Designing a Resilient Cloud-Native Framework for Secure and Scalable News Media Platforms

Ratana SOTH*, Leangsiv SOK†
Paragon International University
Phnom Penh, Cambodia
*rsoth@paragoniu.edu.kh, †lsok@paragoniu.edu.kh

***Abstract*—Online news platforms in Cambodia have become a primary channel for public information, yet many still run on monolithic, single-server deployments of WordPress or custom PHP applications. News platforms face a distinctive load and threat profile: unpredictable breaking-news traffic surges, sustained distributed denial-of-service (DDoS) campaigns against journalism organizations, and a documented history of compromise and defacement of Cambodian outlets. This paper proposes the Resilient News Media Framework (RNMF), a cloud-native hosting and delivery framework that provides a resilient substrate on which a newsroom's content management system (CMS) operates in a decoupled (headless) mode. RNMF combines edge protection (CDN/WAF), load balancing, separation of the public reader plane from the editorial plane, publish-aware caching, a replicated PostgreSQL data tier, observability, and zero-trust administrative access. Following a Design Science Research (DSR) methodology, we (i) establish the problem with evidence on threats and on the platforms used by Cambodian outlets, (ii) map news-specific requirements to architectural mechanisms, (iii) analyze development, operational, ecosystem, and time-to-market costs together with a phased adoption path, and (iv) report qualitative operational validation from a production deployment. Benchmark figures are withheld for infrastructure confidentiality; we discuss this limitation and outline a quantitative evaluation agenda.**

***Index Terms*—Cloud-native architecture, news media platforms, headless CMS, resilience engineering, DDoS mitigation, zero trust, design science research**

I. Introduction

Online news has become a primary channel through which Cambodians receive public information. In late 2025 Cambodia had about 12.0 million internet users (67.3% of the population) and 14.0 million social media user identities [1]. Social platforms drive bursts of traffic to news websites when stories break, so a news site's availability directly affects public access to information during elections, emergencies, and crises.

News platforms are also frequent targets of cyberattacks. Under Cloudflare's Project Galileo, journalism and media organizations were the most attacked category of protected public-interest organizations. Between May 2024 and March 2025, about 97 billion malicious requests against 315 such organizations were mitigated, an average of roughly 290 million per day, and 92.88% of the mitigated traffic was DDoS [2]. Cambodian outlets have been attacked directly. In 2014, the website and online archive of DAP News were wiped and taken offline by a hacktivist group; the outlet had no backup [3]. In 2018, malicious code planted on the Phnom Penh Post website redirected selected visitors to a credential-harvesting page, a campaign attributed to a state-linked threat actor [4]. During the 2025 Cambodia–Thailand border tensions, 139 attacks by 19 hacktivist groups, including DDoS and website defacement, were observed within two weeks [5].

Despite this exposure, many Cambodian outlets still run monolithic stacks. In these stacks, the CMS, the database, and the administrative interface share one server, and a single traffic surge or plugin vulnerability can take the whole site offline (Section III). Existing literature compares CMS products [6], studies cloud-native systems in general [12], [13], or examines cloud migration [16]. It does not specify how a resource-constrained newsroom should structure its delivery infrastructure around news-specific load and threat patterns, or what that would cost.

To address this gap, we propose the *Resilient News Media Framework* (RNMF). RNMF does not replace a CMS. It is the hosting and delivery substrate on which a newsroom's CMS (an existing WordPress installation used headlessly, or a purpose-built content service) runs. The contributions are:

1) an evidence-based problem analysis of threats to news platforms and a survey of the platforms used by Cambodian outlets;
2) the RNMF reference architecture, with an explicit mapping from news-specific requirements to architectural mechanisms (Table II);
3) a cost analysis covering development, operational, ecosystem, and time-to-market costs, with a phased adoption path for small newsrooms;
4) qualitative validation from a production deployment, an assessment of the framework's limits, and an evaluation agenda.

Fig. 1 summarizes the transition from traditional CMS hosting to RNMF.

