In software-as-a-service applications provisioned through cloud computing, locally cached data are often modified with updates from new versions. In some cases, with each edit, one may want to preserve both the original and new versions. In this paper, we focus on cases in which only the latest version must be preserved. Furthermore, it is desirable for the data to not only be compressed but to also be easily modified during updates, since representing information and modifying the representation both incur cost. We examine whether it is possible to have both compression efficiency and ease of alteration, in order to promote codeword reuse. In other words, we study the feasibility of a malleable and efficient coding scheme. The tradeoff between compression efficiency and malleability cost-the difficulty of synchronizing compressed versions-is measured as the length of a reused prefix portion. The region of achievable rates and malleability is found. Drawing from prior work on common information problems, we show that efficient data compression may not be the best engineering design principle when storing software-as-a-service data. In the general case, goals of efficiency and malleability are fundamentally in conflict.