Technischer Bericht

DSGVO-/TTDSG-konforme Speicherverwaltung

"Storage Guard" für die Website Digital & Mobil in Deutschland

Projekt: digital-mobil-deutschland.vercel.app

Branch: test-main

Autor: Artem Finevych

Stand: April 30.04.2026

1. Ausgangslage und Problemstellung

Anlass des Audits

Im Hinblick auf die geplante Migration der Website von der bisherigen Vercel-Subdomain (digital-mobil-deutschland.vercel.app) auf die produktive Domain dmd-nrw.de sowie die anschließende Anmeldung bei der Google Search Console wurde ein vollständiger technischer Audit hinsichtlich der Konformität mit den geltenden datenschutzrechtlichen Anforderungen durchgeführt — insbesondere mit der Datenschutz-Grundverordnung (DSGVO), dem Telekommunikation-Digitale-Dienste-Datenschutz-Gesetz (TDDDG) sowie der einschlägigen deutschen Rechtsprechung.

Ziel des Audits war es sicherzustellen, dass die Website vor der Verbreitung unter eigenem Domainnamen und vor der Indexierung durch Suchmaschinen den höchsten Anforderungen an Datenschutz, Transparenz und Nutzerrechte entspricht. Der vorliegende Bericht dokumentiert die identifizierten Schwachstellen, die durchgeführten technischen und organisatorischen Maßnahmen sowie das Ergebnis der abschließenden Verifikation.

Die Website **"Digital & Mobil in Deutschland"** (digital-mobil-deutschland.vercel.app) ist eine zweisprachige (DE/RU) Single-Page-PWA, die unter anderem aktuelle Wetterdaten, Luftqualität, geomagnetische Aktivität, den Wechselkurs EUR/UAH sowie einen KI-Chat-Assistenten für die Bevölkerung von Hattingen und Umgebung bereitstellt.

Bei einer technischen Prüfung im Web-Inspector (Safari iOS) wurde festgestellt, dass die genannten Funktionsdaten **bereits beim ersten Seitenaufruf** — also **noch vor jeder Einwilligung** durch den Nutzer im Cookie-Banner — in den `sessionStorage` bzw.

`localStorage` des Browsers geschrieben wurden. Konkret betraf dies folgende Schlüssel:

- `weatherData` — vollständige OpenWeatherMap-Antwort einschließlich Geokoordinaten
- `aqiData` — Luftqualitätsindex und Schadstoffkonzentrationen
- `magnetData` — Kp-Index der geomagnetischen Aktivität (NOAA SWPC)
- `eurRate` — Wechselkurs EUR/UAH der Nationalbank der Ukraine
- `dmd_chat_history_de` / `dmd_chat_history_ru` — vollständiger Verlauf des KI-Chat-Assistenten im `localStorage` (persistent)
- `userCity` — die vom Nutzer ausgewählte Stadt für die Wetteranzeige im `localStorage` (persistent)

> **Rechtliche Bewertung**
>
> Das ungefragte Setzen funktionaler Browser-Speichereinträge stellt einen Verstoß gegen **§ 25 Abs. 1 TDDDG** (Telekommunikation-Digitale-Dienste-Datenschutz-Gesetz) sowie gegen **Art. 6 Abs. 1 DSGVO** dar. Nach geltender deutscher Rechtsprechung (BGH, Urteil vom 28.05.2020 — I ZR 7/16, *"Cookie-Einwilligung II"*) ist für jede nicht technisch zwingend notwendige Speicherung eine **aktive, vorherige und informierte Einwilligung** des Nutzers erforderlich.

2. Lösungsansatz: Das Storage-Guard-Pattern

Zur strukturellen Behebung wurde das Konzept eines **"Storage Guard"** eingeführt — einer zentralen, global verfügbaren JavaScript-Funktion, die **vor jeder Schreiboperation** in den Browser-Speicher abgefragt wird und entscheidet, ob die Einwilligung des Nutzers vorliegt.

