

Webcraft

Handbuch

WebAssembly-Büro- und Designprogramme in napp-it CS: Menü,
Installation und Update, Erweiterungen, Beschleunigung mit cs-proxy
und Integration in cs-team

Version 1.0 · 29. Oktober 2026

Inhalt

1. Was ist Webcraft?	3
1.1 Die Bausteine	3
1.2 Wo die Dateien liegen	3
2. Das Menü Webcraft	4
2.1 Übersicht	4
3. Die Basis-Apps	5
4. Installation und Update mit About > Webcraft Downloads	6
4.1 Die Spalten	6
4.2 Installieren, aktualisieren, löschen	6
4.3 Worum sich napp-it CS kümmert	7
5. Erweiterungen mit eigenen .wasm-Apps	8
5.1 Voraussetzungen an eine App	8
5.2 Entfernen, Listendatei	8
6. Beschleunigung mit cs-proxy	9
6.1 Ohne cs-proxy	9
6.2 Prüfen	9
7. Integration in cs-team	10
7.1 Editor öffnen	10
7.2 Die Leiste über der App	10
7.3 Dokumente austauschen	12
7.4 Knopf für eine geöffnete App in Text und Calc	12
7.5 Einstellungen und Hintergrund	13
8. Browser und Fehlersuche	14
9. Nachschlagen	14

1. Was ist Webcraft?

Webcraft ist der Sammelname für eine Familie von **WebAssembly-Apps**, die vollständig im Browser laufen: eine Tabellenkalkulation (GridCraft), eine Textverarbeitung (WordCraft), ein PDF-Editor (PDFCraft) und viele weitere Werkzeuge für Grafik, Audio, Video und CAD. napp-it CS enthält diese Apps nicht selbst. Es **installiert** sie aus den GitHub-Releases ihrer Autoren, **liefert** sie effizient aus (cs-proxy) und **verknüpft** sie (Menü Webcraft, cs-team).

- **Läuft im Browser, nicht auf dem Server:** die App ist eine statische Webseite (index.html, JavaScript, eine .wasm-Datei). Der Server liefert nur Dateien. Dokumente werden mit dem Dateidialog des Browsers geöffnet und gespeichert, sie bleiben auf der **lokalen Platte** des Benutzers und werden vom Server nicht verarbeitet.
- **Keine Installation auf den Clients:** ein aktueller Browser genügt (siehe 8). Jeder Benutzer des napp-it CS Servers bekommt dieselben Apps in derselben Version.
- **Einfach im Betrieb:** eine App ist ein Ordner unter `_my/wwwroot/webcraft/<key>/`. Installieren, Aktualisieren und Löschen geschieht in der GUI (Kapitel 4), es gibt keine Datenbank und keinen Dienst.
- **Erweiterbar:** jede WebAssembly-App, die als Zip in einem GitHub-Release veröffentlicht wird, kann in die Liste aufgenommen werden (Kapitel 5).

1.1 Die Bausteine

Baustein	Aufgabe	Kapitel
Menü Webcraft	Startet die installierten Apps, jede in einem eigenen Browserfenster, und zeigt den Zustand von cs-team.	2
About > Webcraft Downloads	Apps installieren, aktualisieren, löschen und öffnen; eigene .wasm-Apps hinzufügen.	4, 5
cs-proxy (ab 0.13.0, Speicherort ab 0.13.1)	Liefert .wasm und die übrigen Dateien komprimiert, zwischengespeichert und mit dem richtigen Typ (Content-Type application/wasm) aus.	6
cs-team (ab 0.65)	Zeigt WordCraft, GridCraft und PDFCraft direkt im eigenen Fenster (Menüs Text, Calc, Files) und tauscht Dokumente mit ihnen aus.	7

1.2 Wo die Dateien liegen

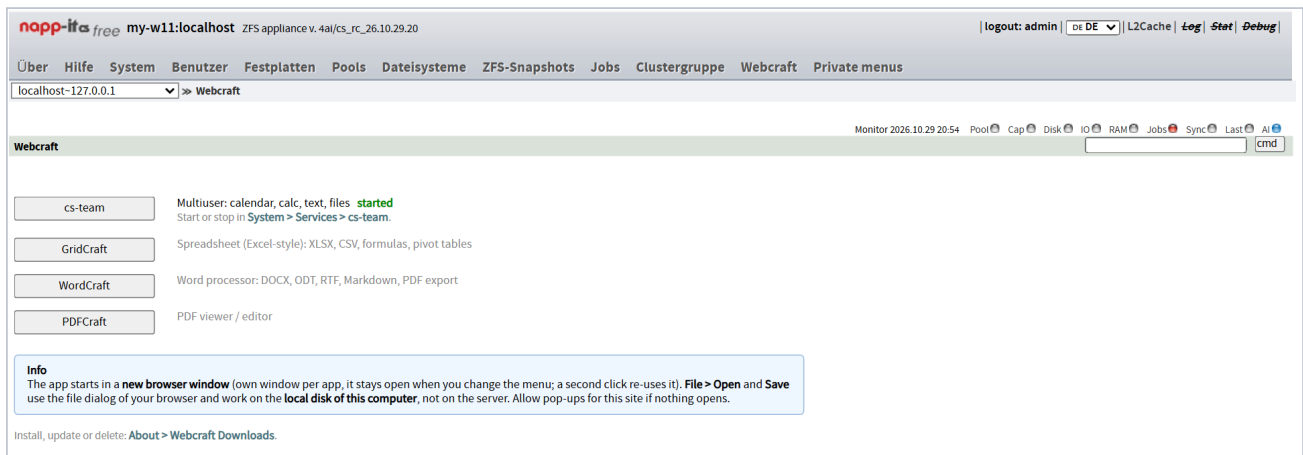
Alle installierten Apps liegen unter `<csweb-gui>/_my/wwwroot/webcraft/<key>/` (zum Beispiel `C:\opt\csweb-gui\_my\wwwroot\webcraft\wordcraft`). Der Ordner `_my` ist **nicht Teil der napp-it CS Distribution**: ein Update oder eine Neuinstallation von napp-it CS berührt die Apps nicht, und die Apps (je etwa 20 bis 50 MB) werden nicht mitgeliefert. Download, Update und Löschen (About > Webcraft Downloads), der Webserver (webserver.pl) und cs-proxy verwenden alle diesen einen Ort. `_my` zusammen mit `_cfg` sichern, wenn die installierten Apps erhalten bleiben sollen. Eine App, die eine ältere Version unter `data/wwwroot/webcraft/` installiert hat, wird weiter ausgeliefert, aber von der Seite nicht mehr verwaltet: den Ordner nach `_my/wwwroot/webcraft/` verschieben.

