Cloud Design Tool: Miro

In the context of modern IT management, the selection of appropriate tools for cloud architecture design is paramount. Having evaluated various platforms, I believe Miro stands out as a superior tool, not merely for its diagramming capabilities but for its emphasis on collaborative flexibility. While platforms like Lucidchart are excellent for producing structured, formal diagrams, Miro's strength lies in its ability to facilitate the entire design lifecycle, from unstructured ideation to polished architectural schematics, within a single digital environment.

Miro's core strength is its expansive, limitless canvas, which functions as a digital whiteboard. This fosters an intuitive, free-form approach to problem-solving, allowing teams to use sticky notes and mind maps for initial brainstorming before transitioning to precise cloud diagrams with dedicated shape libraries for AWS, Azure, and GCP. This seamless integration of creative and technical work on one board promotes a shared understanding and rapid iteration, which is critical for agile methodologies (Yu *et al*, 2022). Academic research supports this, highlighting how digital tools can enhance team communication and design quality (London and Pablo, 2021). Miro's robust real-time collaboration features, such as multi-user editing, commenting, and voting, effectively replicate the dynamic of an in-person workshop, which is a powerful asset for distributed teams.

Despite its strengths, Miro has some notable drawbacks. The open-ended nature of the canvas can, without proper governance, lead to disorganisation and cognitive overload for users (Kleinsmann, Sarri and Melles, 2020). The lack of a strict hierarchical structure can hinder its effectiveness for long-term documentation, as boards may become cluttered and difficult to navigate over time. Moreover, while its integrations are strong, Miro lacks the advanced, data-driven automation features that dedicated diagramming tools often provide, such as the ability to automatically generate diagrams from code or to link directly to live cloud infrastructure data for synchronisation. For highly technical, large-scale projects, this can be a limiting factor.

Word count: 314

References

Kleinsmann, M., Sarri, T. and Melles, M. (2018) 'Learning histories as an ethnographic method for designing teamwork in healthcare', *CoDesign*, 16(2), pp. 152–170. Available at: https://doi.org/10.1080/15710882.2018.1538380.

London, K.A. and Pablo, Z. (2021) 'Simulation-Based Collaboration Training: Strengthening the Industry's Capacity to Collaborate', *Advances in civil and industrial engineering book series*, pp. 404–429. Available at: https://doi.org/10.4018/978-1-7998-6600-8.ch016.

Yu, R., Gu, N., Lee, G. and Khan, A. (2022) 'A Systematic Review of Architectural Design Collaboration in Immersive Virtual Environments', *Designs*, 6(5), p. 93. Available at: https://doi.org/10.3390/designs6050093.

Peer response

Thank you David, for your thought-provoking post. It is an excellent argument, and I appreciate your focus on the methodological discipline enabled by Archi rather than just its features. The recent research you have cited, particularly regarding the benefits of modular models on comprehension and decision quality by Baß *et al* (2024), provides a strong foundation for your position. The extension of ArchiMate to areas like MLOps governance is a particularly forward-thinking and compelling point. However, I would critically evaluate your claim of Archi being the "superior" tool. While you effectively argue for the benefits of the ArchiMate framework, a direct comparison with other commercial or open-source tools that also implement this framework, such as Sparx Enterprise Architect is not mentioned (Heras, 2024). A key weakness of your argument is that it focuses on the potential benefits of the methodology, rather than on the practical, real-world implementation challenges of Archi itself. For example, in a large, regulated enterprise, would the lack of commercial support for an open-source tool like Archi be a significant drawback? Moreover, while you mention that Archi mitigates excessive fragmentation, is this mitigation truly sufficient, or could the cognitive load of a large, complex model still pose a significant challenge for new team members?

Word count: 209

References

Baß, J., Winter, M., Pryss, R. and Reichert, M. (2024) ‘Exploring Comprehension Strategies of Modular Process Models: A Combined Eye-Tracking and Concurrent Think-Aloud Study’, *Brain Sciences*, 14(4), p. 303. Available at: https://doi.org/10.3390/brainsci14040303

Heras, F. (2024) ‘Evolution and Comparative Analysis of Enterprise Architecture Tools’, *Communications in Computer and Information Science*, 2132, pp. 113-131. Available at: https://doi.org/10.1007/978-3-031-67904-9_7