2.1 Eigener Einwilligungsschlüssel

Statt sich auf interne, undokumentierte Schlüssel der eingesetzten Drittanbieter-Bibliothek **Silktide Consent Manager** zu verlassen (deren Schlüsselnamen sich zwischen Versionen ändern können), wurde ein **eigener, kontrollierter Einwilligungsstatus** eingeführt:

```
localStorage.setItem('cookieConsent', 'accepted'); // bei Zustimmung
localStorage.setItem('cookieConsent', 'rejected'); // bei Ablehnung
```

Dieser Schlüssel wird ausschließlich über die `onAccept` / `onReject`-Callbacks der Kategorie **"Funktional"** im Cookie-Banner gesetzt und bildet die alleinige Grundlage für die Storage-Guard-Entscheidung.

2.2 Globale Guard-Funktion

In `js/cookie-consent.js` wurde folgende global verfügbare Funktion implementiert, die in allen funktionalen Skripten vor `setItem(...)`-Aufrufen verwendet wird:

```
window.canSaveToStorage = function () {     try {         return
localStorage.getItem('cookieConsent') === 'accepted';     } catch (e) {
return false;  // Sichere Voreinstellung     } };
```

Diese Funktion wurde anschließend in **allen relevanten JavaScript-Dateien** als Vorbedingung für jede `sessionStorage.setItem(...)`- bzw. `localStorage.setItem(...)`-Operation eingebaut:

- `js/weather.js` — Wetter und Luftqualität
- `js/weather-display.js` — Fallback-Anzeige beim Seitenwechsel
- `js/eur-rate.js` — Wechselkurs EUR/UAH
- `js/magnet.js` — Geomagnetische Aktivität

Beispielhafte Implementierung für die Wetterdaten in `js/weather.js`:

```
if (window.canSaveToStorage && window.canSaveToStorage()) {     try {
sessionStorage.setItem('weatherData', JSON.stringify(d));     } catch (e) {} }
```

2.3 Reaktive Aktivierung nach Einwilligung

Damit nach erteilter Einwilligung die zwischengespeicherten Daten **sofort** verfügbar sind (ohne dass der Nutzer die Seite neu laden muss), wurden die zentralen Datenabruffunktionen global exportiert und im `onAccept`-Callback der Kategorie **"Funktional"** erneut aufgerufen:

```
onAccept: function () {     try { localStorage.setItem('cookieConsent',
'accepted'); } catch (e) {}      // Widgets neu rendern – Storage Guard erlaubt
nun setItem()     if (typeof window.getWeather === 'function')
window.getWeather();     if (typeof window.getGeomagneticActivity === 'function')
window.getGeomagneticActivity();     if (typeof window.loadEurRate === 'function')
window.loadEurRate();     loadNews(); }
```

2.4 Aktive Bereinigung beim Widerruf

Im Sinne von **Art. 7 Abs. 3 DSGVO** (Widerruf der Einwilligung muss "so einfach wie deren Erteilung" sein) bewirkt der `onReject`-Callback nicht nur das Setzen des Status auf `"rejected"`, sondern entfernt auch alle bereits vorhandenen funktionalen Speichereinträge:

```
onReject: function () {     try { localStorage.setItem('cookieConsent',
'rejected'); } catch (e) {}      // Aufräumen vorhandener Daten – auch bei
nachträglichem Widerruf     sessionStorage.removeItem('weatherData');
sessionStorage.removeItem('aqiData');     sessionStorage.removeItem('magnetData');
sessionStorage.removeItem('eurRate');     localStorage.removeItem('userCity'); }
```

3. Sonderbehandlung des KI-Chat-Assistenten

Der KI-Chat-Assistent **"DMD Assistent"** (basierend auf Llama 3.3 70B via Groq API über einen serverseitigen Proxy) stellt einen Sonderfall dar: Er wird **aktiv vom Nutzer aufgerufen**, und der Konversationsverlauf muss zwischen Seitenwechseln innerhalb der laufenden Sitzung erhalten bleiben, damit der Nutzer den Kontext nicht verliert.