Nicht Teil von napp-it CS: die Apps selbst. WordCraft, GridCraft, PDFCraft und die weiteren ArtCraft-Apps stammen aus dem Projekt **storytold** auf GitHub, SolidCraft von **tangency-labs**. napp-it CS lädt sie nur auf Wunsch eines Admins aus den öffentlichen Releases. Es gelten deren Lizenzbedingungen, bitte im jeweiligen Repository nachlesen.

2. Das Menü Webcraft

Das Hauptmenü **Webcraft** (Ordner data/menues/60_Webcraft) ist der Einstieg für die Benutzer. Es listet alle installierten Apps und startet jede per Klick.

2.1 Übersicht



Menü Webcraft: cs-team mit Zustand, installierte Apps, Infokasten und Link zu den Downloads

- **cs-team** ist der erste Eintrag: ein Knopf öffnet die cs-team-Oberfläche in einem eigenen Fenster (nur wenn cs-team gestartet ist) und zeigt den Zustand **started** oder **stopped**. Die Prüfung ist dieselbe wie unter System > Services > cs-team; **Starten und Stoppen geschieht nur dort**.
- **Installierte Apps** (hier GridCraft, WordCraft und PDFCraft) folgen mit einer Kurzbeschreibung. Die Liste entsteht aus den Ordnern, die wirklich unter `_my/wwwroot/webcraft/` vorhanden sind; eine nicht installierte App erscheint nicht.
- Jede App startet in einem **eigenen Browserfenster** (Fenstername `webcraft_<app>`, kein Frame). Das Fenster bleibt beim Wechsel des Menüs offen, ein zweiter Klick nutzt es wieder. Öffnet sich nichts, bitte Pop-ups für die Seite erlauben.
- **File > Open** und **Save** in einer App verwenden den Dateidialog des Browsers und arbeiten auf der **lokalen Platte des Rechners**, auf dem der Browser läuft, nicht auf dem Server.
- Der Link unten führt zu About > Webcraft Downloads für Installieren, Aktualisieren und Löschen (Kapitel 4).

Die Adresse einer App ist `https://<host>/webcraft/<app>/`, zum Beispiel `https://mein-server/webcraft/wordcraft/`. Sie kann als Lesezeichen gesetzt oder den Benutzern direkt genannt werden.

3. Die Basis-Apps

Die Liste unter *About > Webcraft Downloads* enthält die **ArtCraft**-Apps des Projekts storytold und, als eigenes Projekt, SolidCraft. Die Tabelle zeigt den Key (Name des Ordners und der Adresse), den Zweck und die Quelle des Downloads. Welche Apps installiert werden, entscheidet der Admin; für die Büroarbeit in cs-team genügen drei (markiert).

App (Key)	Zweck	Quelle (GitHub)
GridCraft (gridcraft) *	Tabellenkalkulation im Stil von Excel: XLSX, CSV, Formeln, Pivot-Tabellen	storytold/gridcraft
WordCraft (wordcraft) *	Textverarbeitung: DOCX, ODT, RTF, Markdown, PDF-Export	storytold/wordcraft
PDFCraft (pdfcraft) *	PDF-Betrachter und -Editor	storytold/pdfcraft
DeckCraft (deckcraft)	Präsentationen im Stil von PowerPoint	storytold/deckcraft
CadCraft (cadcraft)	2D-CAD und Zeichnen (Stil AutoCAD)	storytold/cadcraft
VectorCraft (vectorcraft)	Vektorgrafik	storytold/vectorcraft
FigCraft (figcraft)	Oberflächen-Entwurf und Prototypen	storytold/figcraft
EffectCraft (effectcraft)	Bewegtgrafik und Compositing	storytold/effectcraft
SoundCraft (soundcraft)	Audio-Bearbeitung	storytold/soundcraft
PhotoCraft (photocraft)	Foto-Editor	storytold/photocraft
FilmCraft (filmcraft)	Video-Editor	storytold/filmcraft
LightCraft (lightcraft)	Zerstörungsfreier Foto-Editor	storytold/lightcraft
SolidCraft (solidcraft)	Parametrisches 3D-CAD (eigenes Projekt, nicht Teil von ArtCraft)	tangency-labs/solidcraft

* Diese drei Apps sind in cs-team integriert (Kapitel 7): WordCraft für **Text**, GridCraft für **Calc**, PDFCraft für **Files**.

Die Liste ist in napp-it CS eingebaut. Neue Releases der Apps erscheinen in der Spalte *Newest*, ohne dass napp-it CS geändert werden muss. Apps, die nicht in der Liste stehen, lassen sich als eigene Einträge hinzufügen (Kapitel 5).

4. Installation und Update mit About > Webcraft Downloads

Die Seite **About > Webcraft Downloads** (nur für Admins) listet alle Apps mit installierter und neuester Version und erledigt alles Nötige: Download, Update, Löschen, Öffnen.