II. Related Work

CMS platforms for news publishing. Martinez-Caro et al. [6] compared Joomla!, Drupal, and WordPress using a newspaper's online edition as the motivating case. They found that each system makes different trade-offs between usability, role management, SEO, and security, and that most vulnerabilities originate in third-party extensions rather than in the core. Later ecosystem data supports this finding. WordPress powers 39.6%

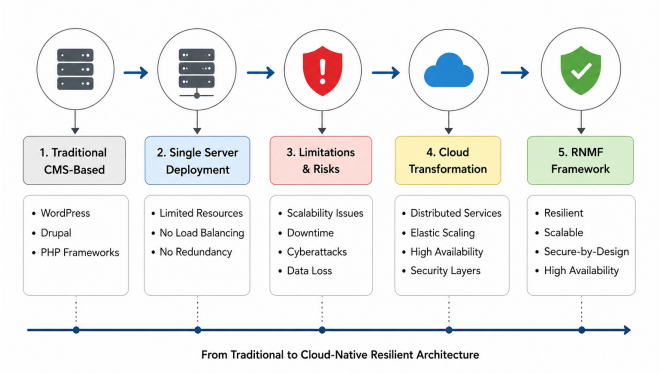


Fig. 1. Evolution of news media infrastructure from single-server CMS hosting toward RNMF.

of all websites and holds 58.4% of the CMS market [7]. Of the 7,966 new WordPress-ecosystem vulnerabilities disclosed in 2024, 96% were in plugins and 33% were still unpatched at public disclosure [8]. Common web risks are catalogued in the OWASP Top 10 [9]. These studies evaluate the CMS software itself, not the infrastructure around it.

Headless CMS. Decoupled (headless) CMS architectures separate the content repository from the presentation layer and deliver content through APIs [10]. Eassa [11] combined a headless CMS with serverless computing and reported average response times of 88–782 ms at up to 10,000 concurrent users. That evaluation was not news-specific, and it relied on a commercial SaaS CMS, which raises cost and data-residency concerns for small outlets.

Cloud-native systems and migration. Kratzke and Quint [12] systematically mapped the literature on cloud-native applications, and Gannon et al. [13] described cloud-native applications in terms of elasticity, microservices, and continuous delivery. Container orchestration has its roots in Borg, Omega, and Kubernetes [14], and Newman [15] discusses the trade-offs of service decomposition. Jamshidi et al. [16] reviewed cloud migration research and highlighted the need for migration processes and tooling. Villamizar et al. [17] showed that the choice of architecture has a material effect on infrastructure cost. This motivates our cost analysis.

Traffic surges, DDoS, and zero trust. Jung et al. [18] characterized flash crowds, the sudden legitimate surges typical of breaking news, and showed that CDNs and caching are the primary means of absorbing them. They also showed that flash crowds must be distinguished from DDoS traffic. Basiri et al. [19] introduced chaos engineering as a way to verify resilience. NIST SP 800-207 [20] defines zero-trust architecture, which we apply to editorial and administrative access.

Gap. No prior work we are aware of combines (a) news-specific load and threat patterns, (b) the realities of newsrooms in developing digital ecosystems such as Cambodia, and (c) cost-aware, incremental adoption into a single reference architecture.

III. Cambodian News Platform Landscape

To ground the problem in observed practice rather than assumption, we checked the homepages of several Cambodian news outlets for publicly visible technology markers, such as WordPress asset paths (`wp-content`) and CDN signatures, on 18 October 2026. Table I shows the results. Two outlets run WordPress, and two run custom PHP-style web applications serving images from separate media hosts. One custom-application outlet showed signs of a CDN (Cloudflare). These observations are consistent with the two dominant deployment styles discussed in this paper. The sample is small, and markers show only what a site exposes publicly. A wider survey is part of our future work (Section X).

