

RECADA / Airholder — vollständige Projektbeschreibung (v50)

Idee · Architektur · Dateien und Formate · Konstruktionsprozess · Konfigurator (Unity) · Prototyping · Prüfungen ISO 11154 · Zubehör · Rekrutierung in verschiedenen Ländern · Lebenszyklus · ISO 9001 · Schutz und VDI · Infrastruktur und Serverwahl · Reverse-Werkzeug (recada.py) · Produkteditor · Automatisierung · Finanzen · Management · Risiken · Roadmap — Revision 50 · 9. September 2026 · deutsche Übersetzung, Teil 1 von 5

Legende: Begriffe und Abkürzungen

Kurzes Wörterbuch zum Lesen des Dokuments. Jeder Begriff ist in einer Zeile erklärt — Details in den jeweiligen Abschnitten.

Begriff	Was es in einfachen Worten ist
JSON	Textformat «Feld: Wert» — die Parametersprache des Systems (params.json, Pässe); mit dem Editor zu bearbeiten
YAML	«JSON für Menschen» — gleicher Inhalt, lesbar als eingerückte Liste; bei uns die Sprache der Montageanleitungen
spec.yaml / SDD	Aufgabenspezifikation vor der Geometrie: wozu das Teil, womit es koppelt, prüfbare Abnahmebedingungen, was nicht gemacht wird; die CI baut daraus den Abnahmelauf; Basis der Stückvergütung (Spec-Driven Development)
Python / build123d	Programmiersprache / Bibliothek, in der die Generatoren geschrieben sind — Code, der Geometrie aus Parametern baut
Generator (generator.py)	Skript, das params.json und den Pass liest und das Teil baut; liest sich wie ein Arbeitsplan
Pass (vehicle/accessory passport)	JSON mit Befestigungspunkten, Winkeln und Flächen des Fahrzeugs oder Zubehörs — alles, was über ein fremdes Objekt zu wissen ist
Skelett / MasterParams	Gerüst aus Punkten und Achsen mit Hauptparametern (Höhe, Winkel) — Brücke zwischen Fahrzeug und Produkt
Git	Versionsverwaltung: vollständige Historie «wer, was, wann und warum geändert hat»; Grundlage der gesamten Arbeit
Gitea	Eigener Webserver für Git (wie GitHub, aber auf eigener Hardware): Repositories, Rechte, Review, Kanban
Repository (repo)	Projektordner unter Git-Verwaltung mit gesamter Historie
Commit	Eine gespeicherte Änderung mit obligatorischer Begründung («warum») in der Nachricht
Diff	Anschaulicher Unterschied zweier Dateiversionen: was hinzugefügt, was entfernt, Zeile für Zeile; «lesbarer Diff» = Änderung mit bloßem Auge sichtbar (height: 60 → 80), deshalb wird die Wahrheit als Text gespeichert
Branch (Zweig)	Parallele Arbeitslinie für Experimente; main — Hauptlinie, geschützt
PR (Pull Request)	Antrag «meine Änderungen in main übernehmen» — Punkt für Review und Unterschrift; direktes Schreiben in main ist verboten
Tag (git-tag) / Release	Benannte Versionsmarke; Release eines Teils = signierter Tag, nur daraus wird produziert
LFS (Large File Storage)	Git-Erweiterung für schwere Dateien: in der Historie ein leichter Zeiger, der Inhalt im Speicher daneben; für Scans, STEP, Berichte
zippey	Filter, der Git den Inhalt von .FCStd (FreeCAD) lesbar macht und sinnvolle Diffs zeigt
CI (Continuous Integration)	Server-Pipeline: baut bei Commit/Tag selbst die Geometrie, führt Prüfungen aus und legt Artefakte ab
Artefakt	Von der CI aus der Wahrheit gebaute Datei (STEP, GLB, PDF, BOM): wird nicht bearbeitet — sondern neu gebaut

Begriff	Was es in einfachen Worten ist
Validator / DFM	Automatische Prüfung «ist es fertigbar»: Biegeregeln von Blechcon, Abstände, Verpackung, Befestigung; rot = Stopp
B-Rep / STEP-Topologie	Speicherform exakter Geometrie: Körper → Schalen → Flächen → Kanten → Ecken; daraus werden bei der Vermessung die Maße gewonnen
AAG (Flächenadjazenzgraph)	Graph «Fläche → Nachbarfläche»: das Abtrennen der Stirnflächen spaltet das Teil in obere und untere Schale — Grundlage der automatischen Biegungserkennung
Teileskelett / Medial Axis	Mittlere Darstellung ohne Dicke: Platten + Biegelinien + Dicke als ein Parameter; Ergebnis der Vermessung und Basis der Parametrisierung
FTG (Biegeübergangsgraph)	Verdichtung des AAG: Biegung = Graphkante, Flansche = Knoten; vereint koaxiale, durch Freischnitte unterbrochene Biegungen zu einem technologischen Übergang
BA / BD (Zugabe und Abzug)	Formeln der Abwicklungslänge über K-Faktor und neutrale Faser (ANSI/DIN) — analytisch, ohne CAD
Kerf (Schnittbreite)	Vom Laser/Plasma verbrauchtes Metall; wird im Zuschnitt berücksichtigt, sonst ist das Teil kleiner als nominal
Freischnitt (Entlastungsschnitt / relief)	Ausschnitt am Biegerand, damit das Metall beim Biegen nicht reißt; im System nach Regel aus t/R berechnet, nicht vom Original kopiert
Rückfederung (springback)	Metall richtet sich nach dem Biegen teilweise auf; bei Edelstahl deutlich — der Winkel an der Presse wird mit Reserve gesetzt
FEM / CalculiX / Gmsh	Festigkeits- und Eigenfrequenzberechnung (Modalanalyse); Gmsh baut das Netz, CalculiX rechnet — kostenlos
CFD / OpenFOAM	Strömungsberechnung (Aerodynamik); schwer, einmalig außerhalb der Pipeline
STEP / DXF	Austauschformate exakter Geometrie: STEP — 3D für Blechcon und vom OEM, DXF — flache Abwicklungen und Schnitte
STL / GLB	Dreiecksnetze ohne exakte Maße: STL — Prototypendruck, GLB — Anzeige im Konfigurator
.FCStd	FreeCAD-Datei; bei uns Doppelstatus: handgebaute Skelette = Wahrheit, CI-Ansichtsbaugruppen = Artefakt
BOM (Bill of Materials)	Stückliste: woraus das Produkt besteht, Massen, Preise — von der CI erzeugt
catalog.json	Vertrag zwischen CI und Konfigurator: Teileliste mit Status, GLB-Verweisen und Regeln
SKU / Rev (Revision)	Artikelnummer / Konstruktionsversion innerhalb des Artikels (Rev A, B ...)
FFF (Form-Fit-Function)	Regel: Passung und Funktion unverändert → neue Revision derselben SKU; verändert → neue SKU
VDI / Kasm	FreeCAD im Browser: das Programm läuft auf unserem Server, zum Menschen fliegen nur Pixel; Dateien verlassen den Server nicht
Container / Docker	Isolierte «Box» mit einer Anwendung; VDI-Sitzung = Einweg-Container
WireGuard / VPN	Verschlüsselter Tunnel in unser Netz; Schlüssel = Zugang, Schlüsselentzug = sofortige Abschaltung
2FA / TOTP	Zweiter Login-Faktor: Code aus der Telefon-App
Webhook	Automatisches Signal eines Systems an ein anderes («Bestand = 0») — Auslöser für CI-Skripte
LLM / Claude / MCP	Sprachmodell als Assistent; MCP — Protokoll, über das Claude FreeCAD und Werkzeuge steuert
Prompt	Aufgabe an das Modell in normaler Sprache («Plattform um 40 mm anheben»); ändert Parameter, nicht Geometrie direkt
GDPR / GeschGehG / StVZO / ISO 11154	Regulatorik: EU-Datenschutz / deutsches Geschäftsgeheimnisgesetz / Zulassungsregeln / Prüfnorm für Dachträger
CM1 / CM2	Produktmarge / Marge inklusive Entwicklungskosten — Ebenen des Finanzmodells
CPQ	Configure-Price-Quote: Händlerwerkzeug «Konfiguration → Preis → Angebot → Fertigungsauftrag»
OCCT / OpenCASCADE	Offener Geometriekern (B-Rep, Boolesche Operationen, Verrundungen, STEP); derselbe Kern wie in FreeCAD; bei uns direkt aufgerufen

Begriff	Was es in einfachen Worten ist
OCF	Python-Bindung von OCCT aus dem Paket cadquery-ocp; darauf ist recada.py geschrieben; Kernversion 7.9.3
recada.py	Einzeldatei des Reverse-Werkzeugs: Vermessung, Schalen, Kinematik, Strecken, Operationen, Abnahme; CLI und Browseroberfläche
headless	Arbeit ohne grafischen Bildschirm — auf dem Server, in der CI, im Stapel
Streamlit	Python-Bibliothek für Browseroberflächen; das Arbeitsfenster des Werkzeugs
three.js	3D-Grafikbibliothek im Browser; unser Baugruppen-Viewer
HLR (Hidden Line Removal)	OCCT-Modul: Projektionen mit sichtbaren und verdeckten Kanten — Strichansichten für Zeichnungen und Anleitungen, Vektor
XCAF	OCCT-Schicht für Baugruppenstruktur: Teilennamen, Baum, Positionen, Farben in STEP
BREP (Format)	Internes Kernformat; Version ändert sich mit dem Release; intern ja, an Grenzen nur STEP
Drei Abnahmekriterien	Metrologie (Genauigkeit), Geschlossenheit (freie Kanten = 0, Solid ≥ 1), Erkennung durch das Lieferantenportal; weiter: Sitz nach Scan, Wareneingang, Feld
Metrologie (point-sampling)	Abstand von Punkten des Referenzteils zur Oberfläche des Kandidaten, beidseitige Stichprobe, Schwelle 99,9 % in 0,05 mm
Freie Kante	Kante mit nur einer Fläche — Loch in der Schale; null davon = Teil geschlossen
Virtueller Taster	Punkt – Normale $\times t$ trifft die Oberfläche $\rightarrow$ Schalenfläche; Methode zur Aufteilung des Blechs in zwei Seiten
Origami-Kinematik	Starre Platten an Gelenken entlang der Biegeachsen; Winkeländerung = Drehung des Teilbaums
K-Faktor	Lage der neutralen Faser in der Biegung (0,33–0,5); bestimmt die Zugabe BA; kalibriert über Portalantworten
Portal Blechcon	Webportal des Schneid- und Biegelieferanten: nimmt STEP an, berechnet die Abwicklung; drittes Abnahmekriterium
product.json / project.json	Wahrheit von Produkt und Projekt: Parameter und Operationsjournal; Geometrie ist Artefakt daraus
Aufbaurichtlinien	Herstellerleitfäden für Aufbauhersteller (VW, Ford BEMM, Stellantis): Lasten, Zonen, Verbote, Befestigung — erste Quelle für Grenzen
OEM-CAD / JT / CATIA	Werksgeometrie der Karosserie; JT und CATIA werden einmal in STEP konvertiert (Inventor öffnet JT)
FBX	Format visueller Modelle (Blender, Unity); Fahrzeug als Netz mit ungefährem Maßstab
ICP / Open3D / trimesh	Registrierung des Scans ins Koordinatensystem (ICP), Bibliotheken für Netze
WebRTC / Viewport-Streaming	Übertragung von Bildern in den Browser; «nach außen nur Pixel» — Geometrie verlässt den Server nicht
WASM / OCP.wasm	Kern im Browser: OCCT und OCP nach WebAssembly kompiliert; Werkzeug beim Kunden ohne Server
Zoo / KCL	zoo.dev — CAD mit Code als Wahrheit (Sprache KCL), Cloud-Engine in den USA; das philosophisch nächste Projekt
Nutenstein / T-Nut	Nutenstein in der Aluminiumprofilnut; Befestigungsschnittstelle für Zubehör und Halter
CAPA	Corrective and Preventive Action — Korrekturzyklus nach ISO 9001; bei uns: Reklamation $\rightarrow$ Validatorregel
Strouhal (St)	Dimensionslose Wirbelablösefrequenz; Berechnung des Brummens an Querträgern per Formel ohne CFD
DIN 33402 / ISO 2768	Anthropometrie (Erreichbarkeit) / Allgmeintoleranzen auf Zeichnungen
EX44 / Storage Box / restic	Hetzner-Server (i5-13500, 64 GB, FSN1) / Backup-Speicher (Helsinki) / Backup-Programm
Mikroversion	Unveränderlicher Änderungsdatensatz mit Verweis auf den Vater (Onshape); bei uns — Journaloperation mit Hash des Vaters
Definition / Ergebnis der Neuberechnung	Was gespeichert wird (Operationsliste) / was gecacht und stets neu gebaut wird (B-Rep, Netz); Wahrheit ist die Definition

Begriff	Was es in einfachen Worten ist
Robuste Änderung	Operation adressiert ein Element über ein dauerhaftes Merkmal, nicht über die Position — anwendbar auf eine andere Version des Originals
Wiedereinspielen des Journals	Anwendung aller Operationen auf eine neue Revision des Ausgangsteils; Ranger-Halter v2 erhält dieselben Korrekturen
Geometrischer Fingerabdruck	Adressierung einer Fläche oder Bohrung über Typ, Normale, Zentrum, Fläche — Ersatz für dauerhafte IDs, die OCCT nicht hat
Topological Naming Problem	Flächennummern ändern sich beim Neuaufbau; bekanntes FreeCAD-Leiden; bei uns durch Fingerabdruck für Blechteile gelöst
Zweig / Zusammenführen des Journals	Produktvarianten als Zweige der Operationsliste; Zusammenführen ohne Konflikte — «Quelle gewinnt» (Onshape-Regel)
Geometrischer Diff	Vergleich zweier Versionen: Flächen hinzugefügt, entfernt, verschoben — als Hervorhebung im Viewer
FeatureScript	Sprache eigener Operationen bei Onshape; Bestätigung, dass «Operation als Code mit Parametern» die richtige Abstraktion ist
6060 / 6063 / 6082	Aluminiumlegierungen für Strangpressprofile; 6082 um ein Drittel fester, aber schwerer zu pressen
1.4301 / 1.4404 / 1.4571	Edelstähle (AISI 304 / 316L / 316Ti); 316-Serie mit Molybdän chloridbeständig — Salz, Küste
A2 / A4	Klassen von Edelstahlschrauben (304 / 316)
EPDM / NBR	Gummis: EPDM UV- und ozonbeständig — für Unterlagen; NBR reißt nach einer Saison
ASA / ABS / PA6 GF / PA12	Kunststoffe: ASA hält UV (ABS nicht); PA6 glasfaserverstärkt für belastete Teile; PA12 für 3D-Druck SLS/MJF
Pulver Polyester / Epoxid	Polyesterpulver hält UV; Epoxid kreidet nach einer Saison — nur innen
Galvanisches Paar	Korrosion von Aluminium im Kontakt mit Edelstahl in nasser Umgebung; Schutz — Isolator, A2/A4-Schrauben mit Scheiben, Beschichtung, Drainage

Revisionshistorie

Rev.	Datum	Was hinzugefügt wurde
v25	31.08.2026	Werkstattpraxis, Baugruppenprüfungen, Klassen «physikalischer» und «Gamedev»-Ideen geschlossen
v26	03.09.2026	15a Reverse-Werkzeug recada.py; 15b Produkteditor; drei Abnahmekriterien; drei behobene Fehler
v27	03.09.2026	15 neu geschrieben: Rolle von FreeCAD und Ingenieur, Workbench-Idee; 15c Bibliothek und Server; Risiken und Roadmap nach Praxis; Zoo
v28	03.09.2026	Wie Zoo streamt; eigene Engine: Kern nein, Umgebung ja
v29	03.09.2026	Kern direkt ohne FreeCAD; mit KI erstellt, durch Realität geprüft; Viewer auf OCCT
v30	03.09.2026	Entscheidung: Cascade-Viewer auf dem Server, nach außen nur Pixel; EX44 ausreichend; Datenschutz
v31	03.09.2026	Kernversionen 7.8/7.9/8.0; STEP an Grenzen; STEP mit Namen als Pflicht
v32	03.09.2026	15d Montageanleitung als Artefakt: Analyse Defender 110, Werkzeuge, vier Schichten
v33	03.09.2026	15e Zwei Welten: B-Rep und Netz; FBX, GLB aus OCCT, Blender
v34	03.09.2026	15f Daten: sechs Quellen, zwei Kataloge, einheitliche Struktur, Werkskoordinatensystem, viertes Kriterium
v35	03.09.2026	Option: Reverse aus STL; Wareneingangsprüfung als fünftes Kriterium
v36	04.09.2026	15g Wettbewerber als Datenquelle; Revisionshistorie
v37	04.09.2026	15h Prüfungen auf Produktebene: Platzierung, Erreichbarkeit, Höhe, Einstellung
v38	04.09.2026	15i Produktdesign: was das System liefert — Käufer, Konstrukteur, Betrieb, Marke
v39	04.09.2026	15k Rückkopplung aus dem Feld: Reklamationen, Kundenerfahrung, soziale Medien; sechstes Kriterium
v40	04.09.2026	Aufbaurichtlinien als erste Quelle für Grenzen; Verweise in Validatorregeln
v41	04.09.2026	Legende um Begriffe v26–v40 ergänzt (26 Begriffe)
v42	04.09.2026	15l Versionsmodell: Lehren aus Onshape — Mikroversionen, robuste Änderungen, Wiedereinspielen, konfliktfreie Zweige; Legende
v43	06.09.2026	15m Werkstoffe und Technologien; Zeiterfassung über das Journal
v44	06.09.2026	15n Externe Ressourcen: OCCT 8.0.1, YouCanNotUnfold, chinesische und russische Kerne
v45	07.09.2026	15o Geschwindigkeit: neun Ideen; 15p Remote-Desktops; 15q Rekrutierung; 15r Rechte und Geschäftsgeheimnis
v46	07.09.2026	15s Monetarisierung: Zubehör-Ökosystem; Konstruieren im System auf Fahrzeugdaten
v47	07.09.2026	15s Modell C: Zubehörhersteller und Selbermacher; Verantwortungsklassen
v48	07.09.2026	15s Modell C: Händler und Installateure — B2B am Tresen
v49	07.09.2026	15t Programm «Airholder Ready»
v50	09.09.2026	15u Fluss statt Codieren: Messmethode am Ranger; vier Mechanismen gegen Iterationen; Entscheidungen vor der ersten Linie

1. Idee und Ausgangslage

RECADA (Augsburg) fertigt Dachträger Airholder: Plattform aus stranggepressten Aluminiumprofilen (Profile am Lager, Querschnitte einmal gezeichnet) + gebogene Beine aus Edelstahl 1.4301 (2–3 mm); Schneiden und Biegen extern (Blechcon.de). Engpass: Beine je Fahrzeug und Zubehör. Ziel: ein System, das am Eingang das Dach (STL-Scan mit Befestigungspunkten oder STEP vom OEM) erhält und am Ausgang Fertigungsdateien, aktualisierten Konfigurator und Anleitungen liefert; das Design wird per Prompt gesteuert; Teamarbeit mit Versionierung; Wissen sammelt sich an; Ausführende sind Remote-Ingenieure aus verschiedenen Ländern, ausgewählt nach Ergebnissen von Testaufgaben.