The screenshot shows the napp-it web interface. At the top, there's a navigation bar with links like 'Über', 'Hilfe', 'System', etc. The main content area is titled 'Webcraft Downloads'. It features a table of installed and available webcraft apps. The table has four columns: 'Webcraft app', 'Installed', 'Newest', and 'Action'. The 'Action' column contains buttons for 'update', 'open', 'delete', and 'download'. Below the table, there's a form to 'Add another wasm app' with fields for 'Key (folder, a-z 0-9 _-)', 'GitHub repo', 'Name (optional)', and 'Asset regex (optional)'. At the bottom, there's a note about the latest GitHub release and how to verify it.

Webcraft app	Installed	Newest	Action
GridCraft Spreadsheet (Excel-style): XLSX, CSV, formulas, pivot tables storytold/gridcraft	installed, version unknown	v0.13.0	update open delete
WordCraft Word processor: DOCX, ODT, RTF, Markdown, PDF export storytold/wordcraft	installed, version unknown	v0.12.0	update open delete
DeckCraft Presentations (PowerPoint-style) storytold/deckcraft	not installed	v0.12.0	download
PDFCraft PDF viewer / editor storytold/pdfcraft	installed, version unknown	v0.15.0	update open delete
CadCraft 2D CAD and drafting (AutoCAD-style) storytold/cadcraft	not installed	v0.14.0	download
VectorCraft Vector graphics storytold/vectorcraft	not installed	v0.18.0	download
FigCraft Interface design and prototyping storytold/figcraft	not installed	v0.12.1	download
EffectCraft Motion graphics and compositing storytold/effectcraft	not installed	v0.18.0	download
SoundCraft Audio editing storytold/soundcraft	not installed	v0.14.0	download
PhotoCraft Photo editor storytold/photocraft	not installed	v0.17.0	download
FilmCraft Video editor storytold/filmcraft	not installed	v0.14.0	download
LightCraft Non-destructive photo editor storytold/lightcraft	not installed	v0.15.0	download
SolidCraft Parametric 3D CAD (separate project, not part of ArtCraft) tangency-labs/solidcraft	not installed	v0.26.1	download

Add another wasm app

Key (folder, a-z 0-9 _-):

GitHub repo:

Name (optional):

Asset regex (optional):

[add to list](#)

The latest GitHub release of the repo must contain a zip with a static web build (index.html, .js, .wasm). It is installed to `data/wwwroot/webcraft/<key>/` and opened as `https://<host>/webcraft/<key>/`. If the release publishes a SHA256SUMS file the download is verified. Newest versions are cached for 1 h (`_cfg/webcraft_versions`, GitHub allows 60 anonymous requests per hour). cs-proxy 0.13.0 or newer serves the apps compressed and cached; without it webserver.pl delivers them uncompressed. After an app update press Ctrl+F5 in the browser the old version still shows.

About > Webcraft Downloads: Apps mit installierter und neuester Version und den Aktionen download, update, open, delete

4.1 Die Spalten

Spalte	Bedeutung
Webcraft app	Name, Kurzbeschreibung und das GitHub-Repository (owner/repo) der App.
Installed	<i>not installed</i> oder die installierte Version (Tag des Releases). <i>installed, version unknown</i> heißt: der Ordner ist vorhanden, wurde aber nicht mit dieser Seite installiert (zum Beispiel von Hand kopiert).
Newest	Tag des neuesten Releases auf GitHub (eine Stunde zwischengespeichert).
Action	download (nicht installiert), update , open (öffnet die App in einem neuen Fenster) und delete .

4.2 Installieren, aktualisieren, löschen

1. **About > Webcraft Downloads** öffnen. Die Seite fragt bei GitHub das neueste Release jeder App ab (anonym, höchstens einmal pro Stunde; **refresh newest versions** erzwingt eine neue Abfrage).
2. **download** hinter einer App anklicken. napp-it CS holt das Zip des neuesten Releases, prüft es und entpackt es nach `_my/wwwroot/webcraft/<key>/`. Der installierte Tag wird in die Datei `.cs-version` in diesem Ordner geschrieben.

3. Ein neues Release zeigt in der Spalte *Newest* einen anderen Tag. **update** ersetzt die App durch die neue Version, **open** startet sie.
4. **delete** entfernt den Ordner der App. Dokumente sind davon nicht betroffen, sie liegen auf den Platten der Benutzer.
5. Nach einem Update im Browser **Strg+F5** drücken, sonst zeigt das Fenster eventuell noch die alte Version.

4.3 Worum sich napp-it CS kümmert

- **TLS wird immer geprüft:** der Download erfolgt mit geprüftem TLS. Das mitgelieferte Perl hat kein CA-Bundle, deshalb dient der Downloader des Betriebssystems (curl oder PowerShell, mit dem Vertrauensspeicher des Systems) als Rückfall. Die Zertifikatsprüfung wird nie abgeschaltet.
- **Integrität:** enthält das Release eine Datei SHA256SUMS (Prüfsummen), wird das Zip vor dem Entpacken dagegen geprüft.
- **Sicheres Entpacken:** nur Pfade innerhalb des App-Ordners werden angenommen, kein . . . , keine absoluten Pfade.
- **GitHub-Limit:** anonym sind 60 Abfragen pro Stunde erlaubt. Die neuesten Versionen werden deshalb eine Stunde lang in `_cfg/webcraft_versions` gehalten; *refresh newest versions* liest sie neu.
- **Welches Asset:** das Release muss ein Zip mit dem Namen `<key>-web- . . . .zip` enthalten (ein statischer Web-Build). Andere Dateien (Installer, Quellcode) werden ignoriert.

Ohne Internet auf dem Server: die entpackte App von Hand nach `_my/wwwroot/webcraft/<key>/` kopieren. Sie wird dann als installiert (Version unbekannt) gelistet und erscheint im Menü Webcraft; nur Update und Löschen über GitHub stehen nicht zur Verfügung.

