Local-First Software: An Energy-Efficient Alternative to Cloud-Centric SaaS Architectures

Lylian Siffre[1,2,*], Thomas Ledoux[1], Renaud Pawlak[3], and John Guery[2]

[1]IMT Atlantique, Inria, LS2N, Nantes, France
[2]Kapela, Paris, France
[3]LocalFlow, Paris, France
[*]*Corresponding Author: lylian.siffre@imt-atlantique.fr*

Abstract

The dominance of Software-as-a-Service (SaaS) has created systemic vulnerabilities in critical infrastructure while driving unsustainable growth in datacenter energy consumption. We analyze how Local-First Software principles, including offline functionality, peer-to-peer synchronization, and device-centric computation, could address these challenges. Our empirical results demonstrate that shifting functionality to end-user devices can reduce total system energy consumption by up to 38% by rendering server infrastructure obsolete. Building on these findings, we propose a three-phase methodology for energy-aware SaaS-to-Local-First migration, encompassing technical transformation patterns, impact quantification frameworks, and change management strategies validated through industry experiments and educational interventions. This work contributes to emerging discussions about sustainable computing infrastructure by providing both technical pathways and socio-technical adoption strategies for reducing cloud dependency.

1 Introduction

Software-as-a-Service (SaaS) has become the *de facto* standard for modern software development, offering undeniable advantages: browser-based accessibility, seamless collaboration, and effortless updates. Yet, this convenience comes at a cost: a deep and growing dependence on centralized cloud infrastructure.

Unlike traditional software, SaaS applications reside entirely on remote servers, leaving end-users powerless when connectivity fails. This model not only introduces single points of failure but also fuels relentless demand for datacenter expansion. Recent incidents, such as the October 2025 AWS outage[1], demonstrate how quickly disruptions can cascade, paralyzing everything from banking systems to mundane office tools.

The fragility of the cloud becomes particularly evident in critical contexts. Emergency services, financial systems, and healthcare platforms cannot afford downtime, yet SaaS architectures tie their functionality to distant servers and stable internet links. Even non-critical applications, like expense management software or note-taking apps, often fail entirely offline, despite requiring minimal computational resources.

This raises urgent questions about the frugality of our cloud reliance: why must a user's ability to edit a spreadsheet hinge on a datacenter thousands of miles away?

The Hidden Energy Burden. While end-user devices grow more efficient with laptops and smartphones now capable of on-device AI inference, the cloud's energy footprint balloons disproportionately. Datacenters, driven by SaaS demands and the AI boom, now consume hundreds of terawatt-scale energy[1], while local devices remain underutilized as mere "dumb terminals".

The mismatch is glaring: we push computations to energy-intensive servers even as consumer hardware surpasses the needs of everyday tasks. The result is an unsustainable trajectory, where efficiency gains in personal devices are eclipsed by the cloud's expanding appetite.

[1] `https://www.bbc.com/news/articles/cev1en9077ro`

Toward a Local-First Alternative. This disconnect points to an opportunity: software architectures that prioritize local computation, reduce cloud dependency, and leverage the untapped potential of end-user hardware. In the following sections, we explore how Local-First principles could address these challenges, balancing resilience, efficiency, and user autonomy.

2 Local-First Software

Local-First (LoFi) software adheres to seven core principles established by Kleppmann et al. [2]: instant interaction without loading delays, multi-device synchronization, offline functionality, seamless collaboration, data portability, end-to-end security by default, and full user ownership of both software and data. These principles require architectural shifts from cloud-centric models to, for example, peer-to-peer synchronization using conflict resolution mechanisms like CRDTs [3].

This paradigm shift has multiple concrete energy-saving impacts. From our position paper [5], let's explore three of the impacts we identified. First, by moving functionalities to end-user devices, entire server clusters become obsolete and can be permanently decommissioned. Second, the offline-first nature creates predictable synchronization patterns: since operations accumulate during offline periods, providers can precisely provision computing resources for batched synchronization tasks, eliminating the need for always-on overcapacity. Third, with primary data storage distributed across user devices, infrastructure requirements shrink dramatically, reducing both redundancy needs and backup overhead.

Experimental Evidence. Our empirical study [4] specifically validated the server reduction potential. By measuring total system energy (server plus smartphone) for equivalent tasks, we found that moving computation to end-user devices allowed significant server-side energy savings. The local-processing variant (demonstrating core LoFi characteristics though lacking collaboration features) showed up to 38% lower total energy consumption than the SaaS version on three of four tested devices, with server energy accounting for the majority of these savings. These results provide concrete evidence that shifting functionality to devices can indeed render server infrastructure obsolete while maintaining equivalent functionality.

3 Toward a Methodology for Energy-Aware Transition

Our findings suggest the need for systematic approaches to migrate from SaaS to LoFi architectures while considering energy impacts. We propose developing a three-phase methodology:

First, *quantification*, establishing metrics to assess both direct (energy consumption) and indirect (resilience, device utilization) impacts of architectural shifts. Second, *pattern extraction*, identifying common transformation patterns for typical SaaS components (e.g., replacing real-time APIs with synchronized local databases). Third, *change management*, addressing the socio-technical challenges of adoption.

To validate and refine this approach, we plan:

- Technical experiments with industry partners, measuring energy impacts of incremental LoFi adaptations in production systems
- Hackathons with both engineers and students to identify practical barriers to adoption
- Curriculum development for CS programs, emphasizing energy-conscious software design

This work ultimately aims to provide both technical pathways and organizational strategies for reducing cloud dependency, not just as an engineering challenge, but as a necessary evolution in how we conceive software systems.

Acknowledgments

We acknowledge DeepSeek-V3 for editorial support during manuscript preparation.

References

[1] IEA. Energy and ai, Apr. 2025.

[2] Kleppmann, M., Wiggins, A., van Hardenberg, P., and McGranaghan, M. Local-first software: you own your data, in spite of the cloud. In *Proceedings of the 2019 ACM SIGPLAN International Symposium on New Ideas, New Paradigms, and Reflections on Programming and Software* (Athens Greece, Oct. 2019), ACM, p. 154–178.

[3] Shapiro, M., Preguiça, N., Baquero, C., and Zawirski, M. Conflict-free replicated data types. In *Symposium on Self-Stabilizing Systems* (2011), Springer, pp. 386–400.

[4] Siffre, L., Ledoux, T., Pawlak, R., and Guery, J. Local Computing vs. Cloud Computing: An Empirical Study of Energy Consumption. In *IEEE/ACM 18th International Conference on Utility and Cloud Computing* (Nantes, France, Dec. 2025), ACM, pp. 1–10.

[5] Siffre, L., Ledoux, T., Pawlak, R., and Guery, J. Local-first software for green it. In *2025 11th International Conference on ICT for Sustainability (ICT4S)* (2025), pp. 58–68.