2. Architekturprinzip und Schichten

Quelle der Wahrheit sind Parameter und Code, nicht Geometrie. FreeCAD-/STEP-Dateien sind Build-Artefakte. Das ermöglicht Prompt-Steuerung (das LLM ändert Zahlen im JSON, bewegt keine Flächen), lesbare Git-Diffs, Teamarbeit ohne Binärkonflikte, automatischen Neuaufbau der Ableitungen.

Schicht	Inhalt
1. Fahrzeugpass	JSON aus STL/STEP (einmalig): Befestigungspunkte mit Normalen, Winkel, Achsabstände; für Teile, die dem Dach folgen (Windschild), Stützflächen als Spline-Approximation (nicht das rohe Netz); Genauigkeitsfeld geometry_source. Roher Scan ins Archiv, Rolle: Kontrollreferenz für Abstandsprüfungen.
2. Skelett + MasterParams	Punkte, Achsen, Ebenen, Hauptparameter (H, alpha). Alles «Feste» liest seine Maße von hier. Das Skelett ist die Brücke zwischen Fahrzeugpass und Produktpass. Eiserne Regel der Skelettmethode: Teile verweisen nie aufeinander — nur auf das Skelett (sonst reißen Verknüpfungen unvorhersehbar); Abhängigkeiten strikt von oben nach unten: MasterParams → Skeleton → Teile.
3. Profilkatalog	Querschnitte einmalig (DXF-Import); Parameter — Schnittlänge; «Breite aus Länge» als Regel.
4. Generatoren (build123d)	Code baut Teile vom Skelett; Werkstoff, R3, Beschichtung fixiert; Bausteinbibliothek (bent_flange, hole_array, bolt_joint) verbirgt die Komplexität.
5. Validator DFM+	Regeln von Blechcon (Abkantungen, Radien, Abwicklung, Kollisionen) + Abstand zum Dach ≥ 20 mm (statische Reserve statt Karosserieverformungsrechnung) + Verpackung nach Limits des Versandkanals (aktuell GLS: ≤ 40 kg/Paket als Hauptlimit, Masse je Paket des Sets mit Aufteilung in der BOM; Länge praktisch bis ~ 2 m, Zuschlagsschwellen aus dem GLS-Tarif in rules/) + Befestigung (Schrauben-/Muttertabelle, Eingriffsprüfung). Rot — Artefakte werden nicht gebaut. Physikalische Kalibrierung der Biegeeregeln (FEA-in-the-loop): CalculiX simuliert den Biegevorgang und zeigt, bei welchem Radius die Spannung an der Außenfaser bei 1.4301 Mikrorisse droht; einmal gerechnet für das Raster «Dicke $\times$ Radius $\times$ Güte» → Tabelle zulässiger R in dfm_blechcon.json → der Validator schlägt sofort nach. So wird das tabellarische «R $\geq 1 \times t$ » durch Rechnung und Lieferantenpraxis bestätigt. Prüfung gegen die reale Werkzeugausstattung (CAM-in-the-Loop in praktischer Form): von Blechcon wird die Liste verfügbarer Stempel und Matrizen erfragt (Radien, V-Nut-Breite, maximale Biegelänge) und als Tabelle in dfm_blechcon.json geführt — der Validator antwortet «R2,5 nein, nächster R3, K-Faktor neu» und prüft die minimale Flanschlänge, damit das Teil nicht in die Matriz e fällt, noch vor der Bestellung. G-Code für deren Presse gehört nicht ins System — die Maschine programmiert der Lieferant.
6. CI-Build	STEP/DXF → Blechcon (vor der Bestellung automatischer Zuschnitt: SheetNest / nest2d / CAD-N, für große Lose u-nesting in Rust mit Python-Schnittstelle — Verschnitt, Rechnungsprüfung, Kerf aus dfm_blechcon.json); GLB → Konfigurator; Studio-Renderings (Blender headless) → Shopify; PDF/HTML → Anleitungen; BOM → Preise und Lager.

recada-cad/ — vehicles/ Fahrzeugpässe · products/ Konfigurationen (params.json) · catalog/ Profile, Befestigung · accessories/ Zubehör und Adapter · rules/ DFM, Regeln · ci/ Build-Pipelines · generators/ build123d + Bausteine

Versionierung: Gitea (self-hosted) + Git LFS; .FCStd über zippey (lesbare Diffs). Release = Git-Tag mit persönlicher Signatur des Reviewers. Neues Fahrzeug = Pass + params.json — der Skalierungsmechanismus.

3. Dateien und Formate: wie die Wahrheit aufgebaut ist

Kein eigenes Format: die Wahrheit eines Teils = zwei gewöhnliche Dateien. `params.json` — Zahlen und Wörter (SKU, Fahrzeug, Werkstoff, Höhe, Bohrungen), mit Editor oder Prompt zu ändern. `generator.py` — Python/build123d, liest sich wie ein Arbeitsplan: «Blech 120×340×3 → Ecken verrunden → Bohrungen nach Passpunkten → Flansch 90° R3 → untere Biegung im Dachwinkel aus dem Pass». Zahlen kommen aus drei Quellen (`params` / `Pass` / Werkstoffkatalog), im Code stehen fast keine. FCStd ist nicht Quelle, sondern Ergebnis (wie eine `.exe` bei Quelltext): die CI baut sie zum Ansehen. Ausnahme: handgebaute Skelette und Reverse-Referenzen — vollwertige FCStd-Quellen in Git.

Der Konstruktionsbaum ist nicht verschwunden — er ist in den Text umgezogen: Zeilenreihenfolge = Operationsreihenfolge, aber statt zerbrechlicher Verweise auf Flächennummern (Topological Naming Problem) robuste Regeln (`edge="top"`, Punkte nach Namen) und erzeugende Konstrukte (Schleifen über Passpunkte, Bedingungen). Dazu eine Historie mit «warum»: jeder Commit trägt Grund/Prompt («Plattform 40 mm anheben — Box berührt Antenne»), was kein GUI-Baum hat; das ist zugleich QMS-Spur und Trainingsdaten des Systems.

Baugruppen: dasselbe Prinzip eine Ebene höher

Das Baugruppen-JSON beschreibt Zusammensetzung und Bindungen, aber keine Koordinaten: Teile veröffentlichen benannte Schnittstellen («sole», «head» mit Ebene, Bohrungen, Normale), die Baugruppe sagt «sole auf `mount_front_L` setzen». Statt eines Constraint-Solvers (Inventor) — deterministische Positionsrechnung aus Schnittstellen: nichts «wird gelb» bei Änderungen. Befestigung ist Eigenschaft der Verbindung: `bolt_joint()` setzt die Schraube selbst entlang der Achse, rechnet die Länge aus dem Paket, füttert die BOM. Der Prompt «Plattform um 40 mm anheben» ändert einen Skelettparameter — alles wird neu gebaut. Hierarchie: Teil → Produktbaugruppe → Kundenkonfiguration, ein Format und eine Pipeline. Preis des Ansatzes: keine freie Kinematik (Gelenke als Posenparameter oder in der Sandbox).

Rollen der Formate und die «Röhre»

JSON+Python (Wahrheit) → STEP/FCStd (exakte Geometrie) → DXF (Abwicklung) → STL/GLB (Netze). Abwärts — immer, automatisch. Aufwärts — nur Reverse Engineering (teuer, einmalig, am Eingang). STEP ist das «PDF der CAD-Welt»: exakt, aber tot; Ausgang zu Blechcon und Eingang vom OEM. STL/GLB — Netze ohne Maße: Druck und Anzeige, nie für Fertigung und Messung. Jede abgeleitete Datei hat einen Wahrheits-Vater (SKU+Revision im Namen); eine Waisendatei ist ein Prozessverstoß. Bei Konflikt hat immer die obere in der Röhre recht, die unteren werden neu gebaut. Heterogene Quellen (Fahrzeugscan + fremdes Träger-STEP) werden nie direkt gekoppelt — jede wird in einen Pass-JSON übersetzt, gekoppelt wird in der Sprache der Pässe; die Beine sind die materialisierte Brücke zwischen zwei Pässen.

FFF-Regel (Form-Fit-Function)

Passung und Funktion unverändert → neue Revision derselben Artikelnummer (austauschbar, das Lager mischt). Verändert → neue SKU mit Ersetzungspolitik. Schützt vor einem Artikelzoo und stiller Inkompatibilität. Maschinell ausgeführt: der Validator vergleicht benannte Schnittstellen zwischen Revisionen; «Funktion erhalten?» bleibt der einzige Punkt für die menschliche Unterschrift.

Referenz der Dateitypen: was in Git liegt und wie damit gearbeitet wird

Dateityp	Was es ist	In Git?	Wer bearbeitet wie
params.json	Parameter des Teils/Produkts: SKU, Fahrzeug, Werkstoff, Maße, Status; daneben <code>spec.yaml</code> : Absicht, Schnittstellen, Abnahmebedingungen, Non-Goals (Aufgabenvertrag)	Ja, Klartext — Hauptquelle der Wahrheit	Ingenieur (Editor/Webformular), LLM per Prompt; jede Änderung ein Commit mit Grund; in main nur per PR
generator.py und recada_lib	Python/build123d: Geometrieaufbau-Code; Bausteinbibliothek (<code>bent_flange</code> , <code>hole_array</code> , <code>bolt_joint</code>)	Ja, Klartext — Quelle der Wahrheit	Senior/Inhaber (+LLM mit Review); Versuche im Zweig <code>convert/*</code> , Squash-Merge bei Abnahme

Dateityp	Was es ist	In Git?	Wer bearbeitet wie
Pässe: vehicle_*.json, accessory_*.json	Befestigungspunkte mit Normalen, Winkel, Stützflächen (Splines), Schnittstellen, Masse/Schwerpunkt, geometry_source mit Genauigkeit	Ja — Quelle der Wahrheit	Einmalig bei der Digitalisierung erstellt (Mensch oder KI-Agent, menschliche Unterschrift Pflicht); bei Pilotmontagen präzisiert
Baugruppen-JSON	Zusammensetzung und Bindungen über benannte Schnittstellen (ohne Koordinaten)	Ja — Quelle der Wahrheit	Ingenieur/LLM; dieselben PR-Regeln
rules/: DFM, Werkstoffe, Tarife	Blechcon-Regeln, materials.json (1.4301, R3), GLS-Limits, Befestigungstabelle, FEM-Schwellen	Ja — Quelle der Wahrheit	Inhaber/Lead; ergänzt aus Reklamationen und Review des Blechcon-Technologen (CAPA-Zyklus)
YAML der Anleitungen	Montageschritte: Teil, Befestigung, Anzugsmoment	Ja	Ingenieur; CI rendert PDF/HTML/Unity-Szenen selbst
.FCStd (FreeCAD)	Zwei Fälle! (a) Skelette und Reverse-Referenzen — handgebaut; (b) Ansichtsbaugruppen — von der CI gebaut	(a) Ja, via zippey (lesbare Diffs) + LFS; (b) NEIN — Artefakt	(a) Inhaber/Senior in FreeCAD; (b) wird nie bearbeitet — neu gebaut
STEP / DXF	Exakte «tote» Geometrie: Austausch mit Blechcon und Annahme von OEM/Ausführenden	Eingehend — Archiv (LFS) als Rohstoff; ausgehend — NEIN, CI-Artefakte mit SKU+Revision im Namen	Nicht bearbeitet; ausgehend nur aus der CI per Tag zu Blechcon; eingehend über Extraktion in den Pass
STL / GLB	Netze ohne Maße: Prototypendruck (STL mit Empfänger-ID), Konfigurator (GLB vergrößert)	NEIN — CI-Artefakte	Nicht bearbeitet; für Fertigung und Messung grundsätzlich ungeeignet
Dachscans (STL), STEP vom OEM	Rohstoff der Digitalisierung	Archiv in LFS (einmalig)	Nicht bearbeitet; Rolle — Kontrollreferenz für Abstände und Quelle des Passes
BOM, catalog.json, Renderings, PDF-Anleitungen	Ableitungen für Lager, Konfigurator, Kunden	NEIN — CI-Artefakte (Artefaktsspeicher; Release-Artefakte dauerhaft)	Nicht bearbeitet; Änderungen nur über die Wahrheit und Neuaufbau
Prüf- und FEM-Berichte	Protokolle physischer Prüfungen, Referenzrechnungen (Statik, Modal)	Ja (LFS) — an den Release-Tag angehängt	Nur hinzugefügt; Verknüpfung «Revision ↔ Protokoll» = QMS-Spur

Drei eiserne Regeln für das ganze Team: (1) bearbeitet wird nur die Wahrheit (JSON, .py, Pässe, rules, YAML, Skelette) — alles andere wird neu gebaut, eine Artefaktdatei wird nie von Hand angefasst, auch nicht «schnell korrigiert»; (2) eine Datei ohne SKU und Revision im Namen darf den Server nicht verlassen; (3) bei Versionsabweichung hat immer die Datei recht, die in der Wahrheitsröhre höher liegt — die unteren werden neu gebaut, nicht repariert. Ein Verstoß gegen Regel 1 (manuelles STEP in die Fertigung an params vorbei) ist das Hauptrisiko für den Tod des Systems, deshalb ist der Versand an Blechcon technisch nur aus CI-Artefakten eines Release-Tags möglich.

4. Wie der Mensch konstruiert

Prinzip «Vorlage, nicht Kopie»: Baum/Generator entsteht aus der konstruktiven Absicht (Basis → Flansche → Biegungen → Befestigung, jedes Maß sinnvoll und an einen Parameter gebunden), nicht als Kopie aller Artefakte eines konkreten Stanzteils — die Kopie ist zerbrechlich und bricht bei großen Änderungen, die Vorlage hält. Weg 1 — neuer Teiletyp: kreative Suche von Hand in der FreeCAD-Sandbox (mit der Maus, wie gewohnt) → die fertige Konstruktion wird zur messbaren Referenz (Volumen, Masse, Oberflächenpunkte) → schrittweise Übersetzung in Code (Kontur → Abgleich → Biegungen → Abgleich → Bohrungen → Abgleich), jeder Versuch wird automatisch mit der Referenz verglichen; Versuche leben in einem eigenen Zweig und werden bei der Abnahme zu einem Commit zusammengefasst; 3 LLM-Fehlschläge in Folge → Eskalation an den Menschen. Weg 2 — täglich: Zahlen in params ändern (Editor/Formular/Prompt) → Neuaufbau → Ansicht; CAD wird zum Ort des Ansehens. Weg 3 — LLM als Übersetzer: der Mensch beschreibt in Worten oder Skizze, Claude schreibt build123d nach Bibliotheksmustern, die CI baut, die Metrologie meldet die Abweichung, das LLM korrigiert.

Teileklassen nach Übersetzungskosten

Klasse	Übersetzung ins System	Aufwand
Profile (Aluminium, Walzware)	DXF-Querschnitt importieren + extrudieren; Parameter — Länge	eine Stunde je Katalog
Gebogenes Blech (Beine, Adapter)	Reverse des Arbeitsplans in Code; schneller mit Bausteinbibliothek	Stunden-Tage
Guss / skulpturale Teile	NICHT übersetzen: Schnittstellenpass (20 min) + STEP als tote Referenz; Offsets von Stützflächen bei Bedarf; vollständiges Reverse nur bei geplanter Größenfamilie	Minuten
Kaufteile (Befestigung)	Normteilibibliotheken build123d	0

Grenzregel innerhalb eines Teils: Form — von Hand (einmalig), Schnittstellen und Anpassung — als Code. Menschen bleiben im System für vier Rollen, in denen die Maschine schwach ist: Aufgabenstellung (Kenntnis der realen Welt), Topologie («wie überhaupt lösen»), Urteil («gültig, aber dumm» — Unterschrift des Reviewers), Lehren des Systems (Erfahrung → Regeln). Die Stunde «Kontur nachzeichnen» nullt das System; die Stunde «verstehen, wie man die Reling umgeht» wird zum ewigen Vermögenswert.

Der Eingang ist für jedes CAD offen: der Ausführende zeichnet in SolidWorks/Inventor/Catia/FreeCAD — liefert STEP + Zeichnung mit Maßen; die Übersetzung in Code hängt nicht vom Ausgangs-CAD ab. build123d von Wettbewerbsteilnehmern zu verlangen = den Pool zehnfach verengen. Lizenzen: fremde Lizenzen sind nicht prüfbar — eine Erklärung im Vertrag überträgt die Verantwortung, und eine VDI-Sitzung löst das Problem vollständig (legaler Arbeitsplatz ohne eine einzige Lizenz beim Ausführenden) und bereinigt die Lieferkette für ISO/GeschGehG. Unterauftrag ist per Konstruktion sicher: Bezahlung je Teil, das den Validator passiert hat; Konto persönlich (Sitzungen aus zwei Ländern sind in den Logs sichtbar); auf der Vertrauensleiter steigt der Unterauftragnehmer nicht.

Ingenieurvorlauf: was bereits praktisch geprüft ist

Reverse Engineering mit metrologischer Abnahme (point-sampling an Kontrollpunkten): Bracket_L — 100 % Übereinstimmung; Bracket_R — Spiegel, 99,12 %; wind_side_v2 — 100,000 % an 88 Punkten; Part55 → neuer Bracket_88_W30 mit voller DFM-Kontrolle; Baugruppe Defender 110 (LRDE-2BR1) — 99,9994 %, SheetMetal-Baum vom Skelett. Werkstoff überall 1.4301 2B, t = 3,0, K-Faktor aus Tabelle. Kritische Lehre der Sitzungen: die Installation von networkx aktivierte den SheetMetal Unfolder V2, der die wahren Biegewinkel 99,389° und 135° zeigte statt der von V1 fehlerhaft gemessenen 80,6° und 45° (Ergänzungswinkel) — ein Fehler mit direkten Fertigungsfolgen; Regel: Abwicklungen nur über V2. Konstruktionsvokabular fixiert: Flansch — Seitenflächen mit Bohrungen, Steg — zentrale Fläche mit Logo.