5. Erweiterungen mit eigenen .wasm-Apps

Unter der Liste steht das Formular **Add another wasm app**. Damit lässt sich jede App aufnehmen, die als statischer Web-Build in einem GitHub-Release veröffentlicht ist, ohne napp-it CS zu ändern. Sie bekommt dann download, update, open und delete wie die Basis-Apps.

Feld	Bedeutung
Key	Name des Ordners und der Adresse, 1 bis 40 Zeichen aus a-z 0-9 _ -, zum Beispiel mycraft. Die App ist dann unter <code>https://<host>/webcraft/mycraft/</code> erreichbar. Der Key muss neu sein.
GitHub repo	owner/repo oder die vollständige Adresse <code>https://github.com/owner/repo</code> , zum Beispiel tangency-labs/solidcraft.
Name (optional)	Anzeigename in Liste und Menü. Ohne Eintrag wird der Key verwendet.
Asset regex (optional)	Regulärer Ausdruck für den Namen des Zip im Release. Vorgabe: <code>^<key>-web-.*\..zip\$</code> .

5.1 Voraussetzungen an eine App

- Ein **Release auf GitHub** mit einem **Zip** des statischen Web-Builds: `index.html`, JavaScript und die `.wasm`-Datei(en). Keine Server-Komponente, kein Build-Schritt auf dem napp-it CS Server.
- Die Seite muss alle Dateien über **relative Adressen** laden (bei trunk- und wasm-bindgen-Builds üblich), weil sie unter `/webcraft/<key>/` ausgeliefert wird.
- Der Browser muss unterstützen, was die App braucht (WebGPU oder WebGL, siehe 8). Die Apps zeichnen mit WebGPU, das einen sicheren Kontext braucht: `https://` oder `localhost`.
- `index.html` muss im Wurzelverzeichnis des Zip oder in **einem** obersten Ordner liegen (zum Beispiel `<app>-web-<version>/`); der Installer nimmt diesen Ordner als App. Ohne `index.html` wird die Installation abgelehnt.

5.2 Entfernen, Listendatei

- Eigene Einträge haben den Knopf **remove entry** (nach einer Rückfrage). Der installierte Ordner wird getrennt mit *delete* gelöscht. Die eingebauten Einträge lassen sich nicht entfernen.
- Die Liste der eigenen Einträge steht in `_cfg/webcraft/custom.conf`, eine Zeile je App: `key|name|owner/repo|asset_regex`. Sie kann von Hand bearbeitet werden und gehört zur normalen Sicherung des Ordners `_cfg`.
- Einträge der eingebauten Liste lassen sich nicht überschreiben, ein bereits vorhandener Key wird abgelehnt.

Beispiel: SolidCraft (parametrisches 3D-CAD) gehört nicht zu ArtCraft. Es steht als `tangency-labs/solidcraft` in der Liste und verwendet denselben Mechanismus wie ein eigener Eintrag. Auf dieselbe Weise lassen sich ein Klon einer App, ein angepasster Build oder ein eigenes Werkzeug den Benutzern anbieten.

6. Beschleunigung mit cs-proxy

Eine WebAssembly-App ist groß: die .wasm-Datei von GridCraft hat rund **21 MB**. Ohne Hilfe würde sie jeder Benutzer bei jedem Start unkomprimiert laden. **cs-proxy ab Version 0.13.0** (Datei `webcraft.go`) hat deshalb einen eigenen Weg für alles unter `/webcraft/`. Er sucht die Dateien zuerst in `_my/wwwroot/webcraft/` (Speicherort, Kapitel 1.2) und fällt ab Version 0.13.1 auf `data/wwwroot/webcraft/` zurück:

Funktion	Wirkung
Richtiger Typ	.wasm wird als <code>application/wasm</code> ausgeliefert. Browser brauchen das für das schnelle Streaming-Kompilieren. Jeder andere Dateityp (<code>js</code> , <code>mjs</code> , <code>json</code> , <code>css</code> , <code>svg</code> , ...) bekommt seinen passenden Content-Type.
gzip, einmal komprimiert	wasm, <code>js</code> , <code>html</code> , <code>css</code> und <code>json</code> werden einmal je Dateiversion komprimiert und im Speicher gehalten. Aus 21,4 MB von GridCraft werden 8,2 MB ; die erste Anfrage dauert etwa 0,8 s, alle weiteren kommen aus dem Cache. Voraussetzung: <code>proxy_compress</code> enthält <code>gzip</code> und der Browser sendet <code>Accept-Encoding: gzip</code> .
Eigener kleiner Cache	Nicht in <code>proxy_cache_max_mb</code> eingerechnet, nur für <code>/webcraft/</code> . Eine geänderte Datei (Größe oder Zeit) wird automatisch neu komprimiert, ein Update braucht also keinen Neustart.
ETag, 304, Range	Unveränderte Dateien werden nicht erneut gesendet (If-None-Match ergibt 304), Teile einer Datei lassen sich anfordern (Range ergibt 206).
Immutable-Caching	Dateien und Ordner mit einem 16-stelligen Hash im Namen (<code>name-32ffbcd5056039fd.js</code> , Stil <code>trunk</code> und <code>wasm-bindgen</code>) bekommen <i>immutable</i> : der Browser fragt nie wieder. Alle anderen Dateien werden mit dem ETag geprüft (<i>no-cache</i>).
Ordner-Adressen	<code>/webcraft/app</code> wird auf <code>/webcraft/app/</code> umgeleitet, dort wird <code>index.html</code> geliefert. Die Apps laden alles über relative Adressen und brauchen diesen Schrägstrich.
Sicher	Nur GET und HEAD, . . . wird abgelehnt (403), symbolische Links müssen innerhalb des Document Root bleiben, eine fehlende Datei ergibt 404.

6.1 Ohne cs-proxy

Wird der Webserver direkt verwendet (`proxy=off`), liefert auch `webserver.pl` die Apps aus (seit Version 26.10.29.45): MIME-Typen für `wasm`, `mjs`, `map` und `webmanifest`, Umleitung von Ordner-Adressen, `index.html`. Das funktioniert, aber **ohne gzip und ohne immutable-Caching**, der erste Start ist also langsamer. Ab einigen wenigen Benutzern wird `cs-proxy` empfohlen.