TABLE I
Publicly observable platform markers of selected Cambodian news outlets (homepage inspection, 18 Oct. 2026)

Outlet	Platform	Evidence
Phnom Penh Post	WordPress	`/wp-content/`
Cambodia Daily (Eng.)	WordPress	`/wp-content/`
Fresh News	Custom application	separate media host
Thmey Thmey	Custom application	CSRF token, Cloudflare

Combined with the incidents above [3]–[5], this landscape motivates three requirements. A Cambodian news platform must stay available through surges and attacks, contain the impact of a compromised CMS or plugin, and be affordable for small newsrooms.

IV. Research Methodology

We follow Design Science Research (DSR) [22] and the process model of Peffers et al. [23]. The activities were as follows. (1) *Problem identification*: threat evidence and the landscape survey (Sections I and III). (2) *Objectives*: availability under surges and attacks, containment of compromise, maintainability, and bounded cost. (3) *Design and development*: the RNMF artifact (Section V). (4) *Demonstration*: deployment in a production news environment (Section VII). (5) *Evaluation*: qualitative operational validation against the objectives, and a cost analysis (Section VIII). (6) *Communication*: this paper. Because the host organization operates in a high-risk threat environment, it did not permit us to publish raw performance and attack telemetry. Evaluation is therefore qualitative, and Section IX discusses this limitation.

V. Proposed Resilient News Media Framework

A. Design Principles

RNMF follows six principles: high availability, elastic scalability, secure-by-design, maintainability, cost-efficiency, and incremental adoptability. The last principle means each layer can be introduced independently, so a newsroom never needs a disruptive full migration.

B. Architecture

RNMF is organized into nine layers (Fig. 2): (1) a CDN/WAF edge layer (Cloudflare in our deployment); (2) load balancing; (3) a frontend rendering layer for web and mobile; (4) a backend layer for editorial and business logic; (5) an API layer exposing content through REST interfaces [10]; (6) a Redis cache for sessions, query results, and rendered fragments; (7) a PostgreSQL primary with replicas and backups [21]; (8) monitoring, logging, and alerting; and (9) a cross-cutting security layer.

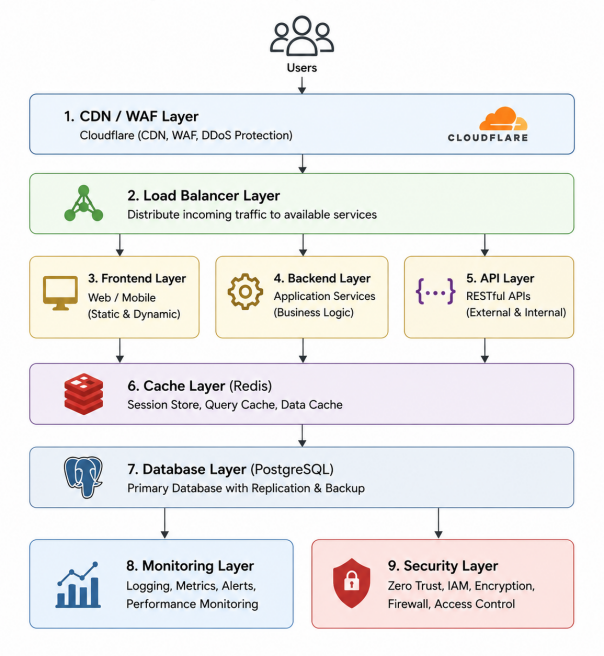


Fig. 2. RNMF hybrid layered and distributed architecture.

C. Relationship to the CMS

RNMF is not a CMS. It separates two planes. The *reader plane* (layers 1–3, 5, 6) serves anonymous public traffic, almost entirely from cache. The *editorial plane* (layer 4) hosts the CMS used by journalists and editors and is reachable only through zero-trust access. The CMS acts as a headless content service. Two adoption options exist: an existing WordPress installation can be kept as the editorial back office and exposed read-only through its REST API, or the newsroom can build or adopt a dedicated content service. Under both options, a vulnerable plugin or a stolen editor credential no longer exposes the public site directly, and a surge on the public site does not degrade the newsroom's ability to publish.

D. News-Specific Mechanisms

The base topology of RNMF resembles that of any horizontally scalable web service. What makes it specific to news media is how each layer is configured for newsroom requirements. Table II summarizes this mapping.