Gelernte technische Regeln (in skills): SMBendWall scheitert an unterbrochenen Biegungen → Revolution oder Rekonstruktion aus Flächenprismen; extend vor dem Anhängen von Tochterwänden setzen (sonst zerreit TNP die Bindungen); ein gescheiterter Konstruktor hinterlässt ein Null-Objekt — sofort löschen; Boolescher Schnitt ist zur Verifikation unzuverlässig — nur point-sampling über

distToShape; vor der Kantensuche volles touch()+recompute (der SheetMetal-Cache setzt sich nicht selbst zurück); Bindungen nur über SubShapeBinder (Schutz vor TNP); für Teile mit Prägungen ein Zwei-Konfigurations-Baum (flaches Modell für die Abwicklung + finales mit Prägung). Zu installierende Module: SheetMetal 0.8.22 (Solid-to-SheetMetal, Biegemarkierungen für die Blechcon-Pressen, Anzeige defekter Verweise), Defeaturing WB (Fasen in der Biegezone entfernen → abwickeln → Ausschnitte zurückholen), Curves WB (Bereinigung von B-Spline-Flächen aus Inventor-STEP), networkx. Werkstoffe — Material Definition Sheet (Tabelle R/T → K-Faktor statt starrem K = 0,42): eine Tabelle speist Claude und die parametrischen Modelle. Unattended Unfold — headless-Abwicklung für die CI. Einstiegspunkt des Repositories — CLAUDE.md: Produktbeschreibung, Standardwerkstoffe, welche Skills wann anzuwenden sind.

Validator mit Autokorrektur: die nächste DFM-Stufe

Regeln werden in drei Schichten digitalisiert: Zahlen → dfm_blechcon.json (R = 1xt, Mindestflansch 15 mm bei Edelstahl, Abstand Bohrung zur Biegung, Entlastungsnut, Abstände $\geq t$, Blechlimits, GLS-Maße); Korrekturen → features.py (Generatoren für Freischnitte: Langloch, Entlastungsnut, Eckfreischnitt); Methoden → skills (Markdown: «wie denken»). Der Validator sagt nicht nur «Fehler: Bohrung zu nah an der Biegung», sondern schlägt die Korrektur vor: «Konflikt → schlage relief_obround vor, hier die Vorschau» — in der RECADA-Workbench rote Hervorhebung + Knopf «korrigieren», in der CI blockiert derselbe Code headless die Freigabe des Unkorrigierten. Kalibrierung der Bibliothek an der realen Fertigung: von Blechcon angenommene Teile mit deren Freischnittkorrekturen werden vermessen (einfaches Reverse), die Formeln «Freischnittgröße aus t/R/d» werden in features.py eingebaut; jede Korrektur-E-Mail von Blechcon wird eine Zeile in dfm_blechcon.json, und der Fehler wiederholt sich bei niemandem. Regelbasiertes Design hängt nicht mehr davon ab, ob der Ingenieur das PDF des Lieferanten im Kopf hat.

Reverse ohne GUI: STEP → Python-Code (Methode und Stack)

Der Konstruktionsbaum als solcher ist nicht nötig — nötig ist Python-Code: er speichert dieselbe Operationsfolge, aber ohne zerbrechliche Flächenverweise (TNP), mit Schleifen und Bedingungen, Diffs und «warum»-Historie. Einen Ansichtsbaum kann die CI fürs Auge bauen, aber die Wahrheit ist der Code. Die gesamte Reverse-Röhre läuft headless (build123d/OCP — derselbe OpenCASCADE-Kern wie FreeCAD): ohne GUI, ohne MCP-Sitzungen, wiederholbar und CI-tauglich (zehn STEP über Nacht). Algorithmus der automatischen Vermessung (step_probe.py): B-Rep-Traversierung über TopExp_Explorer → Flächenklassifikation (eben = Flansche/Stege, zylindrisch = Biegungen, schmale Stirnflächen = Dicke) → Aufbau des AAG (attributierter Flächenadjazenzgraph, networkx): Knoten — Flächen, Kanten — Übergänge → das Abtrennen der Stirnflächen spaltet den Graphen in zwei isolierte äquidistante Schalen (oben/unten), ihr Abgleich liefert die neutrale Fläche und die Dicke t → Extraktion der Biegeparameter (Radius, Winkel, Achse; numpy: Normalen, PCA der Achsen), der Bohrungen (Zylinder mit Achsen und D), der Außenkontur. Ausgabe — measurements.json.

Das Teileskelett — Hauptergebnis der Vermessung und Basis der Parametrisierung

Schlüsselentscheidung: der Vermesser liefert keine «Maßliste», sondern das SKELETT des Teils — eine mittlere Darstellung ohne Dicke: Platten (Flanschebenen mit Konturen) + Biegelinien (Achsen, Winkel, Radien) + Dicke als EIN Parameter ($t = 2 \times$ Abstand von der Mittelfläche zur Grenze). Reverse wird aus «Formwiederherstellung» zur «Skelettwiederherstellung», mit drei Effekten. (1) Dicke ist keine Geometrie mehr: Wechsel 3 mm → 2 mm ist die Änderung einer Zahl, kein Neuaufbau aller Flächen (relevant: beide Dicken werden verwendet). (2) Teileskelett und gemeinsames Baugruppenskelett sprechen dieselbe Sprache — Ebenen und Linien; daher wird die Kopplung «H der Plattform geändert → Flanschwinkel des Beins neu berechnet» geometrisch direkt: die Biegelinie liegt im Schnitt der vom Baugruppenskelett vorgegebenen Ebenen. (3) Die Parametrisierung ergibt sich fast von selbst: Plattenlängen, Winkel zwischen ihnen und Lagen der Biegelinien sind genau die Größen, die sich von Fahrzeug zu Fahrzeug ändern, während Fertigungsartefakte (Körnermarken, zufällige Verrundungen) gar nicht ins Skelett gelangen — sie kollabieren mit der Dicke. Die Medial Axis filtert das Rauschen für den Ingenieur.

Umsetzung — ohne «echte» Medial Axis Transform (OCC hat keine Standardfunktion, robuste Mittelflächen auf realen B-Reps sind nichttrivial): das Skelett wird aus dem bereits vorhandenen AAG gebaut — Paare äquidistanter Flächen liefern Mittelebenen, Biegezyylinder liefern Mittelachsen und

Radien. Das Ergebnis ist identisch, der Code der ohnehin geplante. Das Teileskelett wird in measurements.json als Hauptstruktur gespeichert; daran werden Parameter (spec.yaml) und Regeln (rules/) gehängt.

Krümmung als robuster Detektor für Biegungen und Prägungen

Zweites, von der Flächenbeschreibung unabhängiges Kriterium: für jede Fläche werden Hauptkrümmungen und abgeleitete Größen berechnet — die Ebene liefert Gauß $K = 0$ und Mittel $H = 0$; die zylindrische Biegung $K = 0$ bei $H \neq 0$, wobei die Richtung der Null-Hauptkrümmung die Biegeachse ist und die zweite Krümmung $1/R$ liefert; Prägungen und Tiefzüge $K \neq 0$ (nicht abwickelbare Bereiche). Praktischer Wert: das Kriterium funktioniert unabhängig davon, wie STEP die Fläche beschrieben hat — beim Export aus Inventor kommen Ebenen und Zylinder oft als B-Spline-Approximation, wo es formal keinen «Zylinder» gibt und die Suche nach Flächentyp scheitert; Krümmung hat jede Fläche. Bonus: $K \neq 0$ markiert das Teil sofort als prägungshaltig, also mit Zwei-Konfigurations-Ansatz — noch vor dem Abwickeln. Reihenfolge im Vermesser: erst Flächentyp über OCC, bei Fehlschlag Krümmungsmessung.

Automatische Korrektheitsprüfung der Abwicklung (Gauß-Bonnet): für eine korrekte flache Abwicklung ergibt die Summe der Winkeldefekte entlang der Konturen 2π ; eine Abweichung bedeutet eine topologisch falsche Abwicklung — Falten oder Überlappungen. Eine Formel in der Abnahme: billige Versicherung, die grobe Unfolder-Fehler automatisch fängt, ohne menschlichen Blick auf das DXF (ersetzt die Metrologie nicht).

Was die Automatik NICHT tut: die Wahl, welche Zahl Parameter ist (Höhe, ändert sich je Fahrzeug), welche Regel (R3 immer), welche Formel (Breite aus Länge) und welche ein nicht reproduzierbares Fertigungsartefakt. Das ist die konstruktive Absicht; sie setzt der Mensch oder Claude mit Skills, indem er measurements.json als Text liest. Danach — spec.yaml + generator.py + params.json.

Detektor für Entlastungsschnitte (bend reliefs)

Freischnitte verhindern Reißen und Stauchen beim Biegen, verkomplizieren aber die Topologie — sie verursachen die «unterbrochenen Biegungen», an denen SMBendWall scheiterte und Unfolder V1 sich verhedderte. Erkennung in drei Schritten, im selben Vermesser: (1) billige Heuristik über Konturen und Kanten der Biegefläche — eine kanonische Biegung hat 1 Kontur und 4 Kanten; mehr als eine Kontur = Durchbruch in der Biegung, mehr als vier Kanten = Ausschnitte an den Rändern; (2) Muster im AAG — eine ideale Biegung grenzt an genau zwei ebene Flansche, ein Freischnitt fügt kleine Stirnflächen als Nachbarn hinzu oder zerlegt den Zylinder in mehrere koaxiale (Teilgraphsuche im vorhandenen Graphen, nichts Neues nötig); (3) UV-Projektion der Biegegrenzen auf den Zylinder — die Abweichung vom Rechteck liefert die exakten Maße des Ausschnitts (Tiefe, Breite, Abstand); nur auf gefundene Fälle angewandt.

Der Kernunterschied zum «bloßen Kopieren»: im System ist der Freischnitt eine REGEL, nicht eine Form aus dem Original. Der Generator berechnet ihn aus Dicke, Radius und Knotentyp (features.py), statt die Kontur zu übertragen. Daher hat der Detektor zwei Rollen: Signal «dieser Knoten braucht technologisch einen Freischnitt» (sonst meldet der Validator einen Konflikt und schlägt relief_obround vor) und Kalibrierquelle — die Vermessung der Freischnitte an von Blechcon angenommenen Teilen liefert die reale Abhängigkeit «Größe aus $t/R/d$ », die dauerhaft in features.py eingebaut wird. Dritter, praktischer Gewinn: das Wissen «an dieser Biegung Unterbrechung durch Freischnitt» erlaubt die automatische Wahl der erprobten Abwicklungsumgehung (Revolution statt SMBendWall oder Defeaturing: Ausschnitte entfernen → saubere Biegung abwickeln → Ausschnitte flach zurückholen) — was heute in jeder Sitzung von Hand entschieden wird, wird ein Zweig im Code. Werkzeuge dieselben: OCP (TopExp_Explorer, BRepAdaptor_Surface) + networkx; ein separates pythonocc-core ist nicht nötig.

Hybrid «exakte Zahlen + Sehen» und agentische Korrekturschleife

Wesentliche Ergänzung des Vermessers: Geometrie und Semantik werden mit verschiedenen Mitteln gewonnen und verbunden. OCC/AAG/FTG liefern exakte Zahlen (Dicke, Radien, Winkel, Bohrungskordinaten) — darin ist die Maschine fehlerfrei. Das Rendering des Teils in 4 Ansichten (Projektionen + Isometrie aus build123d) plus ein multimodales Modell liefern die Rollensemantik: «das ist ein Freischnitt am Biegefuß», «das ist ein Falz zur Kantenversteifung», «das ist ein U-Halter», «das ist ein Stanzartefakt, nicht reproduzieren». Genau die Semantik fehlte: der AAG misst, weiß aber

nicht, was Parameter, was Regel und was Artefakt ist. Funktioniert heute ohne Modelltraining und ohne Datensätze — Rendering, Blick, Markierung in measurements.json, Generierung.

Agentische Autokorrekturschleife (Regelmodus des Reverse): Extraktor → LLM schreibt generator.py → CI baut STEP → Abgleich mit dem Original → Abweichungen gehen als Zahlen an das Modell zurück («Ausschnitt um 2 mm verschoben, Zeile 45») → Korrektur → Wiederholung bis Konvergenz.

Stoppregel wie bisher: 3 erfolglose Iterationen in Folge → Eskalation an den Menschen mit Versuchstagebuch (sonst dreht sich das Modell ewig). Richter ist ausschließlich die numerische Metrologie point-sampling $\geq 99,9\%$ und exakte Schnittstellen; der visuelle Diff (Überlagerung der Renderings von Original und Build mit Hervorhebung) dient als Navigation zum Fehler und Erklärung für den Menschen, NICHT als Abnahmekriterium: ein Teil kann «gleich aussehen» bei einem halben Millimeter Abweichung an der Passstelle.

Evolutionäre Parametersuche — für die 10–20 % Fälle, in denen die Erkennung aufgibt (freie Radien, skulpturale Übergänge): die Vorlage schreibt der Mensch, die Zahlen findet der Optimierer (scipy / CMA-ES, ohne ML), indem er dieselbe metrologische Abweichung des point-sampling minimiert. Billig, deterministisch, als Nachtlauf in die CI eingebettet.

Origami-Kinematik und modale Signatur

Kinematische Abwicklung («starres Origami»): das Teil wird als starre ebene Platten (Flansche) und eindimensionale Gelenke entlang der Biegeachsen dargestellt; aufeinanderfolgende Drehungen um die Achsen bis zu Nullwinkeln ergeben den flachen Zuschnitt. Ein Algorithmus liefert zwei Ergebnisse zugleich: die Lösertrajektorie (Folge der Drehungen) ist die Biegereihenfolge, der Endzustand der Zuschnitt. Die Methode schließt zwei Aufgaben: Abwicklung ohne Unfolder (Gelenkwinkel = Biegewinkel, Endzustand = Zuschnittkontur) und automatische Selbstdurchdringungsprüfung beim Falten — die Bending Feasibility, die sonst die regressive Abwicklung rechnet. Umsetzung — reine Geometrie in numpy (sequentielle Drehungen + Schnitttest); Physik-Engines (PyBullet/MuJoCo) sind für starres Origami überflüssig.

Modale Signatur als finaler Validator des Reverse: für Original und rekonstruiertes Modell werden die ersten ~5 Eigenfrequenzen berechnet (CalculiX). Übereinstimmung der Spektren beweist mechanische, nicht nur geometrische Äquivalenz: Frequenzen reagieren auf Dicke, Radien in Biegezonen und Massenverteilung — was die punktuelle Metrologie übersehen kann. Angewandt auf verantwortliche Teile (Beine, tragende Halter), nicht Pflicht für Kleinteile; die FEM-Infrastruktur wird für den Validator ohnehin gebaut, der Check ist fast kostenlos. Schöner Schluss: dieselbe Modalanalyse, die das Unternehmen bereits durchgeführt hat, dient als Reverse-Abnahme und als Regel « $f_1 \geq \text{Strouhal-Schwelle}$ » gegen Brummen bei Geschwindigkeit.

Werkstattpraxis im Code: Bezugsflächen, Walzrichtung, Toleranzen, Werkzeugspuren

Bezug auf technologische Bezugsflächen (datums), nicht auf den CAD-Nullpunkt: das Teil wird von den Ebenen beschrieben, von denen die Qualitätskontrolle misst (Bezug auf der Messplatte, KMG-Taster) und von denen der Bediener es am Hinteranschlag der Abkantpresse positioniert. Dann heißen die Generatorvariablen STOP_DISTANCE_X1 statt offset_x, die Maße im Code entsprechen den in der Fertigung kontrollierten, und die Maßkette geht vom Bezug aus — weniger akkumulierter Fehler. In spec.yaml erscheint der Abschnitt datums (Bezugsebenen und Aufnahmepunkte).

Walzrichtung: Biegen entlang der Faser erzeugt bei Edelstahl Mikrorisse am Außenradius — bei Radien nahe dem Minimum ein reales Risiko. Die Regel verbindet Konstruktion und Zuschnitt: liegt die Biegeachse parallel zur Walzrichtung (die durch die Orientierung des Teils auf dem Blech beim Nesting bestimmt wird), verlangt der Validator entweder einen größeren Mindestradius oder eine Umorientierung des Teils im Schachtelplan — und schreibt die Warnung in den Generatorkommentar.

Dicke als Verteilung, nicht als Zahl: ein reales «3-mm»-Blech misst 2,93. Der Vermesser misst die Dicke an N Punkten und liefert Nennwert plus Streuung; in den Generator geht der Nennwert, in die Spezifikation die Toleranz mit assert-Prüfung, der K-Faktor wird aus der tatsächlichen Dicke berechnet. Der Vermesser stolpert nicht mehr über «schwimmende» Dicke.

Matrizenspuren als Technologiequelle: beim Freibiegen bleiben am Außenradius zwei parallele Übergangslinien zwischen Zylinder und Ebene — ihr Abstand verrät die V-Nut-Breite, auf der das Teil gebogen wurde. Aus einem fremden Teil wird nicht nur die Geometrie, sondern auch die Technologie

rekonstruiert, und der Radius sofort an die Standardreihe der Matrizen gebunden ($V = 6, 8, 10, 12, 16$ mm) — direkte Ergänzung der Werkzeugtabelle des Lieferanten. Die Kollisionsprüfung wird präziser: nicht «biegt es sich überhaupt», sondern «biegt es sich auf dieser Presse in dieser Reihenfolge» — das Teil darf bei der 2. und 3. Biegung den oberen Balken nicht berühren (Hüllmodelle der Werkzeuge werden bei Blechcon erfragt; Amada/Trumpf-Kataloge sind nicht nötig). In Reserve ohne Priorität: Erkennung von Steckverbindungen und Schweißspalten — relevant, falls geschweißte Baugruppen kommen (derzeit Schraubverbindungen).