6.2 Prüfen

```
curl -skI -H "Accept-Encoding: gzip" https://mein-server/webcraft/gridcraft/

HTTP/1.1 200 OK          index.html wird geliefert (keine Umleitung, kein 404)

curl -skI -H "Accept-Encoding: gzip" https://mein-server/webcraft/gridcraft/<name>.wasm

Content-Type: application/wasm
Content-Encoding: gzip    mit cs-proxy; eine weitere Anfrage kommt aus dem Cache
```

Nach einer Neuinstallation von `cs-proxy` muss der laufende Prozess einmal neu gestartet werden (System > Services), sonst wird noch das alte Programm verwendet.

7. Integration in cs-team

cs-team (Kalender, Calc, Text, Files, Chat) zeigt die drei Apps **im eigenen Fenster** und tauscht Dokumente mit ihnen aus. Die Apps bleiben, was sie sind: Einzelplatz-Editoren mit den Dateiformaten von Office; cs-team bleibt das Mehrbenutzer-Werkzeug für die gemeinsamen Dokumente. Beide ergänzen sich:

cs-team Menü	Editor	Typische Verwendung
Text	WordCraft	Briefe und Berichte mit Word-Formaten (DOCX, ODT, RTF, Markdown), PDF-Export.
Calc	GridCraft	Tabellen mit XLSX und CSV, Formeln und Pivot-Tabellen.
Files	PDFCraft	PDF-Dateien ansehen und bearbeiten.

Voraussetzung: cs-team **0.65 oder neuer** (Testbuilds ab 0.64.1-test3) und die Apps installiert wie in Kapitel 4. cs-team und die Apps laufen auf demselben Host.

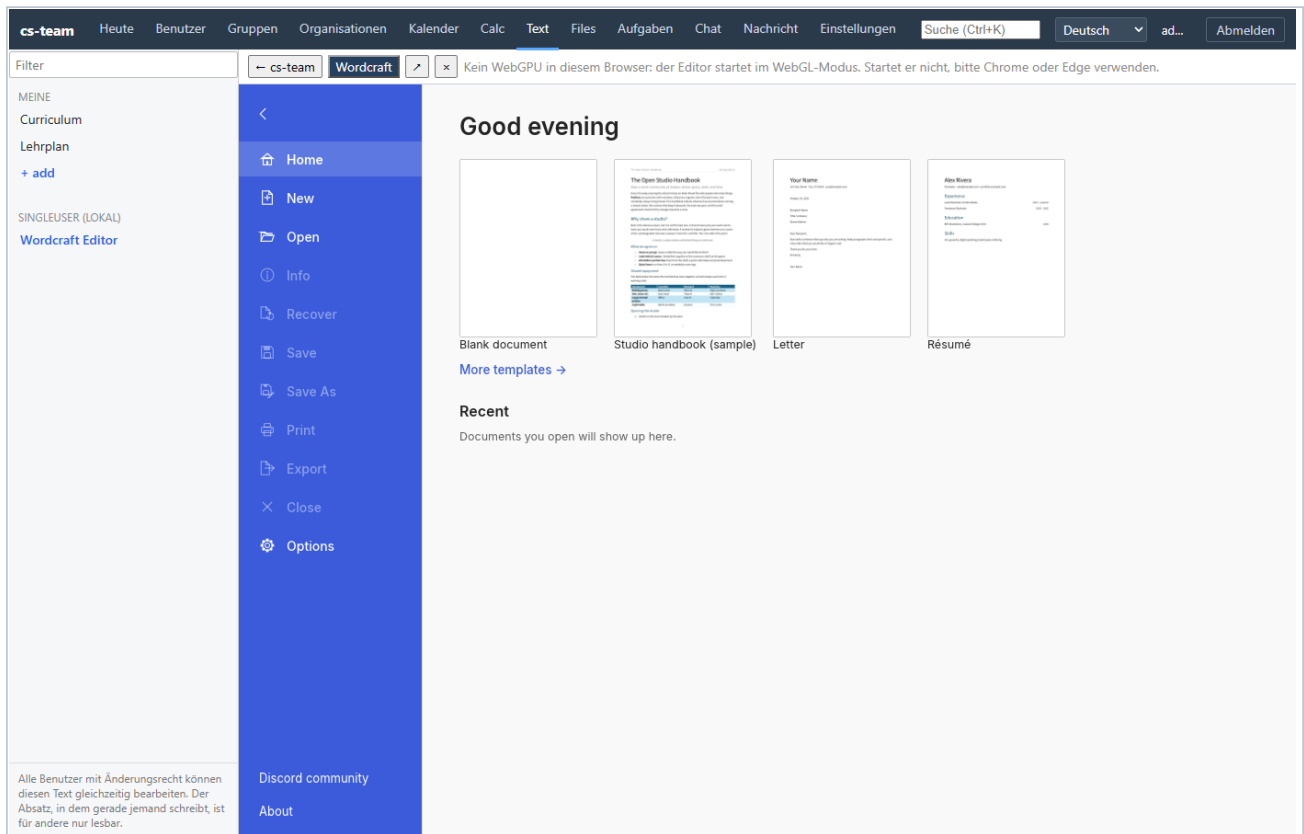
7.1 Editor öffnen

In den Menüs Text, Calc oder Files zeigt die linke Spalte unter **+ add** den Abschnitt **Singleuser (lokal)** mit dem Eintrag des passenden Editors. Ein Klick öffnet die App **im rechten Teil des cs-team-Fensters**.