TABLE II
MAPPING OF NEWS-MEDIA REQUIREMENTS TO RNMF MECHANISMS

Newsroom requirement	**RNMF mechanism**
Breaking-news flash crowds [18]	Edge caching of article pages with short TTL and stale-while-revalidate; origin shielding; read replicas
Corrections and updates must appear immediately	Publish/update webhooks trigger targeted CDN and Redis purges by URL and tag
Politically motivated DDoS and defacement [2], [5]	Always-on edge DDoS mitigation; WAF rules; "under attack" mode for crisis periods; static fallback pages
Editor credential and plugin compromise [4], [8]	Separate editorial plane behind zero-trust access with MFA [20]; read-only public API
Archive loss [3]	Replicated PostgreSQL; off-site, versioned backups; media in object storage
Scheduled and embargoed stories	Scheduled publishing handled in the editorial plane; content not exposed to the reader plane before release
Khmer/English multilingual content	UTF-8 Khmer-safe rendering; language-aware cache keys
Social and search crawler load	Bot management that allows verified crawlers and rate-limits scrapers

VI. Security and Scalability Engineering

Threat model. We consider four adversaries: (i) volumetric and application-layer DDoS operators, including hacktivists during political events; (ii) attackers exploiting CMS, plugin, and OWASP-class vulnerabilities [8], [9]; (iii) credential attackers targeting editor accounts; and (iv) actors seeking to alter or destroy published content or archives.

Defense in depth. Fig. 3 shows the five security layers. They cover edge filtering, network segmentation, application controls, data protection, and operational monitoring. Zero-trust access gates all administrative paths [20], and the origin accepts traffic only from the edge provider, so attackers cannot bypass the WAF.

Scalability. The reader plane scales mainly through caching. Because most article requests are identical, the edge and Redis absorb surges, and the origin serves only cache misses and editorial traffic. Stateless frontend and API instances behind the load balancer can be added horizontally, and PostgreSQL read replicas take read load off the primary.

VII. Implementation and Operational Validation

RNMF was deployed in a production Cambodian digital news environment with continuous availability requirements, fluctuating traffic, and elevated cyber risk. The migration was staged in this order: edge CDN/WAF; origin lock-down; caching; separation of the editorial and reader planes; database modernization to replicated PostgreSQL; and observability. Validation was based on the operators' assessment against the DSR objectives. Operators reported that the site remained available during high-traffic news events and during attack attempts that the edge layer mitigated. They also observed fewer origin-overload incidents after caching was introduced, and that editors could keep publishing while the public site

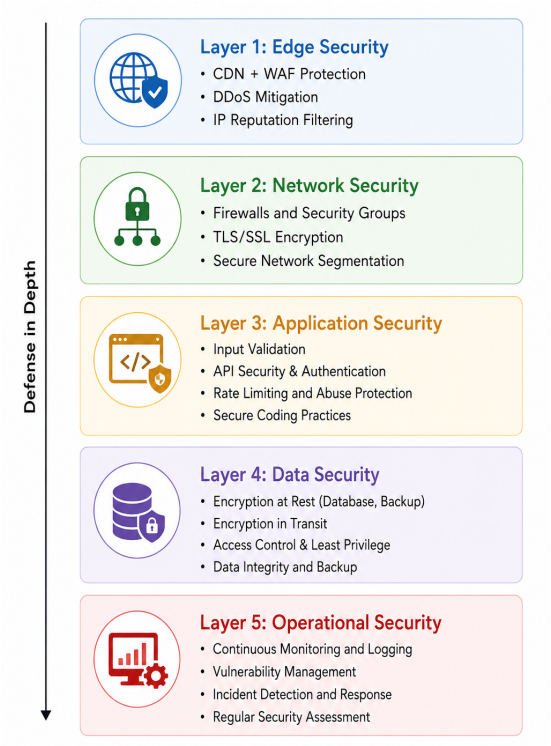


Fig. 3. Layered (defense-in-depth) security model in RNMF.

was under load. Operators also found the platform easier to maintain once the layers were separated. These results are qualitative and come from a single organization. Section IX discusses their limits.