Geprüft und verworfen (um nicht im Kreis zu laufen)

GNN-Flächenerkennung (BRepNet/CADNet/Sheet-metalNet) — keine offenen Modelle für Blech, nötig wäre ein markierter Korpus von tausenden Teilen, Genauigkeit probabilistisch; unsere Teileklasse wird deterministisch durch Regeln erkannt. ML-Rückfederungsprädiktor auf FEA-Datensätzen (ANSYS/LS-DYNA — kostenpflichtig) — löst, was der Biegebediener an seiner Presse schon löst; unser Weg ist eine Zeile in `dfm_blechcon.json` aus der Blechcon-Praxis. Spezialisiertes VLM «Bild → Code» (CAD-Coder) — Forschungsmodelle sind schwächer als die Kombination «exakte OCC-Zahlen + Reasoning eines starken Modells». EvoCAD (Variantenpopulation mit VLM-Jury) — teuer in Tokens und optimiert visuelle Ähnlichkeit statt Maßgenauigkeit. Gaussian Splatting vom Telefon — Genauigkeit auf glänzendem Metall weit von den Schnittstellenanforderungen; die praktische Variante (Photogrammetrie der Hülle + Messschieber an der Befestigung) ist bereits angenommen; wieder aufgreifen bei der Digitalisierung fremden Zubehörs «im Feld». Analysis Situs, separates `pythonocc-core`, `CadQuery` — duplizieren den Stack. Inverse CSG («Luftabdruck»: Teil vom Hüllquader abziehen und das Negativvolumen in Primitive zerlegen) — stützt sich auf Boolesche Operationen, die nach unserer eigenen Lehre auf komplexen Körpern Unsinn liefern; Radien und Bohrungen werden ohnehin direkt von den Zylinderflächen genommen. Synthetischer Laserscan des STEP + Punktwolken-Netze (CAD-Recode / Point2CAD) — bewusster Genauigkeitsverlust für Robustheit gegen STEP-Defekte, die mit zwei Zeilen OCC-Healing geheilt werden. Differenzierbares CAD (SDF + Gradientenabstieg) — technisch funktionsfähig, löst aber dieselbe Aufgabe wie der angenommene CMA-ES, verlangt aber differenzierbares Rendering und Hyperparameter; wieder aufgreifen, falls CMA-ES an Geschwindigkeit scheitert. Programmsynthese über SMT-Löser (SyGuS/Z3) — «bewiesene Korrektheit» ist unerreichbar: Löser arbeiten mit Logik und Arithmetik, nicht mit B-Rep; unsere Garantie ist gemessen (Metrologie + Spec-Lauf), nicht bewiesen. Latente Diffusion von CAD-Bäumen (GenCAD-3D) und Laplace-Beltrami-Spektrum — Training auf Millionen Programmen / Mathematik zur Ähnlichkeitssuche; Ebenen und Zylinder werden direkt erkannt. PINN für dynamischen K-Faktor — Plastizitätsgleichungen für eine Zahl, die aus dem Material Definition Sheet kommt und an angenommenen Blechcon-Teilen kalibriert wird. RL-Agent für die Abwicklung — die Aufgabe ist bereits zweimal deterministisch gelöst: regressive Abwicklung und Origami-Kinematik liefern Biegereihenfolge und Selbstdurchdringungsprüfung in Sekunden, reproduzierbar. Neuro-symbolische DSL mit AST-Validierung über Z3 — die Sorge «das LLM irrt syntaktisch» ist billiger beseitigt: Code wird in der CI ausgeführt (Syntax fällt sofort und geht ans Modell zurück), Semantik fangen Validator und Metrologie. INR/SIREN (neuronales Belegungsfeld) — Genauigkeitsverlust für Gradienten dort, wo exakte Ebenen direkt im B-Rep liegen. Quanteninspirierter genetischer Algorithmus — Hülle um gewöhnliche evolutionäre Suche; der angenommene CMA-ES leistet dasselbe mit prüfbarer Konvergenz. Persistente Homologie (TDA) — nur die Essenz übernommen (Integritätsinvariante) als Checkliste des Elementerhalts, ohne algebraischen Apparat. Optische Kaustiken, Holonomie/Paralleltransport (der Diederwinkel ergibt sich als Skalarprodukt der Normalen in einer Zeile), Wärmefluss nach Fourier zur Ausschnittsuche (Aufzählung innerer Konturen ist zwei Größenordnungen billiger) — Exotik um der Exotik willen. Karte der Martensitumwandlung ($\gamma \rightarrow \alpha'$ bei Kaltumformung) — physikalisch korrekt, aber die Rechnung braucht eine Umformsimulation, für die man schon wissen muss, wo die Biegungen sind: Zirkellokig; der reale Wert des Themas liegt anderswo — Kaltverfestigung in Biegezonen beeinflusst Korrosionsbeständigkeit und Magnetismus des fertigen Teils, eine Produktqualitätsfrage, keine Reverse-Frage.

Die Klasse «fundamentalphysikalischer» Methoden wurde vollständig geprüft und verworfen (Berry-Phase, Ricci-/Yamabe-Fluss, Eichtheorie der Defekte, Knotenpolynome, inverses Streuproblem, holographisches Prinzip, symplektische Mechanik, topologische Randzustände, Onsager-Funktional, Kolmogorov-Komplexität): sie alle lösen drei Aufgaben — Biegewinkel und -radien, Abwicklung, Ausschnittsuche —, die bereits durch Krümmung, Origami-Kinematik und Konturzahl mit Maschinengenauigkeit in Sekunden geschlossen sind. Die Genauigkeit des Systems scheitert heute

nicht an der Erkennungsmathematik, sondern an Physik und Daten: K-Faktor des realen Blechs, Rückfederungskompensation auf der konkreten Presse, Toleranzen des Ausgangs-STEP, Genauigkeit des Dachscans — diese verringern jene Methoden nicht. Lesbarkeit des Codes (Schleifen statt Wiederholungen) erreicht man durch Anweisung im Prompt und Review, nicht durch ein Variationsproblem. Die Klasse «Gamedev-mathematischer» Ansätze wurde ebenfalls geprüft und verworfen: das IK-Rig fällt mit der bereits angenommenen Origami-Kinematik zusammen (angenehme Bestätigung der Natürlichkeit); SDF-Raymarching, Ricci-Fluss, Shape Grammars und Wave Function Collapse, MCTS-Codesuche, Voxel-Octrees, zelluläre Automaten — früher verworfene Klassen in neuer Verpackung; Crowdsourcing à la Foldit braucht ein Spielerpublikum, das es nicht gibt. Aus der Morse-Theorie nur die Folgerung, die sich ohnehin aus dem Skelettansatz ergibt: das Teil im Code von der Basisebene zur Peripherie bauen, nicht in beliebiger Reihenfolge.

Kriterium «100%-Kopie mit parametrischen Änderungen» (in rules/)

Das Reverse ist abgeschlossen, wenn alle Bedingungen erfüllt sind — (1) Schnittstellen (Bohrungen, Passflächen, Winkel) stimmen exakt; (2) Metrologie point-sampling bei Nennparametern $\geq 99,9\%$ gegen das Ausgangs-STEP; (3) der Lauf mit GEÄNDERTEN Parametern ist grün bei Validator und Spec-Test (Beweis, dass die Parametrik lebt und keine Dekoration ist); (4) Checkliste des Elementerhalts: Bohrungen waren N — sind N, Ausschnitte und Freischnitte waren M — sind M (direkt gezählt: Durchgangsbohrungen — paarige Zylinderflächen, Ausschnitte — innere Konturen; fängt den ärgerlichsten Fehler «Loch vergessen»); (5) die Liste bewusster Abweichungen vom Original steht in der Spezifikation (Körnermarken und zufällige Verrundungen werden nicht reproduziert; die technologische Lasche wird reproduziert, aber als Regel, nicht als Kopie). Die restlichen Zehntelprozente sind meist K-Faktor, Spline-Genauigkeit und Toleranzen des Ausgangs-STEP — ihnen nachzujagen hieße fremde Exportfehler zu reproduzieren.

Stack (nur was bereits im System ist): build123d/OCP — STEP-Lesen und B-Rep; networkx — AAG; numpy — Vektormathematik; Metrologie — distToShape point-sampling; Abwicklung — SheetMetalUnfolder V2 headless (FreeCAD hier importierbare Bibliothek, keine Umgebung), mit Migration auf YouCanNotUnfold (schneller Unfolder auf networkx, Drop-in-Ersatz) oder die eingebaute build123d-Abwicklung nach Reife; Nesting — SheetNest / nest2d / CAD-N / u-nesting (NFP-Algorithmen mit Kerf). Als überflüssig verworfen: separates pythonocc-core (derselbe Kern über OCP), CadQuery (dupliziert build123d), Analysis Situs (schwere C++-Plattform für das, was ~100 Zeilen networkx leisten). Rückfederung: Edelstahl 1.4301 —

[Fortsetzung in Teil 2: Abschnitt 4 (Ende), 5-15c]

RECADA / Airholder — vollständige Projektbeschreibung v50

Deutsche Übersetzung, Teil 2 von 5 — Abschnitt 4 (Fortsetzung), Abschnitte 5 bis 15

4. (Fortsetzung) Rückfederung, Spec-Driven Development, Baugruppenprüfungen

Rückfederung (springback): Edelstahl 1.4301 verfestigt stark — der am fertigen Teil gemessene Winkel enthält die Rückfederung bereits; die Kompensation ist Sache des Biegers, aber ihre realen Werte in Abhängigkeit von t/R gehören als Zeile aus der Blechcon-Praxis in dfm_blechcon.json.

Spec-Driven Development: die Aufgabenspezifikation als Vertrag

Das System folgt bereits den SDD-Prinzipien (Wahrheit ist die Spezifikation, Geometrie ist Artefakt, der Validator ist eine ausführbare Spezifikation der Einschränkungen, die «Warum»-Historie steckt in den Commits). Ergänzt um drei Techniken. (1) Spezifikation vor der Geometrie: für jedes neue Teil wird vor der ersten Linie in der Sandbox eine spec.yaml geschrieben — Absicht (intent), Kopplungsschnittstellen (aus den Pässen), prüfbare Abnahmebedingungen (Masse, Last, Abstand, Zielkosten aus der BOM) und ausdrückliche Non-Goals (was NICHT gemacht wird). Die Spezifikation ist eine Wahrheitsdatei in products/ neben params.json. (2) Spezifikation → Test → Umsetzung: die CI erzeugt den Abnahmelaufroutine automatisch aus den constraints, noch bevor die Arbeit beginnt; ein roter Lauf am leeren Teil ist normal, die Arbeit des Ausführenden besteht darin, ihn grün zu machen; die Abnahme verwandelt sich von «das Abgegebene prüfen» in «die Spezifikation grün machen», die Subjektivität des Reviews schrumpft auf die Frage «gültig und nicht dumm?». Die Spezifikation ist die formale Grundlage der Stückvergütung — ein saubererer Vertrag als jedes Fließtext-Lastenheft; der Streit «das Teil ist falsch» wird zu «die Spezifikation war unvollständig», und repariert wird die Spezifikation, was die Vorlage bereichert. (3) Das LLM arbeitet von der Spezifikation, nicht aus dem Chat: der Prompt wird zuerst in spec.yaml materialisiert (Claude formuliert selbst), der Mensch bestätigt in Sekunden, dann folgt die Generierung — die Absicht ist vor der Umsetzung in Git fixiert, «das LLM hat es falsch verstanden» wird an der Spezifikation gefangen, nicht an der Geometrie; die Regel wird in rules/ und in den Systemprompt des RECADA-Assistenten eingebaut. Einführung: Etappe 2, neben dem Validator (der Generator des Abnahmelaufroutines ist ein kleines CI-Skript).

Baugruppe: Prüfungen auf Produktebene

Teile werden zum Dachträger zusammengebaut, deshalb hat die Baugruppe eine eigene Prüfschicht — getrennt von der Teilegeometrie. Die Positionierung ist bereits beschrieben (benannte Schnittstellen statt Constraints, alles vom Skelett); hinzu kommen vier Prüfungen, alle Geometrie und Arithmetik, ohne Physik-Engines und ohne Simulation des Monteurs.

Prüfung	Was sie tut und wozu
Werkzeugzugang (assembly feasibility)	Für jede Schraubverbindung wird ein freier Zugangszylinder geprüft: gerader Kanal von der Schraubenachse nach außen für den Schlüsselkopf (Durchmesser, Hubweg) und Überstand zum Einsetzen des Schafts. Fängt den klassischen Fehler «konstruiert, aber nicht montierbar»; kritisch, weil die Montage auf dem Dach in unbequemer Haltung erfolgt. Zugleich wird geprüft, ob das Teil bei bereits montierten Nachbarn an seinen Platz kommt.
Maßketten (tolerance stack-up)	Die Plattform steht auf vier Beinen: Biegetoleranz ($\pm 0,5-1^\circ$), Lochtoleranz des Lasers, Dachstreuung — die Summe ergibt Schiefstand oder nicht fluchtende Bohrungen am letzten Bein. Das System durchläuft die Kette «Dach → Bein → Profil → nächstes Bein» im Worst Case und statistisch (RSS) und gibt das nötige Mindest-Langloch aus. Beseitigt eine ganze Reklamationsklasse «passt nicht».
Automatische Montagereihenfolge	Die Reihenfolge wird aus dem Verbindungsgraphen abgeleitet (Bein ans Dach → Profil an die Beine → Zubehör ans Profil) unter Berücksichtigung der Zugangsprüfung, statt von Hand geschrieben zu werden. Die Anleitung wird automatisch erzeugt und ist stets aktuell; der Unity-Konfigurator erhält die fertige Sequenz für die interaktive 3D-Anleitung. YAML bleibt für menschliche Hinweise («vor Montage der hinteren Beine nicht festziehen»).

Prüfung	Was sie tut und wozu
spec.yaml der Baugruppe	Auch das Produkt hat einen Abnahmevertrag: Gesamtmasse, zulässige Last, Anzahl eindeutiger SKU (Vereinheitlichungskennzahl — je weniger verschiedene Teile, desto billiger das Lager), Schätzung der Montagezeit nach Anzahl der Befestigungen, Zubehörkompatibilität. Die Baugruppenabnahme ist derselbe grüne Lauf wie beim Teil.

5. Konfigurator (Unity + Shopify) und Anleitungen

Der Konfigurator wird in Unity entwickelt (WebGL für die Website; aus demselben Projekt native mobile Apps mit AR-Anprobe über AR Foundation und Desktop für den Händlerkiosk/die Messe). Der Vertrag CI ↔ Frontend ist engine-unabhängig — catalog.json: SKU, Revision, Status, GLB, Kopplungspunkte, Varianten, Ersetzungsregel; Unity lädt GLB zur Laufzeit über glTFast, Attach-Punkte werden als named nodes gelesen. Möglichkeit «Arbeitsplan im Konfigurator»: die Unity-Szene visualisiert Montageschritte aus denselben YAML-Anleitungen — eine interaktive 3D-Anleitung statt PDF. Risiken von Unity-WebGL: schwerer Build (15–50 MB gegenüber 2–5 bei three.js) — die Erstladezeit auf dem Mobilgerät im Auge behalten (~5 s Schwelle); bei Misserfolg Hybrid: leichte Schaufenster-Vorschau + vollständiger Unity-Konfigurator. GLB wird komprimiert (glTFpack+Draco: zweistellige KB), Revision im Dateinamen gegen CDN-Cache. Kunden sehen nur getaggte Releases. «Lastampel»: Summierung der Zubehörmassen gegen die zulässige Dachlast — Konversionssteigerung, alle Daten liegen bereits im Katalog. Anleitungen aus YAML (Schritt, Teile, Befestigung, Moment) rendert die CI mit aktuellen Bildern; die Kundenkonfiguration ist bekannt — die Anleitung wird für seine Bestellung erzeugt. Ein Konfigurator auf FreeCAD-Basis für den Einzelhandel wurde verworfen (Sekunden je Neuaufbau, Container-Farm, Mobilgeräte): Einzelhandel = vorgefertigte GLB, sofort; VDI-FreeCAD mit RECADA-Panel = CPQ-Werkzeug für Händler und Sonderfälle, Ausgang params.json direkt in die Fertigung.

6. Prototyping: vier Stufen

Stufe	Was geprüft wird	Dauer / Fehlerkosten
1. Digital	Kollisionen, FEM (CalculiX+Gmsh im Validator), AR-Anprobe am Fahrzeug (model-viewer, dieselbe GLB)	Minuten / 0 €
2. Gedruckt	--target=proto → STL, FDM 1:1 (PETG): Passung, Bohrungen, Behinderungen; Empfänger-ID auf der Rückseite geprägt; Drucker 300–500 € amortisiert sich mit der ersten Iteration	Stunden / ~2 €
3. Metall (DC01)	Blechcon als Prototypenwerkstatt; material_override: DC01 statt 1.4301 (2–3× billiger); Kennzeichnung PROTO not for sale	Tage / zweistellige €
4. Pilotsatz	Serienwerkstoff, Montage, Fotodokumentation → Korrekturen in Pass und DFM-Regeln: der Lernzyklus ist geschlossen	Wochen / dreistellige €

7. Zubehör: die zweite Achse

Boxen, Markisen, Starlink, LED-Bars, Solarpanels. Der Schlüssel ist die standardisierte Plattformschnittstelle (T-Nut, Raster): das Zubehör ist an die Schnittstelle gebunden, nicht an das Fahrzeug — einmal konstruiert, passt es allen. Zubehörpas: Befestigung, Volumen + Ausschlusszonen, Masse und Schwerpunkt (der Validator rechnet die Gesamtlast der Konfiguration), Kompatibilität. Für Kaufprodukte wird nur der Adapter konstruiert — dieselbe Pipeline. Digitalisierung: STEP vom Hersteller (wird Händlern gegeben) oder Scan — Photogrammetrie für die Hülle (± 1 –2 mm, Glanz mattieren) + Messschieber für die Befestigung ($\pm 0,05$); Genauigkeitsfeld im Pass, Ungenauigkeit wird konstruktiv kompensiert (Langlöcher). Die gedruckte Anprobe fängt Digitalisierungsfehler für 2 €. Eine Bibliothek der Pässe populären Zubehörs ist ein akkumulierender Vermögenswert.