3.1 Migration: localStorage → sessionStorage

Die ursprüngliche Implementierung speicherte den Chat-Verlauf im `localStorage` (persistent, bis zur manuellen Löschung). Dies hätte unter strenger Auslegung der DSGVO eine Einwilligung erforderlich gemacht, da Chat-Inhalte potenziell personenbezogene Daten enthalten können (z.B. Symptome, Adressen, Familienangelegenheiten).

Die Speicherung wurde auf `sessionStorage` umgestellt. Dieser wird vom Browser **automatisch und vollständig beim Schließen des Tabs gelöscht** und fällt damit unter die Ausnahme nach **§ 25 Abs. 2 Nr. 2 TDDDG** ("technisch unbedingt erforderlich, um einen vom Nutzer ausdrücklich gewünschten Telemediendienst zur Verfügung zu stellen").

3.2 Vereinheitlichung der Sprachschlüssel

Ursprünglich wurde der Chat-Verlauf unter sprachspezifischen Schlüsseln gespeichert (`dmd_chat_history_de` bzw. `dmd_chat_history_ru`). Beim Wechsel zwischen DE- und RU-Versionen einer Seite ging der Verlauf jeweils verloren. Die Schlüssel wurden zu `dmd_chat_history` vereinheitlicht — der Verlauf bleibt nun bei Sprachwechsel erhalten.

3.3 Behebung des Tab-Verlustproblems

Eingebettete `[PAGE:...]`-Marker im Chat öffneten ihre Ziele in einer **neuen Browser-Vkladka** (`target="_blank"`). Da `sessionStorage` pro Tab isoliert ist, war der Chat-Verlauf in der neuen Tab unsichtbar. Die Lösung:

- Interne PAGE-Links öffnen nun in der **aktuellen Tab** — der `sessionStorage` bleibt erhalten
- Externe URL-Links (HTTPS) öffnen weiterhin in einer neuen Tab mit `rel="noopener noreferrer"` — Sicherheitsstandard

> **Rechtliche Grundlage**
>
> Die Speicherung des Chat-Verlaufs im `sessionStorage` erfolgt auf Grundlage von **§ 25 Abs. 2 Nr. 2 TDDDG** und **Art. 6 Abs. 1 lit. f DSGVO** — als technisch notwendige Maßnahme zur Bereitstellung einer vom Nutzer aktiv aufgerufenen Funktion innerhalb der laufenden Sitzung. Eine zusätzliche Einwilligung ist daher nicht erforderlich.

4. Datenfluss zur API: Server-seitiger Proxy

Ein zentraler Bestandteil der DSGVO-Konformität dieses Projekts ist der konsequente Einsatz von **eigenen Vercel Serverless Functions als Proxy** für alle externen API-Aufrufe. Der Browser des Nutzers verbindet sich **zu keinem Zeitpunkt direkt** mit Drittanbieter-APIs:

Endpoint	Datenquelle	Datenflussisolierung
`/api/weather`	OpenWeatherMap (UK)	Nutzer-IP nicht weitergegeben
`/api/air`	OpenWeatherMap Air Pollution (UK)	Nutzer-IP nicht weitergegeben
`/api/magnet`	NOAA SWPC (USA)	Nutzer-IP nicht weitergegeben
`/api/eur-rate`	Nationalbank der Ukraine	Nutzer-IP nicht weitergegeben
`/api/chat`	Groq Inc. (USA, Llama 3.3 70B)	Nutzer-IP nicht weitergegeben

An die jeweiligen Drittanbieter wird ausschließlich die **IP-Adresse des Vercel-Servers** übermittelt, niemals die IP-Adresse des Nutzers. Der Nutzer bleibt für alle externen Datenquellen **anonym**.

5. Verifikation und Ergebnisse

Die Korrektheit der Implementierung wurde mittels **Safari Web Inspector** am iPhone-Endgerät über mehrere Vercel-Preview-Deployments hinweg verifiziert.