VIII. Cost Considerations

The cost of a resilient architecture can be a barrier for small newsrooms. Table III compares the main cost drivers of the two approaches qualitatively, consistent with the finding of [17] that architectural choices shift costs between development and operations.

TABLE III
Qualitative comparison of cost dimensions

Dimension	Traditional CMS hosting	RNMF
Development	Low; themes and plugins	Higher; decoupled frontend, API, purge integration
Operational	Low fixed cost; high incident cost (downtime, recovery)	Higher fixed cost (more services, monitoring); lower incident cost; free/low-cost CDN tiers offset bandwidth
Ecosystem	Large plugin market; plugin-driven risk	Mature open-source components (Redis, PostgreSQL); needs DevOps skills
Time-to-market	Days to weeks	Weeks to months for full adoption; reduced by phasing

To keep costs bounded, RNMF is adopted in three phases. *Phase 1* (low cost, high impact): put the existing site behind a CDN/WAF, restrict the origin to edge traffic, and add off-site backups. *Phase 2*: add Redis caching and move administrative access behind zero trust. *Phase 3*: decouple the reader and editorial planes and replicate the database. Each phase delivers resilience benefits independently, so newsrooms can stop at the phase their budget and skills allow.

IX. Discussion

Generality. The individual components of RNMF (CDN, load balancer, cache, relational database) are standard. Its contribution is the arrangement and configuration of these components around newsroom requirements and the threat environment in which news outlets operate. The main elements are the editorial/reader plane separation, publish-aware cache invalidation, crisis-mode edge policies, and archive protection (Table II). The architecture could be applied to other high-visibility public-information services, such as government portals. We regard this transferability as a strength rather than a limitation.

Limitations. First, the evaluation is qualitative and comes from a single deployment, so we cannot yet report quantitative effect sizes. Second, the platform survey is small. Third, RNMF relies on a third-party edge provider, which introduces vendor dependence and potential jurisdictional considerations. Fourth, smaller newsrooms may lack the DevOps skills needed for Phase 3.

X. Future Work

We plan to: (1) conduct a controlled evaluation in a staging replica, with load tests that reproduce breaking-news surges and anonymized, relative metrics (availability, latency percentiles, cache hit ratio, origin offload) that can be published without compromising operational security; (2) extend the platform survey to a representative sample of Cambodian outlets and interview newsroom technical staff about incidents and budgets; (3) develop a quantitative total-cost-of-ownership model for each adoption phase; (4) apply chaos engineering [19] to verify failover of the cache, database, and edge layers; (5) evaluate container orchestration and automated scaling for the reader plane, including ML-based autoscaling; (6) study defenses against AI-driven content scraping and bot traffic; and (7) explore multi-region disaster recovery and data-residency options suitable for Cambodian media.

XI. Conclusion

This paper presented RNMF, a cloud-native framework that gives news organizations a resilient hosting and delivery substrate on which their CMS can operate in decoupled mode. Grounded in evidence of attacks on news organizations and in the platforms used by Cambodian outlets, RNMF maps newsroom requirements to concrete mechanisms, separates the editorial plane from the reader plane, and can be adopted in cost-bounded phases. Infrastructure for newsrooms is increasingly offered as managed services: edge security, managed databases, and orchestration platforms whose autopilot modes remove much manual scaling configuration. We expect this

trend to lower the cost of resilience for small newsrooms. The remaining challenge for outlets in developing digital ecosystems will then be architectural, namely deciding how to arrange these services around journalism's specific risks. RNMF is intended as a reference point for that decision.

REFERENCES

[1] S. Kemp, "Digital 2026: Cambodia," DataReportal, 2025. [Online]. Available: https://datareportal.com/reports/digital-2026-cambodia

[2] Cloudflare Data Insights Team, "Project Galileo 11th anniversary," Cloudflare Radar, Jun. 2025. [Online]. Available: https://radar.cloudflare.com/reports/project-galileo-11th-anniv

[3] S. Sinary, "Anonymous Cambodia takes local news website offline," *The Cambodia Daily*, Apr. 29, 2014.