8. Karriereleiter: Aufstieg nach Ergebnissen in den Logs

Stufe	Zugang	Übergang
Neuling	Testaufgaben und einfache Aufgaben, nur VDI, im Stücklohn	Eintritt frei
Geprüft	Direktaufträge, höherer Satz	3+ angenommene Arbeiten
Senior	Komplexe Baugruppen, Git-Zugang, Projektvergütung	stabile Arbeitsqualität + Kommunikation
Lead / Reviewer	Erstabnahme, Mentoring, Lastenhefte; löst den Bus-Faktor	separate Auswahl nach Denkqualität in den Kommentaren (Peter-Prinzip: der beste Zeichner ist nicht der beste Reviewer)

Eine gelungene Arbeit ist Rauschen, ein Muster über 4-6 Aufgaben ist ein Signal. Das System protokolliert: Qualität der Rückfragen zum Lastenheft, Iterationen bis Grün, Verhalten bei Misserfolg, ökonomisches Gespür. Die Beförderung wird berechnet, nicht verliehen — Transparenz der Regeln wirkt auf den Märkten der GUS und Asiens stärker als Prämien. Das finale Vertrauen (Kernzugang) ist eine persönliche Entscheidung.

9. Rekrutierung in verschiedenen Ländern: Märkte und Sätze

Markt	€/Std.	Rolle
Indien	4-10	Hauptpool: Tiefe, Englisch, Blechbüros (Pune/Chennai/Bangalore), UTC+5:30. Ping 100-160 ms — mit Probesitzung filtern.
Usbekistan / Kirgisistan	5-12	Russischsprachige Schiene: GOST-Schule, Loyalität; hieraus wachsen Festangestellte und Leads. Ping 60-90 ms — VDI komfortabel.
Pakistan	3-7	Zusätzliche Hände strikt im Stücklohn (Strom/Internet Risiko des Ausführenden; kein PayPal).
Vietnam	5-12	Zurückstellen (Sprache, UTC+7, Ping 180-250); strategisch bei asiatischer Fertigungsausrichtung.
Russland	—	Direkt nein (Zahlungen, Compliance); legal — Relokanten (Armenien/Georgien/Kasachstan) oder Einzelunternehmen im Drittland; Kasm über HTTPS funktioniert bei jeder Sperre.

Einstellungsmodell: entfernte Konstruktionsingenieure im Stücklohn (Bezahlung je Teil, das den Validator passiert hat) mit Wachstum bis zur dauerhaften Zusammenarbeit. Trichter: 100 Einladungen → 30 Rückmeldungen → 15 Tests → 8 abgegeben → 3-4 bestanden → 1-2 auf die Reservebank; Preis eines Festangestellten ~300-500 €. Kanäle: Upwork (aktive Einladungen zum bezahlten Test), hh.uz, GrabCAD, Hochschullehrstühle, Empfehlungen. Angebotsmagnete: bezahlter Test; keine Software und kein starker PC nötig (Browsersitzung — zugleich Legalität und Zugang zum Pool ohne Hardware); automatische Abnahme nach veröffentlichten Regeln; transparente Leiter; kontinuierlicher Auftragsfluss. Stufenweises Aussieben: Filterfrage → bezahlter Test mit Autoabnahme (Referenz Bracket_L) → persönlich betrachten Sie 3-4 von hundert. Der Test enthält eine 10-minütige VDI-Probesitzung (misst den realen Kanal des Kandidaten).

10. Lebenszyklus: zwei Achsen

Konstruktion (status)	Lager / Verkauf (sales_status)
Draft — in Arbeit, nach außen unsichtbar; In Review — Validator grün, PR offen; Released — Signatur + Tag + Rev, einziger fertigbarer Status; Superseded — neue Revision erschienen (automatisch), wird dauerhaft aufbewahrt; Obsolete — untauglich, kein Nachfolger, die CI blockiert die Fertigung; On Hold — ohne Entscheidung gestoppt	Not for Sale — Prototypen (PROTO), Piloten; Active — wird gefertigt und verkauft; Run-out — superseded, Restbestand > 0: verkaufen, nicht nachbestellen, der Shopify-Webhook schaltet bei null den Nachfolger frei; Discontinued — eingestellt, die Produktseite lebt dauerhaft (Garantie, QR-Anleitungen); Service Only — auf Anfrage als Ersatzteil zum Verkauf

Die Ersetzungspolitik A→B wird beim Release gewählt: run-out / sofort (Defekt) / parallel (FFF: gleiche Passung und Funktion = Revision, keine neue SKU). Achse 1 bewegt die CI über Git-Ereignisse; Achse 2 der Shopify-Webhook plus zwei manuelle Entscheidungen.

11. Änderung von Teilen: wo es teuer wird

Erstellen ist billig; Ändern ist teuer proportional zu den Abhängigkeiten. Drei Schweregrade: innerhalb vorhandener Parameter (kostenlos) / neuer Parameter (Generatoränderung — Engpass der Qualifikation) / Bruch der Topologie (= neues Teil; Regel «mehr als N Zeilen — neue Vorlage» gegen Krücken). Die Kaskade (Abwicklung, Masse und Last der Konfigurationen, GLB, Anleitung, BOM) baut die CI selbst neu; als Managementaufgaben bleiben: Unantastbarkeit der Schnittstelle (Änderung = «Version 2 der Plattform»), ausdrückliche Ersetzungspolitik im Lager, Verbot von Umgehungen an params vorbei (STEP an Blechcon nur aus der CI, technisch erzwungen). Dazu: geometrischer Diff im PR (Hervorhebung der Änderungen alt/neu im STEP), Versionierung der Bibliotheksbausteine mit expliziter Migration, ein Gate für Änderungen an released (Issue mit Begründung + Freigabe des Inhabers).

12. ISO 9001

Das System ist ein ausführbares QMS: 8.3 (Eingaben = Pass + Lastenheft, Ausgaben = CI-Artefakte, Verifizierung = Validator, Validierung = Prototypen, Änderungen = PR + Revisionen); 8.5.2 Rückverfolgbarkeit (Teil → Tag → Autor → Reviewer → Auftrag); 7.5 Dokumentation = Git; 10.2 CAPA (Reklamation → Issue → Validatorregel für immer). Freelancer = externe Anbieter (8.4): eine Seite mit Kriterien für Zulassung, Bewertung und Ausschluss (die Daten sammelt Gitea), Kontrollumfang an die Leiter gekoppelt (Risikoorientierung), Kompetenz (7.2) = Wettbewerbshistorie, für den Reviewer Checkliste und persönliche Unterschrift. DSGVO: Verarbeitung von Freelancer-Daten (Konten, Logs) — Vertragspunkt, Eintrag im Verzeichnis, Aufbewahrungsfrist der Logs. Zertifizieren, wenn der Markt danach fragt (B2B/OEM): das Fundament ist ISO-ready, Formalisierung 2–3 Monate, ~3–6 T€. Regel: dokumentiert wird, was ausgeführt oder genutzt wird — sonst Qualitätstheater.

13. Physische Prüfungen und ISO 11154

Ein durch externes Audit geschlossener blinder Fleck: das Konstruktionssystem ersetzt keine Prüfung des fertigen Produkts. Selbst wenn ein abnehmbarer Dachträger rechtlich als Ladung eingestuft wird (§ 22a StVZO — beim Anwalt prüfen, nicht auf Treu und Glauben annehmen) und keine ABE oder kein Teilegutachten verlangt, verwandelt das Fehlen dokumentierter dynamischer Prüfungen nach ISO 11154 (einschließlich City Crash, Beschleunigungen bis ~16 g) im Schadensfall die Produkthaftung in ein persönliches Risiko der Geschäftsführung und verschließt den Zugang zu B2B-Händlernetzen. Maßnahmen: (1) den aktuellen Prüfstatus des Produkts klären; (2) falls keiner vorliegt, eine eigene Prüfspur: Referenzprüfungen der Grundkonstruktion in einem akkreditierten Labor (Budget im vierstelligen Bereich, das ist der Eintrittspreis in B2B); (3) die FFF-Regel um den Punkt erweitern «eine Änderung, die die Tragfähigkeit betrifft, erfordert eine Nachprüfung» — der Validator markiert solche Änderungen automatisch; (4) die Rückverfolgbarkeit «welche Revision geprüft wurde» liefert das System bereits kostenlos (Git-Tag + Prüfprotokoll als Release-Artefakt).

Aktueller Stand und Rechenbasis

Das Unternehmen hat bereits Festigkeits- (statische) und Modalanalysen der Konstruktion durchgeführt; eine aerodynamische ist geplant. Abgrenzung: Berechnung (FEM) und physische Prüfung nach ISO 11154 sind rechtlich nicht austauschbar — die Berechnung sagt «sollte halten», das Prüfprotokoll dokumentiert «hat gehalten»; für Produkthaftung und B2B braucht es das physische

Protokoll, die Berechnungen sind eine starke Ergänzung, die die Zertifizierung verbilligt (weniger Iterationen im Labor). Modalanalyse: Eigenfrequenzen gegen Wirbelablösefrequenzen bei 100–130 km/h (Brummen, Resonanz, Ermüdung) — wird zur Validatorregel «erste Eigenfrequenz des Beins $\geq$ Schwelle» (Schwelle nach Strouhal aus Geschwindigkeit und Profilgröße; CalculiX beherrscht Modalanalyse serienmäßig). Aerodynamik pragmatisch: Windlasten als Eingangskräfte der Festigkeitsrechnung — Normformeln ohne CFD; Ablösefrequenzen nach Strouhal; vollständiges CFD (OpenFOAM) einmalig für die Windschildform und das Brummen, außerhalb der Pipeline. Die vorhandenen Berichte sind Kalibrierreferenzen der CalculiX-Pipeline: dasselbe Teil durchrechnen, Frequenzen und Spannungen abgleichen — der Validator entsteht geprüft (Logik der metrologischen Referenz Bracket_L, aber für die Physik).

14. Offene Community-Plattform

Druckbares Community-Zubehör (Halter, Clips) macht die Plattform «klebrig» und dient als Nachfrageaufklärung (ein populäres Modell → Aufgabe für eine Metallversion). Veröffentlicht wird eine offene Schnittstellenspezifikation (mit dem Messschieber nachmessbar; wer zuerst veröffentlicht, wird zum Standard; die Steuerung bleibt bei Ihnen, Modell USB); öffentliches Repository + STL; automatische Passprüfung durch dieselbe Pipeline; Kategorie Community im Konfigurator. Schutz (Produkthaftung): CC BY-NC-SA + Disclaimer (Sie sind Hoster, nicht Hersteller), automatische Moderation der Kategorien (Tragendes bei Geschwindigkeit auf eigenes Risiko oder nicht veröffentlicht), Schaufenstergrenze «Metall geprüft / Community auf eigene Gefahr». In den Regeln das Recht von Airholder auf eine Metallversion nach Motiven (mit Nennung des Autors). Die Community ist die nullte Stufe der Karriereleiter. Die Web-Visualisierung offenbart kein geistiges Eigentum: in den Browser geht eine vergrößerte GLB («ein Foto der Uhr, keine Zeichnung»); die Dezimierung ist ein CI-Parameter; exakte Geometrie verlässt den Server auf dem einzigen Weg CI → Blechcon. Gegen das Vermessen eines gekauften Produkts schützt nichts — der reale Schutz ist die Geschwindigkeit der Pipeline, die Pässe, die Lösungsbibliothek, der Validator, die Reservebank, der Standard.

15. Die Rolle von FreeCAD und des Ingenieurs: wer was tut

Die Position wurde nach den Praxisergebnissen präzisiert. Der Ingenieur verlässt das System nicht — er bleibt der Autor der Form. Ein vollwertiges CAD schreiben wir nicht: das ist weder unser Produkt noch unsere Expertise. Aber neue Teile müssen entstehen, und sie müssen von Hand entstehen — in FreeCAD. Das Werkzeug recada.py ersetzt den Konstrukteur nicht, es nimmt ihm die Routine ab: Vermessung, Reverse, Prüfungen, Abwicklungen, Fertigungspaket.

Wer	Was er tut	Womit
Ingenieur	erdenkt und zeichnet das neue Teil, entscheidet über Form, Radien, Freischnitte	FreeCAD — von Hand, wie gewohnt
Codegenerator	verwandelt Gezeichnetes oder Rekonstruiertes in Code mit Parametern; der Code öffnet sich in FreeCAD als gewöhnliches Teil	recada.py → Python für FreeCAD
Werkzeug	Vermessung, Reverse, Strecken, Operationen, drei Abnahmekriterien, DXF, BOM, Paket für Blechcon	headless-Kern auf OCCT, Weboberfläche
Produkteditor	Dachträger als Parameter: Höhe, Position, Länge; die Beine werden neu erzeugt	Arbeitsfenster

Warum FreeCAD bleibt und nicht verschwindet

Die Erfahrung zeigt: in der Automatisierungskette wird FreeCAD nicht gebraucht und stört sogar — sein OCCT 7.8 führt Operationen nicht aus, die auf 7.9 in unserem Kern durchlaufen. Aber dort, wo die Hand des Konstrukteurs nötig ist — neue Form, Sonderknoten, Skizze für ein konkretes Fahrzeug — ist FreeCAD unersetzlich und kostenlos. Daher der Hybrid: FreeCAD zeichnet, das Werkzeug rechnet und prüft, der Codegenerator verbindet beide Richtungen.

Idee des Workbench: wie verbinden

Zwei Wege, in der Diskussion geprüft. Erstens ein vollwertiger FreeCAD-Workbench auf dessen Part-API: erfordert die Portierung des Codes von OCP auf eine andere API und stößt auf die Unterschiede der OCCT-Versionen (der Offset fiel bei uns genau dort aus). Zweitens ein hybrider Workbench: Knöpfe in FreeCAD speichern den markierten Körper in ein temporäres STEP, rufen das externe Werkzeug aus seiner Umgebung auf und importieren das Ergebnis zurück. Weder Portierung noch Versionskonflikt; die schwere Geometrie wird dort gerechnet, wo sie funktioniert. Empfohlen ist der zweite Weg. Gebaut werden sollte er, nachdem sich die Kernoperationen gesetzt haben — ein bewegliches Ziel in eine Oberfläche zu verpacken ist verfrüht.

VDI (Kasm/Guacamole) als Weg, externen Ausführenden FreeCAD im Browser zu geben, bleibt in Reserve — für das Szenario mit nicht vertrauenswürdigen Auftragnehmern aus dem Rekrutierungsabschnitt. Für ein vertrauenswürdiges Team genügen WireGuard und lokales FreeCAD

mit Codegenerator.

[Fortsetzung in Teil 3: Abschnitte 15a-15f]

RECADA / Airholder — vollständige Projektbeschreibung v50

Deutsche Übersetzung, Teil 3 — Abschnitte 15a bis 15l

15a. Reverse-Werkzeug recada.py — gebaut und geprüft

In zwei Septembertagen entstand ein funktionierendes Werkzeug, das ein Teil aus fremder Geometrie abnimmt und in eine geprüfte Fertigungsdatei verwandelt. Alles headless, ohne FreeCAD in der Kette: der OpenCASCADE-Kern ist als Bibliothek eingebunden (OCP), FreeCAD bleibt nur Betrachter. Eine Datei recada.py: Kommandozeile und Browseroberfläche.

Funktion	Was sie tut	Geprüft an
Vermessung	Dicke, Biegungen mit zwei Winkeln (Drehung und Diederwinkel), Radien, Bohrungen mit Achsen, Freischnitte	Defender LRDE-2BR1: $t = 3,0$, 7 Biegungen, 6 Bohrungen — bis auf die dritte Stelle reproduzierbar
Schalenaufteilung	virtueller Taster: Punkt – Normale $\times t$ trifft die Oberfläche → Schalenfläche; Paarbildung + Zweifärbung des Graphen	41 Flächen decken 93 % der Oberfläche; Seiten 21/20
Rekonstruktion	Kinematik auf allen Flächen des Körpers, Neuaufbau der Biegezone, Lochverschluss	Volumen 78 031,6 gegen 78 031,6 mm ³ Original
Strecken	Schnitt mit Halbraum entlang der Wandebene, Verschiebung, Prisma des Querschnitts; automatische Wahl der Schnittlage nach dem Kriterium «genau ein Querschnitt»	Wand +10 mm, +20 mm — geschlossen, fertigungstauglich
Operationen	Spiegeln, Bohren, Verschließen, Trimmen, Verrunden — mit Prüfung nach jeder	alle liefern geschlossene Solids
Abwicklung	DXF mit Biegelinien auf eigener Ebene	299 $\times$ 168 mm — deckungsgleich mit dem Portal
Baugruppe	Zerlegung des STEP in Teile mit Namen, Gruppierung gleicher, Stapel-Reverse	84,5 MB, 628 Objekte

Drei Abnahmekriterien — die Hauptschlussfolgerung

Es sind nicht eins und nicht zwei Kriterien, sondern drei, und das dritte wird durch die ersten nicht ersetzt. Das ist keine Theorie: jedes wurde durch einen Fehlschlag gefunden.

Kriterium	Womit geprüft	Was es fängt	Wie gefunden
Metrologie	point-sampling beidseitig, Abstand zur OBERFLÄCHE	Genauigkeit der Flächen	eigen
Geschlossenheit	freie Kanten = 0, Solids ≥ 1	Löcher, offene Schale	Ablehnung des Portals Blehcon
Fertigung	Hochladen ins Lieferantenportal	wird es als Blechteil erkannt	Praxis

Drei gefundene und behobene Fehler

1. Die Metrologie maß den Abstand zum Körper, nicht zur Oberfläche. `distToShape` zu einem Solid liefert null für Punkte innerhalb des Materials — ein «zu dickes» Teil bestand die Abnahme. Prüfung: eine Verschiebung um 0,3 mm ergab 74 % der Punkte in Toleranz, nach der Korrektur 29,8 %.
2. Die einseitige Stichprobe sah überschüssiges Material nicht: ein Kandidat, länger als die Referenz, ergab RMS 0 und 100 %. Die Abweichung wurde beidseitig gemacht.
3. Eine Prüfung der Geschlossenheit gab es gar nicht. Ein Teil kann geometrisch exakt und zugleich löchrig sein — die Metrologie fängt das nicht. Entdeckt durch die Ablehnung des Portals; jetzt in einer Zeile berechnet und sagt die Ablehnung vor dem Versand voraus.

Was nicht funktionierte und warum

Das Aufdicken der gesamten Schale (Offset der kompletten vernähten Schale) lässt Löcher und hängt von der OCCT-Version ab: auf 7.9 funktioniert es, auf 7.8 in FreeCAD scheitert es. Ersetzt durch flächenweises Aufdicken mit boolescher Vereinigung — für eine einzelne Fläche ist der Offset immer korrekt.

Flicken mit BRepFill in der Biegezone erzeugen eine Spline-Fläche — daran scheitert die Blechteilerkennung. Daraus die Regel: in der Biegezone nur kanonische Flächen bauen (Zylinder und ebene Sektoren), das Freischnittprofil nach Regel erzeugen, nicht flicken.

