

Thema: Diary Update

Anfrage an Gemini:

Ich will ein Problem mit dem ohne Theme-Vorlage gecodeten Theme des Digital Diary besprechen. Bzw. es geht um eine möglicherweise nötig werdende „Modernisierung“ des knapp 27 Jahre alten Blogs (seit 3/99), um es zukunftssicher zu machen.

Das aktuelle Theme basiert nicht auf einem großen Theme, das nur geändert wurde, sondern ist exakt für meine Nutzung im Worspress-Blog entwickelt. Es führt also nicht all den Ballast mit, den so ein Großtheme als "eierlegende Wollmilchsau" mitbringt.

Das „Problem“:

Es läuft noch der alte Editor und das alte Widget-System, also keine Gutenberg-Architektur wie für neuere Themes typisch. Derzeit weiß ich nicht, ob das geändert werden muss und wenn ja wie, um auch in Zukunft problemlos zu laufen. **Braucht das Blog eine Modernisierung in Richtung Gutenberg/Block-Theme?** Dabei ist es mir wichtig, die Optik zu erhalten, die seit 1999 durch alle Techniken hindurch ungefähr gleich geblieben ist.

Gemini:

Das ist ein sehr spannendes Projekt – ein "Legacy-Design" auf moderne Füße zu stellen, ohne die Seele des Blogs (und die Performance) zu opfern, ist die Königsdisziplin.

Da du seit 1999 dabei bist, hast du wahrscheinlich viel wertvollen Content. Eine wichtige Vorentscheidung ist, ob du den **Classic Editor** per Plugin weiter nutzen willst oder ob du bereit bist, deine alten Beiträge (ggf. automatisiert) in Blöcke umzuwandeln. Das ist oft die größte Hürde bei so alten Projekten.

Ich:

Wenn ich mir das so überlege: Gibt es eigentlich zwingende Gründe für diese Migration? Was hat es mit der "großen Hürde" auf sich?

Gemini:

Das ist eine sehr berechtigte Frage. Wenn ein System seit über 25 Jahren läuft, ist "Never change a running system" ein verdammt gutes Argument. Es gibt jedoch zwei Ebenen: Die **technische Notwendigkeit** und die **"große Hürde"** beim Content.

Hier ist eine Analyse der Situation:

Warum man überhaupt migrieren sollte (Die Gründe)

Es gibt eigentlich nur drei wirklich zwingende Gründe, warum man das alte System anfassen muss:

1. **PHP-Kompatibilität:** WordPress selbst wird weiterentwickelt. Wenn dein Theme sehr alten Code nutzt (z. B. Funktionen, die in PHP 8.x als "deprecated" markiert wurden), wird der Blog irgendwann bei einem Server-Update auf eine neue PHP-Version weiß bleiben (Error 500).
2. **Sicherheit:** Altes Widget-System und alter Code können Sicherheitslücken enthalten, die in modernen Architekturen durch neue Standards (wie Nonces oder moderne Sanitization) abgefangen werden.
3. **Zukunftssicherheit des "Classic Editors":** Das Plugin "Classic Editor" wird von WordPress offiziell unterstützt, aber das Versprechen gilt nicht ewig. Irgendwann wird der Kern von WordPress so stark auf Blöcke setzen, dass das alte System nur noch mit extremem Aufwand stabil zu halten ist.

Ich:

Was ist die „große Hürde“?

Gemini:

Wenn ich von der "großen Hürde" spreche, meine ich den **Bruch in der Datenbank-Struktur** deiner Beiträge.

- **Der alte Status Quo:** Deine Beiträge von 1999 bis heute bestehen aus reinem HTML/Text in der Datenbank.
- **Die Gutenberg-Welt:** Moderne WordPress-Beiträge bestehen aus "Blöcken", die durch spezielle HTML-Kommentare `<!-- wp:paragraph -->` getrennt sind.