5.1 Vor dem ersten Klick im Cookie-Banner

- **localStorage:** leer (keine projektbezogenen Schlüssel)
- **sessionStorage:** leer (außer Vercel-Preview-Toolbar-Schlüssel, die im Production-Domain nicht erscheinen)
- **Sichtbar im UI:** Wetter (16°C, Hattingen), Luftqualität (AQI 3), Wechselkurs (51,46 ₴) — alle Daten werden korrekt angezeigt, jedoch nicht im Browser zwischengespeichert

5.2 Nach Klick "Funktional akzeptieren"

- `localStorage['cookieConsent']` = `"accepted"`
- `localStorage['userCity']` = `"Hattingen"` (oder benutzerdefinierte Stadt)
- `sessionStorage`: `weatherData`, `aqiData`, `magnetData`, `eurRate`, `dmd_chat_history`

5.3 Nach Klick "Nur notwendige" / Widerruf

- `localStorage['cookieConsent']` = `"rejected"`
- Vorhandene funktionale Daten werden **automatisch entfernt**
- UI funktioniert weiterhin (Daten werden bei jedem Seitenwechsel neu geladen)

> **Ergebnis**
>
> Die Website **erfüllt nun die Anforderungen** der DSGVO und des TDDDG vollständig. Vor jeder Einwilligung des Nutzers werden **keine** funktionalen Daten im Browser-Speicher abgelegt. Die Einwilligung ist granular, jederzeit widerrufbar und über das stets sichtbare Cookie-Symbol unten links erreichbar.

6. Begleitende rechtliche Maßnahmen

Über die rein technische Implementierung hinaus wurden folgende juristische Dokumente **vollständig überarbeitet**, um die neue Architektur korrekt abzubilden:

- **Datenschutzerklärung (DE/RU) — Version 3:** alle 23 Abschnitte aktualisiert; Abschnitt 4 nach Storage-Kategorien neu strukturiert; Abschnitt 14e präzisiert (Chat in sessionStorage, technische Notwendigkeit nach § 25 Abs. 2 Nr. 2 TDDDG); Abschnitt 20 als Übersichtstabelle aller Speicherschlüssel
- **Impressum (DE/RU):** ergänzt um KI-Hinweis nach DDG, Streitbeilegungsklausel (ODR-VO/VSBG), Bildnachweis und Reaktionszeit (24h)
- **Cookie-Banner-Texte:** alle drei Kategorien (Notwendig / Funktional / Externe Medien) inhaltlich an die tatsächliche Implementierung angepasst
- **Barrierefreiheitserklärung (DE/RU):** freiwillig erstellt; orientiert sich an WCAG 2.1 AA und EN 301 549; bekannte Barrieren transparent aufgelistet

- **Web Components:** `<legal-disclaimer>` für medizinische, rechtliche und Migrations-Themen; `<legal-footer>` mit Impressum/Datenschutz/Barrierefreiheit-Links auf jeder Seite (Zwei-Klick-Regel)

7. Zusammenfassung

Mit der Einführung des Storage-Guard-Patterns, der Migration des Chat-Verlaufs in den `sessionStorage`, der konsequenten Server-seitigen API-Abstraktion und der vollständigen Aktualisierung der juristischen Dokumente erfüllt die Website **"Digital & Mobil in Deutschland"** die Anforderungen der **DSGVO**, des **TDDDG** und der einschlägigen deutschen Rechtsprechung.

Das System ist **nachvollziehbar dokumentiert**, **technisch verifizierbar** (im Web Inspector jederzeit überprüfbar) und **rechtlich verteidigbar**. Die Trennung zwischen technisch notwendiger und einwilligungsbedürftiger Speicherung ist für externe Prüfer und die Aufsichtsbehörde transparent.

Технический отчёт

Соответствие нормам DSGVO/TTDSG

«Storage Guard» для сайта Digital & Mobil in Deutschland

1. Исходное положение и проблема

Anlass des Audits

Im Hinblick auf die geplante Migration der Website von der bisherigen Vercel-Subdomain (digital-mobil-deutschland.vercel.app) auf die produktive Domain dmd-nrw.de sowie die anschließende Anmeldung bei der Google Search Console wurde ein vollständiger technischer Audit hinsichtlich der Konformität mit den geltenden datenschutzrechtlichen Anforderungen durchgeführt — insbesondere mit der Datenschutz-Grundverordnung (DSGVO), dem Telekommunikation-Digitale-Dienste-Datenschutz-Gesetz (TDDDG) sowie der einschlägigen deutschen Rechtsprechung.