Die parametrische Drehung des Biegewinkels liefert einen geschlossenen Körper, besteht aber die Erkennung beim Lieferanten noch nicht: das Stirnprofil der Biegung mit Freischnitt muss neu erzeugt und nicht kopiert werden. Das ist die einzige offene Aufgabe der Kette — und sie ist bis ins Detail verstanden.

15b. Produkteditor: der Dachträger als Parameter, nicht als Geometrie

Der Kerngedanke aus der Arbeit: ein Dachträger hat keine beliebigen Freiheitsgrade. Er hat Produktparameter, und Position und Form der Teile sind deren Folgen. Der Editor verschiebt daher keine Teile, sondern erzeugt sie neu.

Das Szenario, um dessentwillen alles gebaut wurde

Der Träger ist konstruiert, zwischen Dach und Unterkante des Querträgers sind 20 mm — die Hand passt nicht durch. Entscheidung: anheben. Sind die Beine bereits in Code rekonstruiert, genügt eine Zahl: die Befestigungspunkte am Dach sind fixiert, die vertikale Wand jedes Beins wird um 20 mm gestreckt, die Abwicklungen werden neu berechnet, die DXF aktualisiert, die drei Abnahmekriterien erneut durchlaufen. Ein vollständiges Reverse des ganzen Systems ist nicht nötig.

Derselbe Kreislauf bedient die Aerodynamik: CFD sagt «Plattform 30 mm nach vorn, 10 mm höher» — das sind zwei Zahlen in product.json, keine Woche Handarbeit. Genau deshalb müssen gebogene Teile als Code mit Parametern leben.

Einheitliches Arbeitsfenster

Zone	Inhalt
Baum links	Tabelle wie eine Stückliste: Sichtbarkeit, Auswahl, Name, Anzahl der Exemplare, Status (Original / rekonstruiert / parametrisch), Rolle (Bein, Dachhalter, Plattform, Querträger, Befestigung, Karosserie). Namenssuche, automatische Rollenerkennung nach Geometrie
Szene	eigener Viewer auf three.js: Explosionsdarstellung, Transparenz, Schnitt, Messen, Ansichten, Isolieren. Ausgewählte hervorgehoben, rekonstruierte grün
Aktionen	für ein Original — «vollständiges Reverse» (gleiche Teile werden einmal gerechnet); für ein rekonstruiertes — Strecken, Bohrungen, Spiegeln, DXF
Produkt	Befestigungspunkte am Dach als Anker, «Plattform um N mm anheben» — Neuerzeugung aller Beine
Journal	project.json: jede Operation mit Parametern festgehalten. Wahrheit ist das Journal, Geometrie ist Artefakt

Fertigungspaket

Ein Knopf packt ein Archiv: step/ — eindeutige Teile, dxf/ — Abwicklungen mit Biegelinien, passport/ — Vermessung jedes Teils als JSON, BOM.csv — Stückliste mit Mengen, project.json — Operationsjournal. Genau das geht an Blechcon.

Was das für die Projektökonomie ändert

Das Reverse des Defender-Halters dauerte Minuten statt Tage. Vier gleiche Beine werden einmal gerechnet. Das Anheben der Plattform ist eine Zahl statt einer Neukonstruktion. Infrastruktur: EX44 für 59 € netto plus Storage Box — rund 63 € im Monat für ein System, das es in Standard-CAD nicht gibt: automatisches Blech-Reverse mit Prüfung der Fertigungstauglichkeit.

15c. Teilebibliothek und Server: ein System mit Rückkopplung

Der Server EX44 ist gekauft, die Bibliothek eigener STEP-Dateien wird aufgebaut. Zusammen verwandeln sie das Werkzeug aus einer Demonstration in ein System, in dem jedes geprüfte Teil das nächste genauer macht.

Was der Server bringt

Trennung von Kern und Oberfläche: der Kern ist eine Funktionsbibliothek mit Kommandozeile, die Oberfläche eine dünne Schicht über einer Aufgabenwarteschlange. Das Reverse wird eingereicht, die Seite blockiert nicht, Ergebnisse werden dauerhaft nach Dateihash gespeichert. Ohne das funktionieren weder ein zweiter Nutzer noch die Bibliothek. Kontinuierliche Regression: jeder Commit läuft nachts über die Bibliothek, morgens eine Zahl — wie viele Teile die drei Kriterien bestehen. Gemeinsamer Zugang: entfernte Ingenieure arbeiten über WireGuard mit einem Werkzeug und einem Regelsatz.

Was die Bibliothek bringt

Anwendung	Wie es funktioniert	Was es schließt
K-Faktor-Kalibrierung	unsere analytische Abwicklung gegen die Portalantwort je Teil (299 × 168 bestätigt)	das dritte Kriterium wird von der Handprüfung zur Datenquelle
Freischnittkatalog	Typen, Tiefen, Randbezug — von realen Teilen abgenommen	die offene Aufgabe der Biegezone: nach Regel erzeugen statt flicken
DFM-Regeln	Statistik der Radian, Winkel, Dicken, Durchmesser, Randabstände	dfm_blechcon.json aus der Praxis statt aus dem Kopf
Ähnlichkeitssuche	Fingerabdruck: t, Anzahl Biegungen, Winkelsatz, Bohrbild	ein bewährtes Teil strecken statt neu zu konstruieren
Fahrzeugpässe	Befestigungspunkte und Dachprofile als Bibliotheksdaten	Produkteditor ohne manuelle Koordinateneingabe
Hausstandard	ein neuer Konstrukteur sieht die üblichen Radian und Lösungen	Wissensweitergabe ohne mündliche Überlieferung

Grenze der Ehrlichkeit: die Bibliothek rät und warnt, entschieden wird vom Menschen. Sie sagt nicht, ob ein Teil die Last hält, und ersetzt keine Ingenieursentscheidung dort, wo ein neues Fahrzeug etwas verlangt, das es in der Praxis noch nicht gab.

Markt: Zoo als Orientierung

Zoo (zoo.dev, ehemals KittyCAD) ist das philosophisch nächste Projekt: ihre Sprache KCL ist lesbarer Text als Wahrheitsquelle, Versionierung in Git, der Agent Zookeeper schreibt STEP in Code um. Zwei Unterschiede halten unsere Nische frei: die Geometrie-Engine von Zoo ist cloudbasiert und amerikanisch — die 3D-Ansicht ist ein Videostream von ihren Servern, die Zeichnungen gehen zu ihnen; und sie haben keine Fertigungsabnahme für Blech — keine Metrologie, keine Geschlossenheit, keine Erkennung durch das Lieferantenportal.

Wie Zoo streamt und was daraus folgt

Die Engine läuft auf ihren Servern. Der Client rechnet nichts: Modellierbefehle gehen über WebSocket hinaus, zurück kommen Videobilder über WebRTC — wie ein Videoanruf. Gestreamt wird nur der 3D-Viewport; Oberfläche, Baum und Code-Editor leben im Client. Das ist nicht Kasm: Kasm streamt einen ganzen Desktop oder eine Anwendung über Bildschirmaufnahme, schwer und mit Latenz; Zoo hat Rendering und Codec direkt in die eigene Rust-Engine eingebaut, deshalb ist der Client leicht. Drei Gründe für ihre Wahl: die Engine nicht auf Plattformen portieren zu müssen, beliebig große Modelle auf schwachen Rechnern zu öffnen, und vor allem: die Geometrie verlässt ihre Infrastruktur nie, der Client sieht nur Pixel.

Für uns ist das eine unabhängige Bestätigung des Prinzips aus dem VDI-Abschnitt — «Pixel, nicht Geometrie». Unterschied in der Umsetzung: sie haben das Streaming in ihre Engine eingebaut, wir würden eine fertige Anwendung streamen. Kehrseite unseres aktuellen Schemas: der three.js-Viewer gibt ein trianguliertes Netz in den Browser — für ein vertrauenswürdiges Team über WireGuard

normal, für externe Auftragnehmer ein Leck, das Netz lässt sich speichern. Deshalb bleibt für Nichtvertrauenswürdige das Streaming von Pixeln.

Eigene Engine: Kern nein, Umgebung ja

Die «Engine» von Zoo besteht aus drei Schichten: Geometriekern (B-Rep, Boolesche Operationen, Verrundungen, Abwicklungen), Rendering und Streaming mit API. Den Kern schreiben sie seit Jahren mit Spezialisten und zweistelligen Millionenbeträgen — und im März 2026 schrieben sie noch, Kantenverrundungen blieben ihr Schmerz. Das ist der Preis eines Kerns. Wir brauchen keinen: OpenCASCADE ist dreißig Jahre Arbeit, kostenlos und offen, und darauf wurde gestern ein Teil rekonstruiert, das das Portal angenommen hat. Der Kern ist eine gelöste Aufgabe — von fremder Hand.

Realistisch bauen lassen sich die zwei oberen Schichten um OCCT herum. Rendering auf dem Server: unser three.js-Viewer in einem headless-Browser oder die eingebaute OCCT-Visualisierung. Streaming: Bild über WebRTC zum Client, Kamerabefehle zurück — fertige Bibliotheken, keine Forschung, Größenordnung Wochen. Das ist «Viewport-Streaming um OCCT», nicht «eigene Engine». Nötig ist es nur für externe Ausführende, denen man keine Geometrie geben darf. Zoo musste einen Kern schreiben, weil sie eine Plattform verkaufen. Wir verkaufen Dachträger — andere Ökonomie, andere Entscheidungen.

Der Kern direkt: OpenCASCADE ohne FreeCAD

Präzise Formulierung der Architektur: OpenCASCADE ist der Kern in FreeCAD. Wir rufen ihn direkt über die Bindung OCP auf, an FreeCAD vorbei — der Motor auf dem Prüfstand statt im Auto. Das ergibt headless-Betrieb auf Server und CI, Stapelverarbeitung der Bibliothek und eine eigene Kernversion (7.9, wo das Aufdicken funktioniert; in FreeCAD 7.8 scheitert es). FreeCAD bleibt das Cockpit des Ingenieurs. Eine Geometrie, zwei Zugänge: programmatisch für die Automatik, manuell für den Menschen. Praxis: eine FreeCAD-Version halten, deren OCCT zum Kern passt.

Der Kern wurde mit KI erstellt, durch die Realität geprüft und läuft ohne KI. Die KI schrieb den Code, arbeitet aber nicht in ihm: jede Funktion ist gewöhnliches deterministisches Python, ein Eingang liefert ein Ergebnis. Die drei Fehler fand nicht die KI — sie fanden Gegenprüfung, Negativtest und die Ablehnung des Portals. Daraus die Entwicklungsregel: Test vor Code (Spec-Driven Development auf das Werkzeug selbst angewandt), Bibliothek als Regression danach. Ein eigenes Modell zu trainieren ist weder nötig noch möglich — Dutzende Teile sind keine Stichprobe; die Bibliothek arbeitet als Nachschlagewerk. Fertige KI wird als Übersetzer von Absichten an die Kernoperationen angeschlossen, nicht als Quelle der Geometrie.

Viewer auf demselben Kern (Stand September 2026)

Werkzeug	Was es ist	Wo einsetzen
OCP.wasm + Yet Another CAD Viewer	OpenCascade und OCP nach WebAssembly über Pyodide kompiliert; build123d läuft vollständig im Browser. Dieselbe OCP wie in recada.py	Werkzeug beim Kunden im Browser ohne Server — nichts verlässt den Rechner; Antwort auf das Netz-Leck bei externen Auftragnehmern
OpenCascade.js	Portierung von OCCT nach JavaScript/WASM über Emscripten, nahezu native Geschwindigkeit, Mehrfädigkeit, schlanke Builds; darauf CascadeStudio	Browseransicht und leichte Operationen ohne B-Rep-Leck
OCCT WebGL Viewer	eigene OCCT-Visualisierung in WASM — öffnet BREP im Browser, benötigt WebGL 2.0	zeigen «genau wie im Kern» ohne FreeCAD
Mayo	Desktop-Viewer und Konverter auf Qt + OCCT: STEP, IGES, BREP, Export glTF	kostenloser Desktop-STEP-Viewer auf unserem Kern
OCP CAD Viewer (VS Code)	zeigt OCP-/build123d-Objekte direkt aus Python, ohne STEP	Debugging des Kerns bei der Entwicklung
Unser three.js-Viewer	Netz vom Server, Auswahl, Explosion, Schnitt, Messen	schnelle Oberfläche für das vertrauenswürdige Team

Strategisch am interessantesten ist OCP.wasm: unsere Bindung ist dieselbe, also kann recada.py beim Kunden im Browser ohne Server laufen — der umgekehrte Ansatz zu Zoo, wo alles in die Cloud geht.

Entscheidung: Cascade-Viewer auf dem Server, nach außen nur Pixel

Getroffen nach dem Erwerb des EX44. Kern und Rendering von OCCT laufen auf dem Server; die OCCT-Visualisierung (V3d, AIS) ist aus Python über dieselbe OCP erreichbar — ohne C++ und ohne WASM. Der Browser erhält Bilder und sendet Klicks; der Server bestimmt selbst, welche Fläche, Kante oder Ecke getroffen wurde — die Auswahl ist exakt nach B-Rep, mit Fang auf Bohrungsmitten und Kantenmitten. Die Geometrie verlässt den Server nicht: weder Netz noch STEP. Die verworfene Alternative — Hybrid mit lokaler Drehung am Netz — ist schneller und billiger für den Server, aber das Netz gelangt in den Browser und kann gespeichert werden; für den Datenschutz inakzeptabel.

Reicht der EX44

Keine Grafikkarte — das Rendering läuft in Software über Mesa auf der CPU. Eine CAD-Szene ist einfach: Hunderttausende Dreiecke, zweistellige Bildraten je Kern. Dazu die Kodierung — noch ein Kern. Macht zwei Threads je aktiver Sitzung; aktiv ist sie nur, solange der Mensch dreht, beim Betrachten ist die Last null. Abzüglich Vermessung und Infrastruktur: fünf bis sechs gleichzeitig Drehende. Für das Team mit Reserve, für externe Auftragnehmer einige gleichzeitig. Was die Reserve ohne Hardware erweitert: während der Drehung ein Bild niedriger Auflösung, nach dem Stopp ein scharfes; Begrenzung der Bildrate je Sitzung. Wann andere Hardware nötig wird: ab zehn gleichzeitig Aktiven — GEX-Server von Hetzner mit GPU, Preis drei- bis vierfach; Entscheidung für später.

Datenschutz — nicht nur Streaming

Maßnahme	Was sie schließt
Alle Bibliotheken (three.js, pako u. a.) auf dem eigenen Server	derzeit lädt der Viewer sie von externen CDN — die Tatsache, dass ein Teil geöffnet wurde, ist für Dritte über die Anfragen sichtbar
Zugang nur über WireGuard, persönlicher Schlüssel je Person	in den Logs ist sichtbar, wer wann was geöffnet hat
Keine Download-Punkte für externe Rollen	Export von STEP und DXF nur für Vertrauenswürdige
Wasserzeichen mit Sitzungsname im Bild	einen Screenshot kann man nicht verbieten, aber zuordnen
Rendering auf Anforderung, kein Stream	Serverlast und Datenvolumen

Vorab anerkannte Grenze: gegen ein Foto des Bildschirms schützt nichts. Pixel schließen das Leck des Systems — ganze Modelle, Parameter, der Bibliothek. Ein einzelnes Teil lässt sich immer nachzeichnen: «ein Teil kann man nachbauen, ein System nicht».

Kernversionen und Format an den Grenzen

OCCT 7.8 in FreeCAD 1.1, 7.9.3 in unserem Kern, 8.0 erschien im Mai 2026, aktuell 8.0.1 vom 30. Juli 2026 (größter Schritt seit Jahren: STEP-Lesen bis 75 % schneller, threadsicheres Lesen und Schreiben von STEP, Graphentopologie BRepGraph, beschleunigte Extrema — alles für Server und Bibliothek; neue Blechoperationen gibt es nicht). FreeCAD auf der Acht — realistisch 2027. Regel: die Kernversion des Werkzeugs wird fixiert, in project.json geschrieben und nur als Ganzes gewechselt — nach einem Regressionslauf der Bibliothek davor und danach. Kern und FreeCAD müssen nie übereinstimmen; die Architektur überlebt das, weil der Codegenerator fertige Körper einbettet, statt sie in FreeCAD neu zu rechnen.

Format an den Grenzen ist immer STEP. BREP ist das interne Kernformat, seine Version ändert sich mit dem Major-Release, und 7.9 liest möglicherweise nicht, was 8 geschrieben hat. STEP ist ISO 10303, wird von jeder Version in beide Richtungen gelesen, von Inventor, SolidWorks, vom Lieferantenportal. Regel in einer Zeile: BREP innerhalb des Prozesses, STEP an jeder Grenze — zwischen Werkzeug und FreeCAD, zwischen Server und Auftragnehmer, zwischen heutigem und morgigem Kern. Der Preis — Dateigröße (800 KB gegen 120 bei BREP) — ist tragbar.

STEP mit Namen: verpflichtende Exportanforderung

STEP speichert Teilennamen und Knoten, den Baugruppenbaum mit Positionen, die Exemplarzahl, Farben und Ebenen — sofern über XCAF geschrieben wird (STEPCAFControl_Writer mit Dokument). Am Eingang liest das Werkzeug genau so: die Namen Bracket_F_v2.1, wind_side_v2.1 kommen mit Positionen aus der Baugruppe. Am Ausgang schreibt es derzeit vereinfacht (STEPControl_Writer) und

verliert die Namen — der Baum wird zu einem namenlosen Compound. Das ist ein Fehler des Werkzeugs, nicht des Formats; behoben durch den Austausch des Writers: Name, Position und Farbe je Teil. Die Farbe kodiert den Status: grün = rekonstruiert.

STEP überträgt	STEP überträgt nicht — geht in project.json
Teilenamen und Knoten	Vermessung: Dicke, Biegungen, Bohrungen
Baugruppenbaum mit Positionen	Reverse-Status, Rolle
Exemplarzahl	Operationsjournal
Farben und Ebenen	Befestigungspunkte

Das Paar «STEP + project.json» ist keine Redundanz, sondern Arbeitsteilung: STEP trägt, was alle verstehen, JSON, was nur wir verstehen; verbunden über den Teilnamen. Namenlose Exemplare der Form =>[0:1:1:5] in der Baugruppe sind kein Leseverlust, sondern unbenannte Komponenten im Ausgangs-CAD; heilbar durch Benennung vor dem Export aus Inventor.

