in place to avoid catastrophe. Sommerville had his doubts about serverless considering its "age", and his instinct to not elaborate on the subject proved to be helpful, since there is still much more work to do to the model before it is established as a stable, reliable, and safe execution paradigm.