Ziel des Audits war es sicherzustellen, dass die Website vor der Verbreitung unter eigenem Domainnamen und vor der Indexierung durch Suchmaschinen den höchsten Anforderungen an Datenschutz, Transparenz und Nutzerrechte entspricht. Der vorliegende Bericht dokumentiert die identifizierten Schwachstellen, die durchgeführten technischen und organisatorischen Maßnahmen sowie das Ergebnis der abschließenden Verifikation.

Сайт **«Digital & Mobil in Deutschland»** (digital-mobil-deutschland.vercel.app) — это двуязычное (DE/RU) одностраничное PWA-приложение, которое предоставляет, помимо прочего, актуальные данные о погоде, качестве воздуха, геомагнитной активности, курс EUR/UAH, а также ИИ-чат-ассистента для жителей Хаттингена и окрестностей.

В ходе технической проверки в Web-Inspector (Safari iOS) было обнаружено, что вышеперечисленные функциональные данные записывались в `sessionStorage` и `localStorage` браузера **уже при первой загрузке страницы** — то есть **ещё до того, как пользователь дал согласие** в баннере cookie. Это касалось следующих ключей:

- `weatherData` — полный ответ OpenWeatherMap, включая координаты
- `aqiData` — индекс качества воздуха и концентрации загрязняющих веществ
- `magnetData` — индекс геомагнитной активности Kp (NOAA SWPC)
- `eurRate` — курс EUR/UAH Национального банка Украины
- `dmd_chat_history_de` / `dmd_chat_history_ru` — полная история ИИ-чата в `localStorage` (постоянное хранение)
- `userCity` — выбранный пользователем город для виджета погоды в `localStorage` (постоянное хранение)

> **Правовая оценка**
>
> Несанкционированное создание функциональных записей в браузерном хранилище является нарушением **§ 25 абз. 1 TDDDG** (Закона о защите данных в области телекоммуникаций и цифровых услуг), а также **ст. 6 абз. 1 GDPR**. Согласно действующей немецкой судебной практике (BGH, решение от 28.05.2020 — I ZR 7/16, *«Cookie-Einwilligung II»*), для любого хранения, не являющегося технически необходимым, требуется **активное, предварительное и информированное согласие** пользователя.

2. Решение: паттерн «Storage Guard»

Для структурного решения проблемы был внедрён концепт **«Storage Guard»** — централизованной, глобально доступной JavaScript-функции, которая опрашивается **перед каждой операцией записи** в браузерное хранилище и определяет, дал ли пользователь согласие.

2.1 Собственный ключ согласия

Вместо того чтобы полагаться на внутренние, недокументированные ключи библиотеки **Silktide Consent Manager** (имена которых могут меняться от версии к версии), был введён **собственный, контролируемый статус согласия**:

```
localStorage.setItem('cookieConsent', 'accepted'); // при согласии
localStorage.setItem('cookieConsent', 'rejected'); // при отказе
```

Этот ключ устанавливается исключительно через коллбэки `onAccept` и `onReject` категории **«Функциональные»** в баннере cookie и является единственным основанием для решения Storage Guard.

2.2 Глобальная функция Guard

В файле `js/cookie-consent.js` реализована следующая глобально доступная функция, которая используется во всех функциональных скриптах перед вызовами `setItem(...)`:

```
window.canSaveToStorage = function () {     try {         return
localStorage.getItem('cookieConsent') === 'accepted';     } catch (e) {
return false;  // Безопасное значение по умолчанию     } };
```

Эта функция была встроена во **все соответствующие JavaScript-файлы** в качестве предварительного условия для каждой операции `sessionStorage.setItem(...)` и `localStorage.setItem(...)`:

- `js/weather.js` — погода и качество воздуха
- `js/weather-display.js` — fallback-отображение при переходах между страницами
- `js/eur-rate.js` — курс EUR/UAH
- `js/magnet.js` — геомагнитная активность