Das Problem:

Wenn du ein modernes "Block Theme" (Full Site Editing) nutzt, erwartet WordPress, dass der Content in Blöcken vorliegt. Wenn du dein altes Theme einfach nur "modernisierst", hast du zwei Möglichkeiten:

1. **Der "Classic-Hybrid"-Weg:** Du behältst alles wie es ist. Die Hürde hier: Neue WordPress-Funktionen (wie einfache Bild-Galerien, Spalten oder moderne Einbettungen) lassen sich im alten Editor nur mühsam nachbauen. Du bleibst technisch in einer Sackgasse.
2. **Die Konvertierung:** Du migrierst deine tausenden Beiträge in das Block-Format. **Das ist die Hürde:** Dabei zerschießt es oft die Formatierung alter Beiträge. Bilder verrutschen, Abstände stimmen nicht mehr. Bei einem Blog von 1999 ist das ein massives Projekt, das händische Nacharbeit erfordert.

Mein Rat für deine Situation: Da dir die **Schlankheit** und die **Optik** extrem wichtig sind, gibt es einen "Königsweg", den viele Puristen wählen:

Das "Modern Classic" Theme:

Du lässt das Theme von einem Entwickler neu programmieren, aber **ohne** Full Site Editing (FSE). Er schreibt ein sauberes, neues PHP-Theme nach aktuellen Standards, das exakt so aussieht wie das alte. Du kannst dann den Classic Editor weiter nutzen, hast aber unter der Haube modernen, schnellen und sicheren Code.

Zusammenfassend: Zwingend ist die Migration nur, wenn dein aktueller Code mit modernen PHP-Versionen bricht oder du merkst, dass Plugins nicht mehr mit deinem alten Theme harmonieren.

Ich: Was ist der "Classic-Hybrid-Weg"?

Gemini:

Der „**Classic-Hybrid-Weg**“ ist im Grunde die pragmatische Brücke zwischen der alten WordPress-Welt (PHP-Templates) und der neuen Welt (Gutenberg/Blöcke).

In deinem Fall bedeutet das: Du behältst die Kontrolle über den Code (wie bei deinem jetzigen Theme), machst ihn aber fit für die Zukunft, ohne dein gesamtes Archiv von 1999 anfassen zu müssen.

Hier sind die Details dieses Ansatzes:

1. Das Theme bleibt PHP-basiert (Classic Theme)

Anstatt ein modernes „Block Theme“ (HTML-basiert) zu bauen, bleibt die Struktur so, wie du sie kennst:

- Die Optik wird über `header.php`, `index.php`, `single.php` und `style.css` gesteuert.
- **Vorteil:** Du hast die volle Kontrolle über jedes HTML-Tag und lädst keinen Ballast. Es gibt keine „Theme-Engine“, die im Hintergrund mitläuft.

2. Punktueller Block-Unterstützung

„Hybrid“ bedeutet, dass das Theme zwar klassisch programmiert ist, aber WordPress sagt: *„Ich weiß, wie man mit Blöcken umgeht, falls der Nutzer sie verwenden will.“*

- Du kannst für **neue** Beiträge den Gutenberg-Editor nutzen (um z. B. moderne Galerien oder Spalten einzubauen).
- **Alte** Beiträge bleiben einfach im „Classic-Block“ oder werden weiterhin über das Classic-Editor-Plugin bearbeitet. Sie sehen im Frontend exakt so aus wie früher.

3. Entkopplung von Design und Editor

Das ist der wichtigste Punkt für dich:

- Im Hybrid-Modell wird das Design im Frontend (was der Leser sieht) weiterhin primär durch dein CSS bestimmt.
- Du bist nicht gezwungen, das „Site Editing“ (FSE) zu nutzen, bei dem man den Header und Footer im Editor zusammenklickt. Dein Header bleibt fest in der `header.php` codiert – sicher vor versehentlichen Änderungen und extrem schnell.