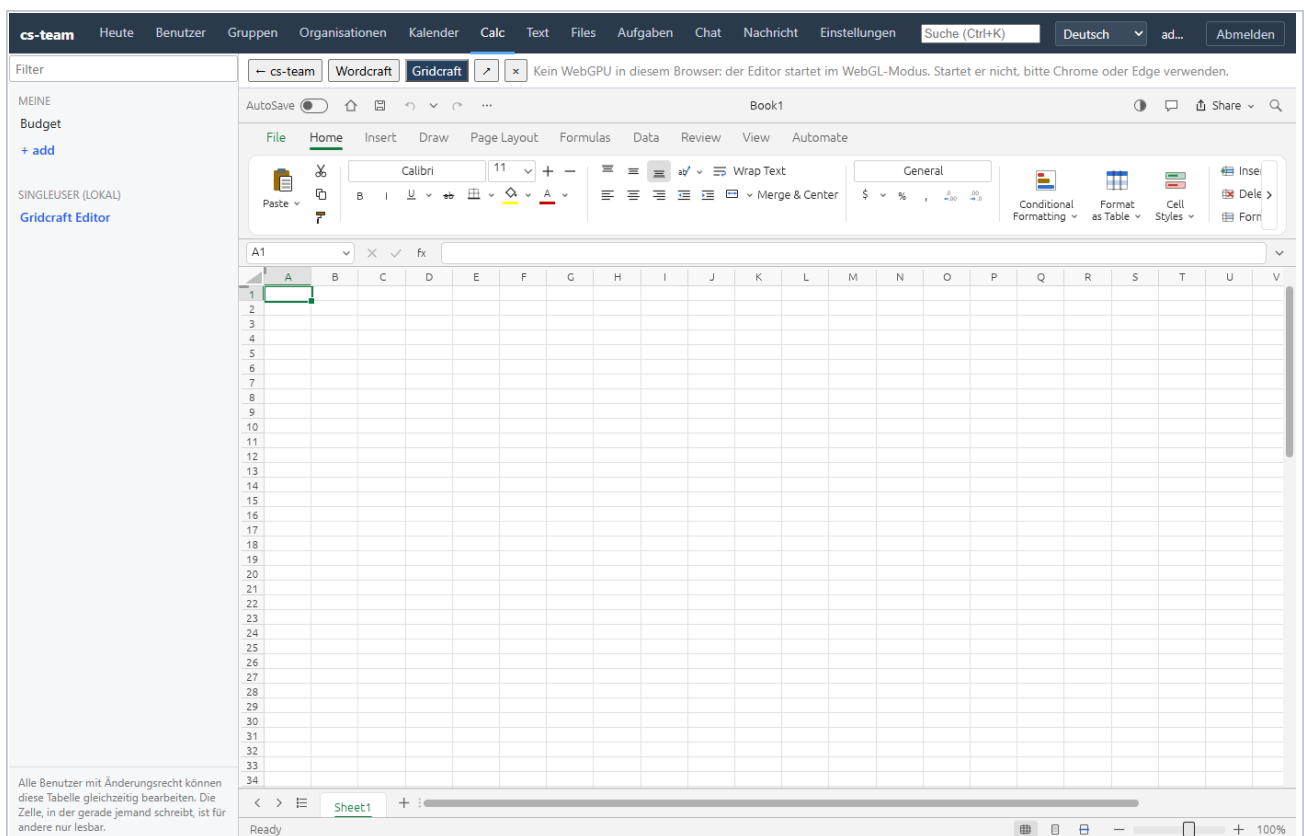
Menü Text: unter + add der Abschnitt Singleuser (lokal) mit Wordcraft Editor

7.2 Die Leiste über der App

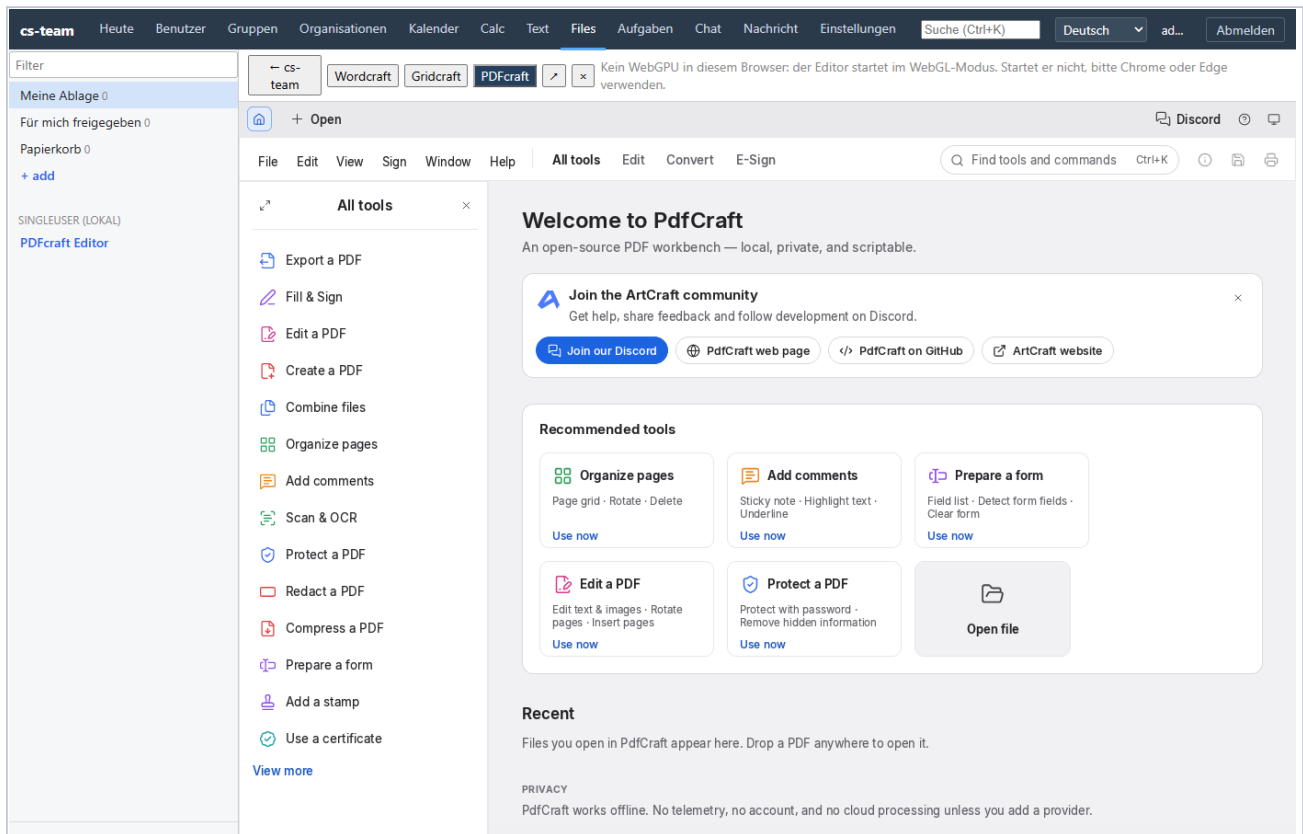
- **← cs-team** geht zurück ins cs-team-Menü. Die Apps bleiben mit ihrem Inhalt im Hintergrund geöffnet.
- Ein Knopf je **geöffneter App** schaltet zwischen WordCraft, GridCraft und PDFCraft um; die aktive ist hervorgehoben.
- **↗** öffnet die aktive App in einem eigenen Browserfenster (für einen großen Bildschirm oder einen zweiten Monitor).
- **×** schließt die App. Änderungen, die nicht mit *File > Save* gespeichert wurden, gehen verloren.