Пример реализации для данных о погоде в `js/weather.js`:

```
if (window.canSaveToStorage && window.canSaveToStorage()) {     try {
sessionStorage.setItem('weatherData', JSON.stringify(d));     } catch (e) {} }
```

2.3 Реактивная активация после согласия

Чтобы после получения согласия кэшированные данные были доступны **сразу** (без необходимости перезагрузки страницы пользователем), центральные функции загрузки данных были экспортированы в глобальную область видимости и заново вызываются в коллбэке `onAccept`:

```
onAccept: function () {     try { localStorage.setItem('cookieConsent',
'accepted'); } catch (e) {}      // Перерисовка виджетов — Storage Guard теперь
разрешает setItem()     if (typeof window.getWeather === 'function')
window.getWeather();     if (typeof window.getGeomagneticActivity === 'function')
window.getGeomagneticActivity();     if (typeof window.loadEurRate === 'function')
window.loadEurRate();     loadNews(); }
```

2.4 Активная очистка при отзыве согласия

В соответствии со **ст. 7 абз. 3 GDPR** (отзыв согласия должен быть «так же прост, как его получение») коллбэк `onReject` не только устанавливает статус `«rejected»`, но и удаляет все уже существующие функциональные записи:

```
onReject: function () {     try { localStorage.setItem('cookieConsent',
'rejected'); } catch (e) {}      // Очистка существующих данных — даже при
последующем отзыве     sessionStorage.removeItem('weatherData');
sessionStorage.removeItem('aqiData');     sessionStorage.removeItem('magnetData');
sessionStorage.removeItem('eurRate');     localStorage.removeItem('userCity'); }
```

3. Особая обработка ИИ-чат-ассистента

ИИ-чат-ассистент **«DMD Ассистент»** (на основе Llama 3.3 70B через Groq API с серверным прокси) представляет собой особый случай: он **активно вызывается пользователем**, и история разговора должна сохраняться при переходах между страницами в течение текущего сеанса, чтобы пользователь не терял контекст.

3.1 Миграция: localStorage → sessionStorage

Исходная реализация хранила историю чата в `localStorage` (постоянно, до ручного удаления). Это потребовало бы согласия при строгой трактовке GDPR, поскольку содержание чата может включать персональные данные (например, симптомы, адреса, семейные обстоятельства).

Хранение было перенесено в `sessionStorage`, который **автоматически и полностью удаляется браузером при закрытии вкладки**, и таким образом подпадает под исключение **§ 25 абз. 2 № 2 TDDDG** («технически безусловно необходимо для предоставления услуги, явно запрошенной пользователем»).

3.2 Унификация языковых ключей

Изначально история чата хранилась под языко-зависимыми ключами (`dmd_chat_history_de` и `dmd_chat_history_ru`). При переключении между DE- и RU-версиями страниц история каждый раз терялась. Ключи были унифицированы в `dmd_chat_history` — теперь история сохраняется при смене языка.

3.3 Устранение проблемы потери истории при переходах

Встроенные в чат маркеры `[PAGE:...]` открывали свои цели в **новой вкладке браузера** (`target="_blank"`). Поскольку `sessionStorage` изолирован для каждой вкладки, история чата в новой вкладке была недоступна. Решение:

- Внутренние PAGE-ссылки теперь открываются в **текущей вкладке** — `sessionStorage` сохраняется
- Внешние URL-ссылки (HTTPS) по-прежнему открываются в новой вкладке с `rel="noopener noreferrer"` — стандарт безопасности

> **Правовое обоснование**
>
> Хранение истории чата в `sessionStorage` осуществляется на основании **§ 25 абз. 2 № 2 TDDDG** и **ст. 6 абз. 1 лит. f GDPR** — как технически необходимая мера для предоставления функции, активно вызванной пользователем, в рамках текущего сеанса. Дополнительное согласие не требуется.