RECADA / Airholder — vollständige Projektbeschreibung v50

Deutsche Übersetzung, Teil 4 — Abschnitte 15d bis 15l

15d. Montageanleitung als Produktartefakt

Analysiert wurde die geltende Airholder-Anleitung für den Defender 110 (Montageanleitung, 15 Seiten). Ergebnis: rund zwei Drittel des Inhalts sind direkte Folgen aus product.json und BOM, ein Drittel sind feste Regeln und Rechtstexte. Die Anleitung wird zum siebten Artefakt des Fertigungspakets neben STEP, DXF, Pässen, BOM, Zeichnungen und project.json — mit dem Käufer als Adressat statt Blechcon.

Seite / Block	Quelle	Modus
Stückliste Komplettsatz: 16 Positionen mit Mengen (S13A ×6, B13A ×14, M8×14 DIN 912 ×56, Halterung 1/2 ×8+8 ...)	BOM; Schraubenzahl aus der Zahl der Verbindungen	auto
Piktogramme der Teile, Ansichten von oben und Isometrie, Explosionsdarstellung	HLR-Projektionen vom Server im Strichstil	auto
Positionen der Nutensteine: 14, 54, 44, 344, 15, 815 mm in den Profilen A/B/C/D	berechnet aus den Positionen der Querträger und Halter entlang des Profils	auto
Masse des Trägers 48 kg im Lasthinweis	BOM mit Dichten	auto
Schritte 1–13: Verbindung der Profilhälften D, Rahmen, Ecken, Halter, Montage aufs Dach; Befestigungsverweise	Reihenfolge aus dem Verbindungsgraphen; Text aus Vorlagen je Verbindungstyp	Vorlage + Zahlen
Rechtstexte, Warnhinweise, Regel zur B-Profil-Ausrichtung, Montage auf ebener Fläche, «richtig / falsch» 10.1 und 10.2, zwei Wege zum Einsetzen des Nutensteins	feste Blöcke — Konstruktionswissen, als Produktregeln gespeichert	manuell

Fund für das Produktmodell: «Completed connector» = Halterung 1 + Halterung 2. Das Bein besteht aus zwei gebogenen Teilen — Dachhalter und Ständer — plus Gummiauflage und Schraube M6x22 in die Dachrinne. Im Produkteditor ist das Bein eine Baugruppe aus zwei Teilen mit gemeinsamer Schnittstelle; das Anheben der Plattform streckt nur den Ständer, der Dachhalter bleibt unberührt. Das bestätigt die im Werkzeug umgesetzte Mechanik der Fixpunkte.

Werkzeuge

Aufgabe	Werkzeug	Stand
Strichansichten wie in der Anleitung	HLR aus OCCT über OCP — Projektionen mit sichtbaren und verdeckten Kanten, Vektor	Kern vorhanden, Modul ungenutzt
Raster mit Schatten bei Bedarf	OCCT-Visualisierung headless / three.js im headless-Browser	Reserve
Vektor im PDF	reportlab (baut dieses Dokument) + svgwrite	vorhanden
Verbindungsgraph, Schrittreihenfolge	networkx + Kontakte über BRepExtrema	vorhanden
Textvorlagen, DE/EN	Jinja2 + Wörterbücher	hinzufügen
EAN-Barcode	python-barcode	hinzufügen
DIN-Befestigungskatalog	eigene Tabelle	hinzufügen

Vier Schichten, die im System fehlen: (1) Verbindungsgraph — welches Teil welches berührt, aus der Geometrie; daraus die Montagereihenfolge von unten nach oben. (2) Verbindungsmodell in product.json: Teilepaar → Befestigung, Anzahl, Position entlang des Profils, Moment, Schraubensicherung; daraus die 56 Schrauben, die Nutensteinpositionen und die Verweise — das inhaltsreichste Stück, es wird auch für Abstands- und Kollisionsprüfungen gebraucht. (3) Serverseitiges Rendering nach Ansichtsvorlagen. (4) Dokumentgenerator: Seitenvorlage mit Rahmen, feste Blöcke, Datenersetzung, zwei Sprachen. Server: eine HLR-Projektion der Baugruppe dauert Sekunden je Kern; die vollständige Anleitung mit rund fünfzig Bildern ein bis zwei Minuten; Speicher im dreistelligen MB-Bereich; keine GPU nötig. Die Anleitung wird bei jeder Produktänderung in der CI

neu gebaut wie das DXF. Was das über die Handarbeit hinaus bringt: Querträgerzahl geändert — Nutensteinpositionen, Schraubenmengen und Masse werden neu berechnet; der Ranger bekommt seine Anleitung am Tag der fertigen Geometrie; Widersprüche zwischen Stückliste und Schemata gibt es konstruktionsbedingt nicht.

15e. Zwei Welten: B-Rep und Netz

Präzisierung aus der Praxis: das Fahrzeug liegt nicht als STEP vor, sondern als FBX von Fremdmodellierern, oft mit falschem Maßstab und falschen Proportionen. Derzeit wird der Träger in Blender aufs Auto gesetzt: das STEP des Trägers wird nach GLB konvertiert, der Maßstab des Fahrzeugs von Hand angepasst. Das System zerfällt in zwei Welten unterschiedlicher Genauigkeit, die nicht vermischt werden dürfen.

	Welt B-Rep	Welt Netz
Objekt	Träger, Teile, Befestigung	Fahrzeug, Umgebung
Format	STEP (an Grenzen), BREP (intern)	FBX am Eingang, GLB an der Grenze
Genauigkeit	fertigungstauglich: 0,05 mm, drei Abnahmekriterien	visuell: Zentimeter, kalibriert
Zweck	Fertigung, Reverse, Parametrik	Bild, Anprobe, grobe Abstände, Konfigurator
Werkzeug	recada.py auf OCCT	Blender (skriptbar, MCP vorhanden)

Folgen: Befestigungspunkte dürfen nicht vom Fahrzeugmodell genommen werden — es ist ungenau. Quelle sind die eigenen Halter, an der realen Karosserie erprobt: beim Defender ist die Geometrie der Halterung 1 zugleich der Punktepasse, vom Metall abgenommen. Die Logik dreht sich um: nicht «das Fahrzeug bestimmt den Träger», sondern «der bereits stehende Träger bestimmt die Koordinaten des Fahrzeugs». Das FBX wird zur visuellen Kontrolle untergelegt. Der Maßstab wird an Werksmaßen kalibriert — Radstand, Breite, Dachhöhe; ein Koeffizient wird im Fahrzeugpass gespeichert statt bei jeder Anprobe nach Augenmaß angepasst. Konvertierung STEP → GLB ohne Blender: OCCT hat einen eigenen glTF-Writer (RWGlTF_CafWriter, aus OCP erreichbar), der Teilnamen über XCAF erhält. Blender bleibt für das, was OCCT nicht kann: FBX lesen, fotorealistische Renderings für die Website, das Fahrzeugbild in der Anleitung — skriptbar über das vorhandene Blender-MCP. Formatgrenze: STEP innerhalb der B-Rep-Welt und an der Fertigungsgrenze; GLB an der Grenze zu Visualisierung und Unity-Konfigurator. Datenschutz: fremde FBX-Fahrzeugmodelle sind oft auf einen Nutzer lizenziert und verbieten Weitergabe — auf dem Server liegen sie in derselben geschützten Zone wie die Zeichnungen, nach außen gehen nur Renderings oder vereinfachte Netze.

15f. Daten: Quellen, Kataloge, einheitliche Struktur

In der Praxis erwiesen sich sechs Quellen als relevant, und sie brauchen einen Ort. Datenmodell: zwei Kataloge — Fahrzeuge und Zubehör — mit einheitlicher Struktur «Objekt → Darstellungen mit Genauigkeitsklasse». Eigene Teile in derselben Struktur: STEP, Zeichnung, Portalantwort, Vermessungspass. Eine Bibliothek für alles.

Quelle	Was sie liefert	Einschränkung
Aufbaurichtlinien der Hersteller (VW Aufbaurichtlinien, Ford Body and Equipment Mounting Manual, Stellantis Conversion Guidelines)	was mit der Karosserie erlaubt ist: Dachlast statisch und dynamisch, tragende Struktur (B-Säulen, Dachspriegel), Befestigungszonen mit Koordinaten, Verbotszonen (Airbags, Sensoren, Leitungen), Schwerpunkt, Momente und Schraubenklassen, Korrosionsschutzanforderungen, Elektrikpunkte	Zugang über Portale für Aufbauhersteller nach Registrierung; versioniert nach Modelljahren
OEM-CAD (VW, Ford, Stellantis)	exakte Karosseriegeometrie, Serienpunkte, Werkskoordinatensystem	NDA, Lizenz; JT/CATIA einmal nach STEP konvertieren (Inventor öffnet JT)
2D-Zeichnungen der Hersteller	Nennmaße und Toleranzen der Serienpunkte — was für alle Fahrzeuge garantiert wird; einzige offizielle Geometrie für ältere Modelle (Defender)	DXF liest ezdxf; PDF von Hand, es sind ein Dutzend Zahlen
STL-Scan	reales Fahrzeug oder Zubehör wie es ist: Fertigungstoleranzen, Anbauteile, Modifikationen	Registrierung ins gemeinsame Koordinatensystem, Rauschen und Löcher bereinigen
Halter in Metall	bestätigte Befestigungspunkte	nur dort, wo wir bereits stehen
CAD des Zubehörs vom Lieferanten	STEP, Masse, Befestigungsschnittstelle	nicht alle geben es heraus
FBX	Bild, Konfigurator	Maßstab ungefähr, wird an Werksmaßen kalibriert

Aufbaurichtlinien stehen aus einem Grund über dem OEM-CAD: CAD zeigt die Geometrie, die Richtlinien sagen, was damit erlaubt ist. Die Hälfte der Validatorregeln auf Produktebene erhält eine offizielle Quelle mit Seitenzahl statt unserer Schätzung. Ein Verweis auf die Richtlinien in der Produktdokumentation ist eine Position im Gewährleistungsstreit und in der Zertifizierung: «der Halter steht in einer vom Hersteller freigegebenen Zone, die Last liegt im angegebenen Bereich». Für den Käufer bedeutet es den Erhalt der Werksgarantie auf die Karosserie; ISO 11154 plus Richtlinienkonformität ist das Paket, das ein Händler akzeptiert. Extraktion: fertige KI zieht die Abschnitte zu Dach, Säulen, Lasten und Zonen als JSON mit Seitenangabe heraus, der Ingenieur prüft die Zahlen; jede Passregel speichert Dokument, Version, Seite.

Einheitliche Struktur: Objekt → Darstellungen

Darstellung	Fahrzeug	Zubehör	Eigenes Teil	Rolle
CAD	OEM	STEP des Lieferanten	STEP aus Reverse oder FreeCAD	Nennwert
2D-Zeichnung	Werkszeichnung	Lochbildschema	Zeichnung aus HLR	Nennwert mit Toleranzen
Scan	reales Dach	reales Zelt, Kanister	—	Realität
Visuelles Modell	FBX	GLB	GLB aus OCCT	Bild
Bestätigung	Halter in Metall	—	Portalantwort, Vermessungspass	Tatsache

Gemeinsames Koordinatensystem ist das Werkssystem: OEM-Dateien kommen im Standardsystem des Fahrzeugs — X entlang, Nullpunkt vom Hersteller. Der Scan wird darauf ausgerichtet (ICP, Open3D), das FBX darauf skaliert, der Pass darin geschrieben. Wo kein OEM-CAD vorliegt, gibt der Scan das System vor. Einmal ausgerichtet — alle Prüfungen in einem Raum.

Was Scans eröffnen — das vierte Kriterium

Sitzprüfung: die Fläche des Halters, die auf dem Dach aufliegt, wird mit derselben Metrologie gegen den Scan verglichen — Punkte der Fläche → Abstand zum Netz. Ergebnis ist eine Anlagekarte: wo es

aufliegt, wo es hängt, wo es ansteht. Das ist ein Montagekriterium, kein Fertigungskriterium; weder Blechcon noch das Portal liefern es. Abstände in der Realität: Querträger zum Dach, Bein zur Reling, Plattform zur Antenne — von B-Rep zum Scan, ohne FBX. Das Szenario «zwanzig Millimeter, die Hand passt nicht durch» wird am echten Dach geprüft. Scan gegen OEM-CAD ist selbst eine Information: die Streuung zwischen Fahrzeugen einer Serie zeigt, wie viel Verstellweg ein Halter haben muss. Die Langlöcher der Halterung 2 sind genau das.

Grenze: Reverse aus einem Scan in B-Rep ist eine Aufgabe anderer Größenordnung als aus STEP; für einen einfachen Halter machbar, für einen komplexen Guss nicht. Der Zubehörsan ist Quelle für Maße und Prüfungen, kein Fertigungsoriginal; ein Adapter für fremdes Zubehör wird in FreeCAD nach den Zahlen aus dem Scan konstruiert.

Option: Reverse aus STL

Status — Option, kein Plan. Unterscheidet sich grundsätzlich vom Reverse aus STEP: in STEP sind Ebenen und Zylinder bereits definiert, in STL nur Dreiecke — die Ebene muss gefunden und angepasst, der Biegezyylinder über die Krümmung erkannt, die Bohrung über den Rand rekonstruiert werden, jeder Schritt mit Scanfehler. Für Blech machbar: zwei Flächentypen — Ebenen und Zylinder — sind der beste Fall für die Erkennung (Segmentierung nach Normalen, RANSAC, Zylinderanpassung). Es entstehen dieselben Zahlen wie bei der Vermessung. Schlüsselentscheidung: nicht in B-Rep, sondern in Parameter rekonstruieren und die Geometrie vom Generator neu bauen lassen — angepasste Flächen müssen dann nicht zu einem Körper vernäht werden, den Körper baut der Generator, der bereits die drei Kriterien besteht. Deckt sich mit dem Prinzip «Wahrheit sind Parameter». Einschränkungen: Scanqualität (strukturiertes Licht 0,05–0,1 mm, Consumer-Geräte schlechter; Kanten verrunden, Bohrungen wachsen zu); Blech von beiden Seiten aufnehmen, sonst ist die Dicke eine Annahme; Guss, Tiefzug und Freiformflächen liegen außerhalb der Methode. Werkzeuge: Open3D, pyransac3d, scipy, danach unser Generator — alles offen. Wo es gebraucht wird: fremde Teile ohne CAD; Kundenteile für dessen Befestigung; und vor allem die Wareneingangsprüfung: Scan eines Teils von Blechcon gegen unser STEP mit derselben Metrologie. Reverse rückwärts — nicht rekonstruieren, sondern prüfen. Fünftes Kriterium: Abnahme der Lieferung.

Zubehör: die zweite Achse mit Daten

Die Befestigungsschnittstelle wird einmal für die Plattform beschrieben (Nutraster, Profilquerschnitt) und einmal für das Zubehör (in die T-Nut, auf die Airline-Schiene, an den Halter, Befestigungsmittel). Danach automatisch: das Zubehör erscheint im Konfigurator nur dort, wo es physisch passt; seine Masse geht in die Dachlast ein (Herstellerlast minus Träger minus Zubehör); sein Bauraum wird auf Kollisionen mit Haltern und Windabweiser geprüft; es kommt als eigener Schritt in die Anleitung; ein Zelt am vorderen Plattformrand ist ein eigener CFD-Lauf. Werkzeuge für Netze: OCCT liest STL (RWStl); für schwere Scans trimesh für Punkt-Netz-Abstände, Open3D für Registrierung und Bereinigung. Alles auf der CPU; Speicher für einen 300-MB-Scan einige GB während der Verarbeitung, der EX44 genügt.

Schutz: OEM-Dateien unter NDA sind das Sensibelste auf dem Server; ihr Abfluss ist keine Know-how-Frage, sondern ein Vertragsbruch mit dem Hersteller. Pixel nach außen, persönliche Schlüssel, Downloadverbot für Externe — genau dafür. OEM-Geometrie geht nie in Konfigurator und Renderings — dort nur Träger und FBX.

15g. Wettbewerber als Datenquelle

Wettbewerber veröffentlichen Fotos, Anleitungen-PDFs, Maße und Gewichte. All das ist nutzbar — legal, solange analysiert und nicht Geometrie kopiert wird. Ein Wettbewerber ist dasselbe Bibliotheksobjekt mit Darstellungen geringer Genauigkeit.

Was	Wie	Was es bringt
Struktureller Benchmark	aus Anleitungen-PDFs: Teilezahl, Befestigungspositionen, Schrittzahl, Montagezeit, Beintyp (ein oder zwei Teile), Ecklösung, genutzte Dachpunkte	Tabelle über Wettbewerber je Fahrzeug: wo wir einfacher, wo komplexer sind
Hinweise für den Pass	die Anleitung zeigt, wo der Wettbewerber befestigt: Dachrinne, Reling, Seriengewinde	Bestätigung der Punkte dort, wo OEM-Daten fehlen
Effizienzkennzahlen	Masse je m ² Plattform, Höhe über Dach, Masse je Befestigungspunkt — in denselben Einheiten, die unser System aus der BOM rechnet	Argument für die Produktseite und Ziel für den Konstrukteur (55 mm Airholder gegen 70 mm Front Runner)
Fotos	Photogrammetrie — grobes Netz	nur Abmessungen, Proportionen, Umströmung; nicht für Reverse
Extraktion aus PDF	fertige KI: PDF hinein, strukturiertes JSON heraus, Prüfung mit den Augen	Spezifikationen der Wettbewerber in der Datenbank, Vergleiche automatisch
DFM-Lehren	wie sie Freischnitte, Winkel, Profilstöße lösen	anschauen und lernen, nicht nachbauen
Patentrecherche	zu Dachbefestigungsmechanismen vor der Konstruktion	günstige Versicherung gegen Ansprüche

Grenze: öffentliche Dokumente dürfen analysiert werden; Geometrie, kennzeichnende Formen und Baugruppen unter Patent oder eingetragenen Geschmacksmuster nachzubauen nicht. Der Wert liegt nicht darin, ihren Träger zu wiederholen, sondern den eigenen schneller und genauer herauszubringen.

15h. Prüfungen auf Produktebene: Platzierung, Erreichbarkeit, Höhe, Einstellung

Die Produktebene ist keine Teilegeometrie, sondern der Träger auf einem konkreten Fahrzeug. Jede Frage «wohin darf man ihn setzen», «erreicht der Mensch das», «passt es ins Parkhaus» wird zu einer Validatorregel: Zahl, Datenquelle, Schwelle. Jede Einstellung ist ein Parameter in product.json, jeder Parameter zieht die Prüfungen erneut nach sich.