WordCraft im cs-team-Fenster: Leiste mit Zurück-Knopf, App-Knöpfen, neuem Fenster und Schließen



GridCraft im Menü Calc



PDFCraft im Menü Files

7.3 Dokumente austauschen

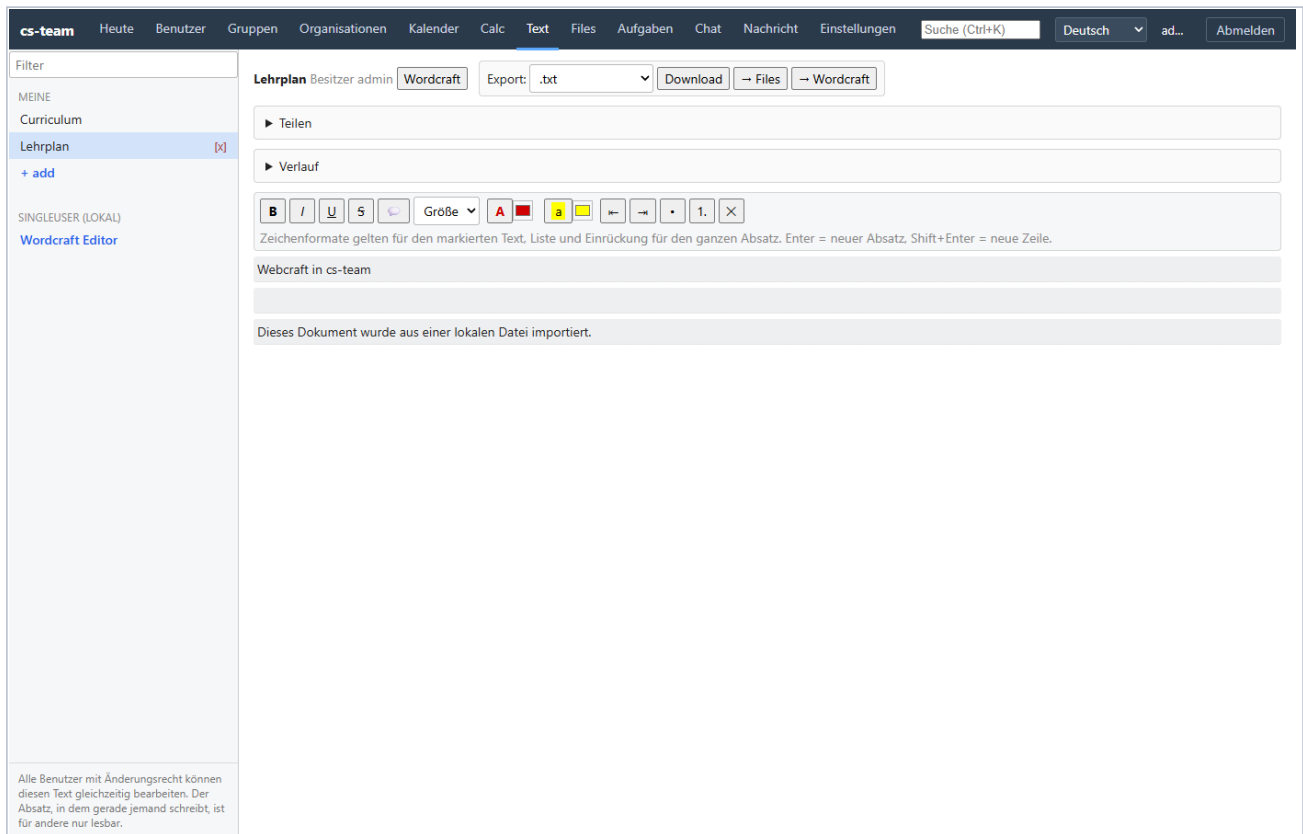
Es gibt drei Wege, vom einfachen zum komfortablen:

1. **Zwischenablage:** in einem geöffneten Text- oder Calc-Dokument kopiert der Knopf → **Wordcraft** bzw. → **Gridcraft** (neben Download) den Inhalt (Text als reiner Text, Calc als Zellwerte mit Tabulator getrennt, ohne Formeln und Formate) in die Zwischenablage und schaltet auf die App um. Dort mit **Strg+V** einfügen.
2. **Datei, von der App nach cs-team:** in der App **File > Save** (DOCX, TXT, XLSX oder CSV) auf die lokale Platte, dann in cs-team + **add > Aus lokaler Datei importieren:** Datei wählen, der Name wird aus dem Dateinamen übernommen und kann geändert werden, Freigabe für die Gruppe wie bei jedem neuen Dokument. Formate: Text: txt, docx, ctext; Calc: csv, xlsx, cscalc.
3. **Datei, von cs-team in die App:** das Dokument in cs-team herunterladen (Export), dann in der App **File > Open**.

cs-team kann die Apps **nicht** steuern: sie bieten keine Schnittstelle zum Host (nur die Parameter `?webgl` und `?sample`). Der Austausch läuft deshalb immer über die Zwischenablage oder über eine Datei auf der lokalen Platte.