4. Поток данных к API: серверный прокси

Центральным компонентом DSGVO-соответствия проекта является последовательное использование **собственных Vercel Serverless Functions в качестве прокси** для всех вызовов внешних API. Браузер пользователя **ни в какой момент не соединяется напрямую** с API сторонних провайдеров:

Endpoint	Источник данных	Изоляция данных
`/api/weather`	OpenWeatherMap (Великобритания)	IP пользователя не передаётся
`/api/air`	OpenWeatherMap Air Pollution (UK)	IP пользователя не передаётся
`/api/magnet`	NOAA SWPC (США)	IP пользователя не передаётся
`/api/eur-rate`	Национальный банк Украины	IP пользователя не передаётся
`/api/chat`	Groq Inc. (США, Llama 3.3 70B)	IP пользователя не передаётся

Сторонним провайдерам передаётся исключительно **IP-адрес сервера Vercel**, никогда не IP-адрес пользователя. Пользователь остаётся **анонимным** для всех внешних источников данных.

5. Верификация и результаты

Корректность реализации была проверена через **Safari Web Inspector** на iPhone в нескольких preview-деплоях Vercel.

5.1 До первого клика в баннере cookie

- **localStorage:** пусто (нет ключей проекта)
- **sessionStorage:** пусто (за исключением ключей Vercel-Preview-Toolbar, которые на production-домене не появляются)
- **Видимо в UI:** погода (16°C, Hattingen), качество воздуха (AQI 3), курс (51,46 ₴) — все данные корректно отображаются, но не сохраняются в браузере

5.2 После клика «Принять функциональные»

- `localStorage['cookieConsent']` = `"accepted"`
- `localStorage['userCity']` = `"Hattingen"` (или выбранный пользователем город)
- `sessionStorage`: `weatherData`, `aqiData`, `magnetData`, `eurRate`, `dmd_chat_history`

5.3 После клика «Только необходимые» / отзыва

- `localStorage['cookieConsent']` = `"rejected"`
- Существующие функциональные данные **автоматически удаляются**
- Интерфейс продолжает работать (данные перезагружаются при каждом переходе)

> **Результат**
>
> Сайт **теперь полностью соответствует требованиям** GDPR и TDDDG. До получения согласия от пользователя **никакие** функциональные данные не сохраняются в браузерном хранилище. Согласие — гранулярное, в любой момент отзывается, доступно через постоянно видимую иконку cookie в нижнем левом углу.

6. Сопутствующие правовые меры

Помимо чисто технической реализации, были **полностью переработаны** следующие юридические документы, чтобы они корректно отражали новую архитектуру:

- **Политика конфиденциальности (DE/RU) — версия 3:** обновлены все 23 раздела; раздел 4 переструктурирован по категориям хранения; раздел 14e уточнён (чат в sessionStorage, техническая необходимость по § 25 абз. 2 № 2 TDDDG); раздел 20 как обзорная таблица всех ключей хранения
- **Импрессум (DE/RU):** дополнен указанием об ИИ согласно DDG, оговоркой о разрешении споров (ODR-VO/VSBG), указанием источников изображений и временем ответа (24ч)
- **Тексты cookie-баннера:** все три категории (Необходимые / Функциональные / Внешние медиа) приведены в полное соответствие с реальной реализацией
- **Декларация о доступности (DE/RU):** создана добровольно; ориентируется на WCAG 2.1 AA и EN 301 549; известные барьеры открыто перечислены
- **Web Components:** `<legal-disclaimer>` для медицинских, правовых и миграционных тем; `<legal-footer>` со ссылками на Импрессум/Конфиденциальность/Доступность на каждой странице (правило двух кликов)

7. Резюме

После внедрения паттерна Storage Guard, миграции истории чата в `sessionStorage`, последовательной серверной абстракции API и полного обновления юридических документов сайт **«Digital & Mobil in Deutschland»** соответствует требованиям **GDPR**, **TDDDG** и соответствующей немецкой судебной практики.

Система **прозрачно задокументирована**, **технически верифицируема** (можно проверить в Web Inspector в любой момент) и **правовой защищаема**. Разделение между технически необходимым и требующим согласия хранением понятно для внешних проверяющих и надзорного органа.