AETHER

Impacts & Mitigation

High-Performance Memory-Safe Web Browser

Contents

Executive Summary 2

Part 1: Risk Management 2

The Risk Matrix 2

Part 2: Environmental Impact (GreenIT) 4

Part 3: Deployment & Resilience 4

Deployment & Migration Strategy 5

Resilience & Mitigation Strategy 5

SPOF (Single Point of Failure) 5

Backups 5

Degraded Mode 5

Executive Summary

Aether is an open-source, high-performance web browser built on Rust, Servo rendering engine, and MPV media player. The project aims to address critical issues in current browsers: excessive RAM and CPU usage, security vulnerabilities from aging C++ codebases, privacy concerns with invasive trackers, and bloated user interfaces.

Key Competitive Advantages:

- Memory efficiency: 240-245 MiB startup (vs. Firefox 475-605 MiB, Chrome 1.2-1.5 GiB)
- Fast startup: 0.90s CPU time (vs. Firefox 3.27s, Chrome 1.15s)
- Memory safety through Rust language
- Optimized 4K video playback with MPV integration
- Built-in ad and tracker blocking

Part 1: Risk Management

The Risk Matrix

Risk	Probability (1 – 5)	Impact (1 – 5)	Criticality (P * I)	Mitigation Strategy	Contingency Plan
Technical					
Servo stops	2	3.5	7	Use an Abstraction Layer (Interface) to avoid hard-coding Aether to Servo's API.	Fork the project and make our own modifications if needed.
Operational					
Somebody doesn't work	2	2	4	Use Agile methodology (Scrum) with weekly sprints and clear task assignments via Jira.	Re-evaluate the MoSCoW backlog and prioritize MVP features to meet deadlines.

Risk	Probability (1 – 5)	Impact (1 – 5)	Criticality (P * I)	Mitigation Strategy	Contingency Plan
Somebody leaves the group	1	3	3	Ensure Knowledge Sharing through mandatory code reviews and up-to-date technical documentation (Wiki).	Redistribute the core tasks among the remaining members and reduce the project scope.
Security					
Data leak	2	5	10	Implement AES-256 encryption for sensitive local data (passwords, history) and a Zero-Telemetry policy.	Immediately revoke compromised API keys and deploy a security patch to force-reset local storage.
Out of sandbox breach between 2 tabs	1	5	5	Implement Process Isolation (Site Isolation) ensuring each tab runs in a restricted, low-privilege OS process.	Disable the affected browser features (e.g., JIT, WebAssembly) via an update

Risk	Probability (1 – 5)	Impact (1 – 5)	Criticality (P * I)	Mitigation Strategy	Contingency Plan
Zero-Day security breach in Servo	2	5	10	Continuous monitoring of CVE databases and integration of automated security scanners in the CI/CD.	Deploy an Emergency Hotfix using a pre-configured rapid-deployment pipeline.

Part 2: Environmental Impact (GreenIT)

Green IT Category	Estimated Environmental Impact	Estimated Environmental Impact
Hosting	**Primary:** Self-hosted on reused/refurbished servers (2+ life cycles) paired with new performance components. **Redundancy:** Eco-designed datacenters (prioritizing OVH).	Drastically reduces e-waste by extending hardware lifespans. Lowers the overall carbon footprint of the infrastructure by utilizing green-conscious providers.
Data Transfer	**Updates:** Compressed and optimized binary files served via custom API. **Web Assets:** Exclusive use of the lightweight WebP image format.	Minimizes bandwidth consumption, reducing the energy required by network hardware to transmit data over the internet.
Compute	**Core App:** Strict optimization of CPU and RAM usage, minimizing system resource overhead.	Lowers local electricity consumption. Crucially, it permits the use of older hardware, directly combating planned obsolescence and consumer e-waste.

Part 3: Deployment & Resilience

Deployment & Migration Strategy

CI/CD: Using GitHub actions to build and perform WPT tests (splits in blocks). Release and Beta binaries are periodically built (1 per week, and security are directly built if tag like this), but the authorization of deployment is manually confirmed.

Migration: the Sqlite files of the app are separated to permit easy migration and update. For the site, nothing is required unless the contact page.

Resilience & Mitigation Strategy

SPOF (Single Point of Failure)

- Database with replication (primary + replica) to avoid a single point of failure (SPOF).
- Backend deployed across multiple instances behind a load balancer.
- Redundant storage on multiple points.
- Automatic monitoring and restarting in case of a crash.
- No single critical component without a backup solution.

Backups

- Incremental backups every hour.
- Encrypted storage.
- Documented disaster recovery procedures.
- Retention: 2 days for hourly backups, 30 days for daily backups.

Degraded Mode

If a server fails, we have many simultaneous instances. The same principle applies to the website: use a load balancer between the different instances.

Document Date: February 10, 2026

Modify Date: November 13, 2026