Warum das für dich die beste Lösung sein könnte:

Merkmal	Full Block Theme (Modern)	Classic-Hybrid (Empfehlung)
Code-Kontrolle	Vieles wird von WP generiert	Du bestimmst jede Zeile
Lernkurve	Hoch (JSON/React-Kenntnisse)	Niedrig (PHP/CSS wie gewohnt)

Alte Beiträge Müssen oft konvertiert werden Bleiben unangetastet
Geschwindigkeit Sehr gut (wenn richtig gemacht) Exzellent (da kein Overhead)

Die „große Hürde“ wird damit zum kleinen Schritt:

Die Hürde bei einer kompletten Umstellung auf ein modernes **Block Theme** ist, dass diese Themes darauf angewiesen sind, dass alles ein Block ist. Ein Hybrid-Theme ist da toleranter: Es akzeptiert deinen alten HTML-Content aus den Jahren 1999 bis 2025 klaglos als „einen großen Klumpen Text“ und stellt ihn korrekt dar.

Zusammengefasst: Du modernisierst nur den „Motor“ (den PHP-Code, damit er mit PHP 8.x läuft), behältst aber die Karosserie (dein Design) und die Innenausstattung (deinen alten Workflow). Wäre das ein Ansatz für dich?

Ich: Ja, es geht also nur um ein Update auf moderneren Code, passend für aktuelle PHP-Versionen. Wie könnte ich herausfinden, ob mein Blog (Php 7.4) auf php 8 läuft?

Gemini:

Das ist der kritische Punkt: Da dein Theme seit 1999 „mitgewachsen“ ist, enthält es mit Sicherheit PHP-Funktionen, die in PHP 8 nicht mehr existieren (sogenannte *Breaking Changes*).

Hier sind die drei Wege, um das ohne Risiko zu prüfen:

1. Der "Site Health"-Check (Status Quo)

Zuerst solltest du prüfen, welche Version genau gerade aktiv ist.

- Gehe in deinem WordPress-Dashboard auf **Werkzeuge** → **Website-Zustand**.
- Klicke auf den Tab **Bericht** und öffne den Punkt **Server**.
- Dort steht deine aktuelle PHP-Version (z. B. 7.4.33).

2. Der "Trockentest" mit einem Plugin

Es gibt spezialisierte Plugins, die deinen Code scannen, ohne die PHP-Version wirklich umzustellen.

- **PHP Compatibility Checker:** Dieses Plugin (z. B. von WP Engine) scannt deine Themes und Plugins auf veralteten Code und gibt dir einen Bericht aus, welche Zeilen in PHP 8 zu Fehlern führen würden.
- **Wichtig:** Diese Scanner finden viel, aber nicht alles. Sie sind eher ein "Voralarm".

3. Die "Sichere Testumgebung" (Staging)

Das ist die einzige Methode, um 100% sicher zu sein.

- **Staging-Seite:** Viele Hosts bieten an, mit einem Klick eine Kopie deines Blogs auf einer Test-Adresse zu erstellen.
- **Der Test:** Dort stellst du im Hosting-Panel nur für diese Kopie auf PHP 8.0 oder 8.1 um. Wenn die Seite weiß bleibt ("Critical Error"), weißt du, dass dein Theme Anpassungen braucht.
- **Lokale Kopie:** Alternativ kannst du den Blog mit Tools wie [LocalWP](#) auf deinen eigenen Rechner kopieren und dort einfach per Dropdown die PHP-Version wechseln.

Warum PHP 8 oft Probleme macht:

In PHP 8 wurden viele Dinge, die in PHP 7.4 nur "Warnungen" auslösten, zu "harten Fehlern". Ein Theme von 1999 nutzt oft alte Befehle zur Datenbankabfrage oder Variablen-Verarbeitung, die PHP 8 schlichtweg nicht mehr versteht.

Soll ich dir erklären, wie du in die **Error-Logs** deines Servers schaust? Dort steht bei einem Absturz nämlich exakt die Datei und die Zeile deines Themes, die den Fehler verursacht.