[4] L. Cochrane, "Vietnamese hackers trigger software trap after Australian sale of newspaper in Cambodia," *ABC News*, May 15, 2018.

[5] Group-IB, "Hacktivist at war: The Cambodia–Thailand cyber escalation, July–August 2025," Group-IB Research Hub, 2025.

[6] J.-M. Martinez-Caro, A.-J. Aledo-Hernandez, A. Guillen-Perez, R. Sanchez-Iborra, and M.-D. Cano, "A comparative study of web content management systems," *Information*, vol. 9, no. 2, Art. no. 27, 2018, doi: 10.3390/info9020027.

[7] W3Techs, "Usage statistics of content management systems," Oct. 2026. [Online]. Available: https://w3techs.com/technologies/overview/content_management

[8] Patchstack, "State of WordPress security in 2025," whitepaper, Mar. 2025.

[9] OWASP Foundation, "OWASP Top 10: 2021," 2021. [Online]. Available: https://owasp.org/Top10/

[10] R. T. Fielding, "Architectural styles and the design of network-based software architectures," Ph.D. dissertation, Univ. California, Irvine, CA, USA, 2000.

[11] A. M. Eassa, "Optimizing web application development: A proposed architecture integrating headless CMS and serverless computing," *Int. J. Comput. Inf.*, vol. 12, no. 1, pp. 103–119, 2025.

[12] N. Kratzke and P.-C. Quint, "Understanding cloud-native applications after 10 years of cloud computing—A systematic mapping study," *J. Syst. Softw.*, vol. 126, pp. 1–16, 2017, doi: 10.1016/j.jss.2017.01.001.

[13] D. Gannon, R. Barga, and N. Sundaresan, "Cloud-native applications," *IEEE Cloud Comput.*, vol. 4, no. 5, pp. 16–21, 2017, doi: 10.1109/MCC.2017.4250939.

[14] B. Burns, B. Grant, D. Oppenheimer, E. Brewer, and J. Wilkes, "Borg, Omega, and Kubernetes," *Commun. ACM*, vol. 59, no. 5, pp. 50–57, 2016, doi: 10.1145/2890784.

[15] S. Newman, *Building Microservices*, 2nd ed. Sebastopol, CA, USA: O'Reilly Media, 2021.

[16] P. Jamshidi, A. Ahmad, and C. Pahl, "Cloud migration research: A systematic review," *IEEE Trans. Cloud Comput.*, vol. 1, no. 2, pp. 142–157, 2013, doi: 10.1109/TCC.2013.10.

[17] M. Villamizar *et al.*, "Cost comparison of running web applications in the cloud using monolithic, microservice, and AWS Lambda architectures," *Serv. Oriented Comput. Appl.*, vol. 11, no. 2, pp. 233–247, 2017, doi: 10.1007/s11761-017-0208-y.

[18] J. Jung, B. Krishnamurthy, and M. Rabinovich, "Flash crowds and denial of service attacks: Characterization and implications for CDNs and web sites," in *Proc. 11th Int. Conf. World Wide Web (WWW)*, 2002, pp. 293–304, doi: 10.1145/511446.511485.

[19] A. Basiri *et al.*, "Chaos engineering," *IEEE Softw.*, vol. 33, no. 3, pp. 35–41, 2016, doi: 10.1109/MS.2016.60.

[20] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," NIST, Gaithersburg, MD, USA, SP 800-207, 2020, doi: 10.6028/NIST.SP.800-207.

[21] M. Kleppmann, *Designing Data-Intensive Applications*. Sebastopol, CA, USA: O'Reilly Media, 2017.

[22] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quart.*, vol. 28, no. 1, pp. 75–105, 2004.

[23] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *J. Manage. Inf. Syst.*, vol. 24, no. 3, pp. 45–77, 2007.