7.4 Knopf für eine geöffnete App in Text und Calc

Ist WordCraft (oder GridCraft) bereits geöffnet und Sie arbeiten in einem Text- (oder Calc-) Dokument von cs-team, zeigt die Kopfzeile des Dokuments einen **Knopf mit dem Namen der App** neben dem Namen des Dokuments (oberhalb von Text bzw. Tabelle). Ein Klick schaltet auf den geöffneten Editor zurück, **ohne zu kopieren**. Der Knopf erscheint nur, solange die App geöffnet ist; → **Wordcraft** (kopieren und umschalten) bleibt weiter verfügbar.



Geöffnetes Dokument in Text: der Knopf Wordcraft in der Kopfzeile schaltet auf den offenen Editor um

7.5 Einstellungen und Hintergrund

- **Adresse:** cs-team lädt die App von `https://<Host der Seite>/webcraft/<app>/`. Antwortet napp-it CS unter einer anderen Adresse (anderer Port), einmal in der Konsole des Browsers eintragen: `localStorage.setItem("cs.webcraft.base", "https://host:8443/webcraft/")`.
- **Sicherheit:** die Content Security Policy von cs-team erlaubt Frames nur von der Seite selbst und vom selben Host (https, beliebiger Port). Andere Adressen werden nicht eingebettet.
- **Import aus lokaler Datei** in der REST-Schnittstelle: POST `/api/docs/importfile?file=<name.ext>&name=<optional>`, der Body ist der Inhalt der Datei; gleiche Prüfungen und Formate wie der Import aus Files.
- **Start von cs-team:** im Menü Webcraft (Zustand) und unter System > Services > cs-team (starten, stoppen).

8. Browser und Fehlersuche

Die Apps zeichnen ihre Oberfläche mit **WebGPU**. Chrome und Edge (aktuelle Versionen) haben es; andere Browser unter Umständen nicht (ältere Firefox-Versionen, je nach System und Grafiktreiber). Die Seite muss mit `https://` oder von `localhost` aus geöffnet werden (sicherer Kontext).

Symptom	Ursache und Abhilfe
Die App bleibt bei <i>Loading ...</i>	Der Browser hat kein WebGPU. In cs-team wird die App automatisch im WebGL-Modus gestartet (Parameter <code>?webgl</code>) und die Leiste zeigt einen Hinweis. In einem eigenen Fenster <code>?webgl</code> an die Adresse hängen oder Chrome bzw. Edge verwenden.
Beim Klick öffnet sich nichts	Der Browser blockiert Pop-ups der Seite. Erlauben (das Menü Webcraft öffnet ein Fenster je App).
404 oder leere Seite	Die App ist nicht installiert (About > Webcraft Downloads) oder die Adresse hat keinen Schrägstrich am Ende. Ohne cs-proxy 0.13.0 muss der Webserver eine Version mit wasm-Unterstützung sein (26.10.29.45).
Sehr langsamer erster Start	Ohne cs-proxy werden die 20 MB unkomprimiert gesendet. cs-proxy und <code>proxy_compress = gzip</code> aktivieren (Kapitel 6).
Nach einem Update erscheint die alte App	Strg+F5 drücken (Browser-Cache).
Download bricht mit GitHub-Fehler ab	Anonymes Limit von 60 Abfragen pro Stunde oder kein Internet. Warten oder die App von Hand installieren (Kapitel 4.3).
Frame in cs-team bleibt leer	Adresse weicht ab (anderer Port): <code>cs.webcraft.base</code> setzen (Kapitel 7.5). cs-team und napp-it CS müssen unter demselben Hostnamen erreichbar sein.
Meine Dokumente sind nach dem Schließen weg	Dokumente werden nicht auf dem Server gespeichert. Mit <i>File > Save</i> auf die lokale Platte speichern, danach in cs-team importieren, wenn sie geteilt werden sollen.

9. Nachschlagen

Was	Ort / Wert
Installierte Apps (Speicherort)	<code>_my/wwwroot/webcraft/<key>/</code> (nicht in der Distribution), Version in <code>.cs-version</code>
Adresse einer App	<code>https://<host>/webcraft/<key>/</code>
Eigene Einträge	<code>_cfg/webcraft/custom.conf</code> (key name owner repo asset_regex)
Cache der Versionen	<code>_cfg/webcraft_versions</code> (1 Stunde)
Menü Webcraft	<code>data/menues/60_Webcraft/action.pl</code>
Download-Seite	<code>data/menues/_lib/tools/Webcraft_Downloads/action.pl</code> , Logik in <code>data/menues/_lib/windows/webcraftlib.pl</code>
Auslieferung	cs-proxy 0.13.0 <code>webcraft.go</code> ; Rückfall <code>webserver.pl</code> (26.10.29.45)
cs-team	0.65 oder neuer: <code>web/app.js</code> (Editoren in Text, Calc, Files), Route <code>/api/docs/importfile</code>
Weitere Dokumentation	napp-it CS Erste Schritte, cs-proxy Handbuch, cs-team Handbuch (napp-it.org > Downloads)