

Alles Wichtige zum Mitnehmen.

PDFs, Tabellen, Grafiken, Gesprächsnotizen und die Skripte für Bau und Prüfung.

📅 Stand 21.09.2026 📶 Lokales Lesepaket 🛡️ Keine Produktivfreigabe

Die kompletten Exporte.



Gesamter Bericht

Alle zwölf Kapitel in einer PDF-Lesefassung. Interaktive Demos bleiben im HTML-Paket.



Alle Kapitel-PDFs

Zwölf einzelne Kapitel und das Gesamt-PDF in einem Archiv.



Daten, Grafiken & Skripte

Kuratiertes JSON/CSV, SVG-Grafiken, Gesprächsnotizen und reproduzierbare Prüfskripte.

Einzelne Kapitel als PDF.



Überblick

Statische Lesefassung · gleiche Berichtsdaten



Was funktioniert

Statische Lesefassung · gleiche Berichtsdaten



Leistung & Aufwand

Statische Lesefassung · gleiche Berichtsdaten

	Ablauf & Tour Statische Lesefassung · gleiche Berichtsdaten
	Technik & Demos Statische Lesefassung · gleiche Berichtsdaten
	One-Shot & Konzepte Statische Lesefassung · gleiche Berichtsdaten
	Wirtschaftlichkeit Statische Lesefassung · gleiche Berichtsdaten
	Markt & Branche Statische Lesefassung · gleiche Berichtsdaten
	Research & Methode Statische Lesefassung · gleiche Berichtsdaten
	Gesprächsnotizen Statische Lesefassung · gleiche Berichtsdaten
	Qualität & Quellen Statische Lesefassung · gleiche Berichtsdaten
	Downloads & Skripte Statische Lesefassung · gleiche Berichtsdaten

Tabellen und Text.

	Funktionsmatrix · CSV Alle Funktionen inklusive Stand und Beleggrenze
	Funktionsmatrix · JSON Maschinenlesbare Einträge
	Marktdaten · JSON WKO-Quellen, Jahre und Grenzen
	Wettbewerbs-Matrix · CSV 16 sichtbare Funktionen aus dem Research-Stand
	Gesprächsnotizen · TXT Redaktionelle, anonymisierte Zusammenfassungen
	Quellenregister · JSON Öffentliche Belege und private Quellen-IDs



Alle Piktogramme · SVG

Lokales Icon-System, keine externen Fonts



Paket-Manifest · JSON

Dateinamen, Größen und SHA-256

Die Skripte werden mitgeliefert.



build.py

Erzeugt die zwölf HTML-Unterseiten aus den kuratierten Daten.



graphics.py

Erzeugt die SVG-Diagramme.



icons.py

Zentrale Piktogrammdefinitionen.



privacy.py

Redaktion auf Kopien und strenge Archivprüfung.



export_pdf.py

Erzeugt Kapitel-PDFs und Gesamtbericht; kein Browserersatz.



package.py

Baut Archive ausschließlich aus einer überprüften Dateiliste.



test_static.py

Prüft Daten, Quellen, Links, Marke und Rechenmodelle.



test_browser.py

Prüft Browserinteraktionen und erzeugt Screenshots in erlaubter Umgebung.



test_privacy.py

Testet Redaktions- und Archivregeln mit synthetischen Prüfdaten.

Ohne Installation ansehen.



01

ZIP vollständig entpacken

Die Ordnerstruktur beibehalten. Die HTML-Dateien verwenden lokale Assets.



index.html öffnen

Im eigenen Browser per Doppelklick öffnen. Ein Server ist zum Lesen nicht nötig.



Gewünschtes Format laden

PDF, CSV, JSON und SVG liegen als Dateien bei. Die Website versendet nichts.



Für die technische Prüfung

```
python tools/build.py
python tools/test_static.py
python tools/test_privacy.py
python tools/export_pdf.py
python tools/test_browser.py
python tools/package.py
```

Die PDF-Erzeugung benötigt WeasyPrint und PyMuPDF. Der optionale Browserlauf benötigt Playwright/Chromium. Er verändert keine Browserrichtlinien. Ist die Navigation gesperrt, wird BLOCKED statt PASS protokolliert.

Alle Berichtsinhalte und Datei-Downloads sind ohne JavaScript lesbar. Filter, Touren und Rechner benötigen JavaScript.