Im Fahrzeugpass stehen Zonen. Erlaubt: Dachrinne, Reling, Seriengewinde, mit zulässiger Punktlast vom Hersteller. Verboten: Schiebedach, Antenne, Haifischflosse, Sensoren der Assistenzsysteme, Kamera, Bremsleuchte, Schwenkbereich der Hecktür. Die Position entlang des Dachs wird automatisch gegen die Zonen geprüft: nach vorn verschoben — der Validator meldet, dass das vordere Bein in die Zone des Regensensors gerät. Quellen: OEM-CAD und Zeichnungen für die Zonen, der Scan für das reale Fahrzeug, Anleitungen der Wettbewerber als Bestätigung der Punkte.

Prüfung	Was berechnet wird	Schwelle / Quelle
Befestigungszonen	Beine und Halter innerhalb erlaubter, außerhalb verbotener Zonen	Fahrzeugpass (OEM, Zeichnungen, Scan)
Dachlast	Hersteller – Trägermasse – Zubehör; –50 % im Gelände	werksseitige zulässige Last; aus der Airholder-Anleitung
Punktlast	Massenverteilung auf die Befestigungspunkte	zulässig je Punkt vom Hersteller
Türen und Heckklappe	Öffnungsbogen gegen Beine und hinteren Plattformüberstand	Türkinematik aus OEM oder Messung; volles Öffnen zwingend
Höhe fürs Parkhaus	Fahrzeughöhe + Träger + höchstes Zubehör	1,85 / 1,90 / 2,00 / 2,10 m; im Konfigurator je Konfiguration anzeigen
Beladung vom Boden	Höhe der Plattformunterkante über Grund	Anthropometrie DIN 33402; bei Überschreitung Trittstufe als Zubehör
Erreichbarkeit	bis zur Plattformmitte von der Seite nach Breite und Höhe	DIN 33402
Abstand zum Dach	Querträgerunterkante ↔ Dach, Bein ↔ Reling, Plattform ↔ Antenne	B-Rep ↔ Scan oder OEM; Szenario «20 mm, Hand passt nicht»
Sicht	Blick durch die Heckscheibe, Rückfahrkamera, Antenne	Sichtstrahl aus der OEM-Spiegelposition
Aerodynamik	Windabweiser, Geräuschkanten, Ablösefrequenzen nach Strouhal an den Querträgern	Formeln im Validator; CFD einmalig
Schwerpunkt	Höhe des Schwerpunkts mit Dachlast und Zubehör	Grenzwert aus den Aufbaurichtlinien; bei hohen Kastenwagen mit Zelt der begrenzende Faktor
Dynamik ISO 11154	Stadt, Notbremsung, Seitenaufprall — rechnerisch	vor der Laborprüfung
Zubehörkompatibilität	Schnittstelle, Position, Masse, Kollisionen mit Haltern und Windabweiser	Zubehörkatalog

Jede Regel speichert ihre Quelle mit Verweis — Dokument, Version, Seite. Für Befestigungszonen, Dachlast, Punktlast, Schwerpunkt und Befestigung sind es die Aufbaurichtlinien; für Abstände und Sitz Scan und OEM-CAD; für Erreichbarkeit DIN 33402; für Dynamik ISO 11154. Im Konfigurator wird dem Käufer angezeigt: «entspricht den Herstelleranforderungen an Aufbauten».

Ohne Neukonstruktion einstellbar: Position entlang des Dachs innerhalb der erlaubten Zonen; Höhe über Dach; Raster und Anzahl der Querträger nach Ladung; Position des Zubehörs mit Kompatibilitätsprüfung; Vorhandensein und Ausrichtung des Windabweisers. Alles Parameter in product.json — geändert wird eine Zahl, Beine und Halter werden neu erzeugt, die Prüfungen laufen erneut, Anleitung und DXF werden neu gebaut.

15i. Produktdesign: was das System liefert

Alles Vorherige betraf den Weg eines Teils in die Fertigung. Dieser Abschnitt betrifft, wie der Träger selbst sein soll und wo das System dem Konstrukteur nicht Bequemlichkeit, sondern neue Lösungen gibt. Drei Dinge stützen sich auf bereits gesammelte Daten: Gemeinsamkeit der Teile zwischen Fahrzeugen, Verstellbereich aus der Scanstreuung, Montagezeit aus dem Verbindungsgraphen. Der Rest sind Konstruktionsregeln, jede eine Zeile im Validator.

Für den Käufer

Montagezeit als Projektkennzahl. Heute 2–3 Stunden. Jeder Anleitungsschritt sind Minuten; die Summe wird über den Verbindungsgraphen berechnet. Das System zeigt, wie vormontierte Baugruppen sie verkürzen: das Bein ab Werk komplett (Halterung 1 + 2 + Unterlage, vier Schrauben) statt zweier Teile beim Kunden. Ziel: eine Stunde. Ein Werkzeug für die ganze Baugruppe: heute M8 und M6, zwei Schlüssel. Alles auf eine Größe: kürzere Anleitung, weniger Fehler, ein Schlüssel im Karton. Validatorregel: Anzahl verschiedener Befestigungsgrößen — Zielwert 1, zulässig 2. Ladungsszenarien: Zelt, Fahrräder, Kajak, Kanister, Kisten — jedes mit eigenen Anforderungen an Querträgerraster, Befestigungspunkte und Massenverteilung. Der Käufer wählt «was ich transportiere», der Konfigurator schlägt Anzahl und Positionen der Querträger sowie Zubehör vor; der Validator prüft Last und Parkhaushöhe mit dieser Ladung.

Für den Konstrukteur

Beinfamilie statt Beine je Fahrzeug — das Stärkste, was die Bibliothek liefert. Gemeinsamkeitsanalyse zwischen Fahrzeugen: deckt der Ständer Halterung 2 mit Langlochverstellung sowohl Defender als auch Ranger ab und unterscheidet sich nur der Dachhalter Halterung 1 — das bedeutet um ein Vielfaches weniger Artikel, Werkzeuge bei Blechcon und Lagerbestände. Das System berechnet die Gemeinsamkeit automatisch, sobald Pässe mehrerer Fahrzeuge vorliegen: für jedes Teil, auf wie vielen Fahrzeugen es unverändert, mit Streckung oder mit anderem Halter anwendbar ist. Zielkennzahl: Anteil gemeinsamer Teile in einem neuen Produkt. Verstellbereich aus der Streuung: Scan gegen OEM-CAD zeigt, wie stark reale Fahrzeuge von der Zeichnung abweichen — Karosserietoleranzen, Anbauteile, Absacken. Langlochlänge und Höhenreserve des Halters müssen diese Streuung mit Faktor abdecken — eine Zahl, keine Intuition. Regel: Verstellweg $\geq 2 \times$ gemessene Streuung. Festigkeit und Masse zusammen: das geplante CalculiX deckt den Durchlauf von Dicke und Form des Beins unter Last ab — wo 2,5 statt 3,0 mm möglich ist, wo Rippen oder Abkantungen nötig sind, wo Bohrungen schwächen. Masse je m² Plattform als Zielkennzahl gegen Wettbewerber; Modalanalyse gegen Klappern an den Querträgern. Gesamthöhe des Profils als Wettbewerbsfeld: 55 mm bei Airholder auf dem T6 gegen 70 mm bei Front Runner — unter dem Wettbewerber bleiben ist eine Validatorregel für jedes neue Fahrzeug.

Für den Betrieb

Galvanisches Paar: Edelstahl 1.4301 auf Aluminium in nasser, salzhaltiger Umgebung — Korrosion an der Fuge binnen einer Saison. Die Gummiunterlagen gibt es; Validatorregel: zwischen ungleichen Metallen immer ein Isolator, und er muss an jedem Kontakt in der BOM stehen. Geprüft über den Verbindungsgraphen: Kontakt Stahl-Aluminium ohne Isolator ist ein Fehler. Verbrauchs- und Ersatzteile: Unterlagen, Stopfen, Schrauben verschleißen und gehen verloren. Eine BOM mit Namen und Positionsnummern aus der Anleitung ermöglicht die Bestellung eines Einzelteils nach Nummer — Serviceumsatz, den es heute nicht gibt, und Kundenbindung. Jede BOM-Position ist ein potenzieller Shopify-Artikel. Diebstahlschutz: Sicherheitsmuttern an den Dachhaltern als Zubehör — werden nachgefragt, kosten wenig, erhöhen den Bon. Wartung nach Geländefahrt: die Anleitung enthält die Drehmomentkontrolle; das System kann eine Karte der Kontrollpunkte ausgeben — welche Schrauben mit welchem Moment zu prüfen sind — aus demselben Verbindungsmodell.

Für Marke und Rückverfolgbarkeit

Seriennummer und QR-Code am Produkt: führt zur Anleitung genau dieser Konfiguration, zur Garantiregistrierung, zur Ersatzteilbestellung. Rückverfolgbarkeit nach ISO 9001 ab Werk: Seriennummer → product.json → Charge bei Blechcon → Portalantwort. Ein Chargenrückruf bei Defekt läuft über die Datenbank, nicht über das Gedächtnis. Markenelemente als Parametrik: das Logo wird bereits per Laser geschnitten — das Teil U5-Logo in der Baugruppe. Logo, Eckform und

Windabweiserzeichnung werden im selben System geführt und mit der Produktgröße neu erzeugt, ohne Wiedererkennbarkeit zu verlieren. Zerlegbarkeit und Recycling: Aluminium und Edelstahl trennen sich ohne Werkzeug — für den deutschen Markt ein auslobbarer Vorteil; das System kann den Materialanteil aus der BOM berechnen und die Trennbarkeit über den Verbindungsgraphen belegen (keine geklebten oder unlösbaren Verbindungen ungleicher Metalle).

Lösung	Worauf gestützt	Kennzahl
Beinfamilie	Pässe mehrerer Fahrzeuge, Bibliothek	Anteil gemeinsamer Teile im neuen Produkt
Verstellbereich	Scan gegen OEM, Streuung über Fahrzeuge	Verstellweg $\geq 2 \times$ Streuung
Montagezeit	Verbindungsgraph, Minuten je Schritt	Ziel 1 Stunde; Vormontage von Baugruppen
Ein Werkzeug	BOM	Befestigungsgrößen: 1, zulässig 2
Galvanische Isolation	Verbindungsgraph + Werkstoffe	Kontakte Stahl-Aluminium ohne Isolator: 0
Masse und Festigkeit	BOM + CalculiX	kg/m ² unter dem Wettbewerber; Reserve nach ISO 11154
Profilhöhe	Geometrie	unter dem Wettbewerber am selben Fahrzeug
Rückverfolgbarkeit	Seriennummer → product.json → Charge → Portal	Rückruf über die Datenbank

Allgemeines Prinzip: jede Designentscheidung, die sich in einer Zahl ausdrücken lässt, wird zur Validatorregel und zur Zielkennzahl. Dann wird das nächste Fahrzeug nicht «wie es kommt» konstruiert, sondern gegen messbare Ziele, und Abweichungen sind vor der Fertigung sichtbar.

15k. Rückkopplung aus dem Feld: Reklamationen, Kundenerfahrung, soziale Medien

Bisher kam die Rückkopplung vom Portal und aus Prüfungen. Dieser Abschnitt betrifft die Zeit nach dem Verkauf: von der Straße und den Menschen. Drei Quellen; alle sind zwei weitere Darstellungen in der Bibliothek bei Produkt und Fahrzeug: «Felddaten». Der Kreislauf ist derselbe wie beim Portal: Feld → Bibliothek → Regel → Validator → nächste Revision von product.json.

Reklamationen — wie Portalantworten, nur über den Betrieb

Jede Beschwerde ist ein strukturierter Datensatz: Fahrzeug, Konfiguration über die Seriennummer, Teil, Ausfallart (Schraube gelockert, Korrosion an der Fuge, Brummen bei Geschwindigkeit, passte nicht aufs Dach), Foto. Die Seriennummer führt zu product.json, zur Charge bei Blehcon, zur Portalantwort. Daraus: Ausfallhäufigkeit je Teil und je Fahrzeug; Grundursache entlang der Kette — Konstruktion, Charge oder Montage durch den Kunden; Regel in den Validator: drei Beschwerden über das Lockern derselben Schraube — Anzugsmoment und Schraubensicherung in die Anleitung, Anforderung ins Verbindungsmodell. Genau wie die Portablehnung zum Abnahmekriterium wurde. Das ist der CAPA-Zyklus nach ISO 9001 mit Daten statt Aktenordner; die Ausfallkosten je Art bestimmen, was zuerst repariert wird.

Kundenerfahrung — direkt aus der Anleitung

QR-Code an jedem Schritt: «Schritt unverständlich», «dauerte länger», «Teil passte nicht». Die Montagezeit, die in 15i zur Projektkennzahl wurde, wird von realen Menschen gemessen statt über den Graphen geschätzt. Der Schritt, an dem alle stolpern, wird überarbeitet: Vormontage der Baugruppe oder ein anderes Bild.

Soziale Medien — das, worüber in Beschwerden nichts steht

Was gezeigt wird	Was es bedeutet	Wohin es geht
Selbstgebaute Halter für Kanister, Lampen, Schaufeln	Nachfrage nach Zubehör, das es nicht gibt	Zubehörkatalog, Entwicklungspriorität
Wo in die Plattform gebohrt wird	die Serienschnittstelle reichte nicht; jedes fremde Loch ist eine fehlende Nutposition oder ein fehlendes Zubehör	Plattformschnittstelle, Katalog
Wie beladen wird	reale Ladungsszenarien und Massenverteilung	Ladungsszenarien in product.json (15i)
Auf welchen Fahrzeugen	Träger auf einem Fahrzeug, für das er nicht gemacht wurde — Nachfrage nach einem Programm	Warteschlange der Fahrzeugeinführungen
Was bricht und wie repariert wird	Reklamationen, die nie ankamen	Ausfallbibliothek, Validator

Sammelmechanik: Beiträge nach Hashtags und Erwähnungen; fertige KI liest und ordnet nach Labels — Zubehör, Fahrzeug, Modifikation, Ausfall, Szenario. Keine Big-Data-Analytik, sondern eine wöchentliche Zusammenfassung: «12 Beiträge mit selbstgebaute Schaufelhalter im Monat». Dazu Material fürs Marketing mit Zustimmung des Autors.

Sechstes Kriterium — das Feld

Die Abnahmekriterien bilden eine Kette: Metrologie, Geschlossenheit, Portalerkennung — Fertigung; Sitz nach Scan — Montage; Wareneingangsprüfung; und das Feld — das Produkt haben nicht Blehcon und nicht das Labor angenommen, sondern die Straße und die Menschen. Über die Seriennummern ist sichtbar, welche Revision bei wem steht; Rückruf und Verbesserung sind adressiert statt serienweit.

15I. Versionsmodell: Lehren aus Onshape

Quelle ist der Artikel «Under the Hood: How Collaboration Works» von Ilya Baran, VP Architektur bei Onshape (dem einzigen CAD mit Git-ähnlicher Verzweigung). Darin steht das Prinzip, zu dem wir selbst gekommen sind — «Wahrheit ist die Definition, Geometrie ist Artefakt» — über zehn Jahre zu Ende gedacht. Onshape als Produkt passt uns nicht (Cloud von PTC in den USA), das Datenmodell übernehmen wir.

Drei Datentypen werden strikt getrennt: der Oberflächenzustand wird nicht gespeichert; die Definition — Operationsliste, Namen, Importe — ist das Einzige in der Datenbank; die Ergebnisse der Neuberechnung — B-Rep, Dreiecke, Fehler — werden gecacht und stets aus der Definition neu gebaut. Mikroversion: die Definition ist ein ewiges unveränderliches Objekt; jede Änderung erzeugt eine neue Mikroversion mit Verweis auf den Vater und der Änderung selbst («Tiefe Extrude 1 → 4 Zoll»). Die gesamte Entwicklung wird gespeichert, jeder Zustand ist wiederherstellbar. Robuste Änderungen — der Schlüsselfund: eine Änderung adressiert die Operation über eine interne dauerhafte Kennung, nicht über die Position; solange die Ausgangsoperation existiert, ist die Änderung auf eine andere Mikroversion anwendbar. Deshalb legen sich Korrekturen mehrerer Menschen auf die letzte Version, ohne einander zu stören. Verzweigung: der Arbeitsbereich ist ein Zeiger auf die letzte Mikroversion; die Mikroversionen bilden einen Baum. Zusammenführen heißt, die Änderungen aus «woher» zu finden, die in «wohin» fehlen, und eine Mikroversion mit ihnen zu erzeugen. Bewusster Unterschied zu Git: Konflikte gibt es nicht — wird ein Maß in zwei Zweigen unterschiedlich geändert, gewinnt «woher», der Nutzer wird nie blockiert. Rückgängig ist kein Zurücksetzen, sondern das Anwenden der Umkehrung auf die letzte Version: meine Bohrung verschwindet, die fremde Verrundung bleibt. Compare vergleicht Versionen auf denselben unveränderlichen Mikroversionen.

Bei Onshape	Bei uns	Stand
Definition / Ergebnis	project.json / STEP-Artefakte	vorhanden; formalisieren: STEP ist Cache, keine Wahrheit; in Git nur für Releases
Mikroversion mit Vater und Delta	flaches Journal	zu tun: jede Operation als Objekt mit Hash des Vaters
Adressierung über dauerhafte ID	Bohrungen über Zentrum, Wand über Normale und Fläche	teilweise; als Regel für alle Operationen ausbauen
Wiedereinspielen auf neuen Vater	nicht vorhanden	das Wichtigste: das Journal wird auf eine neue Revision des Originals angewandt — Ranger-Halter v2 erhält dieselben Korrekturen
Zweige und konfliktfreies Zusammenführen	nicht vorhanden	auf Journalebene, Regel «Quelle gewinnt»
Versionsvergleich	nicht vorhanden	geometrischer Diff: Flächen hinzugefügt, entfernt, verschoben — als Hervorhebung
Tessellierung auf Anforderung, Cache	Streamlit-Cache	formalisieren: Cache nach Hash der Mikroversion
FeatureScript	Generator in Python/OCCT	Bestätigung der Abstraktion; nichts zu ändern

Wo es bei uns schwieriger ist: Onshape hat einen eigenen Kern, der dauerhafte Flächenkennungen über Neuberechnungen führt. OCCT hat sie nicht — das bekannte Topological Naming Problem, an dem FreeCAD jahrelang litt. Wir adressieren Elemente über einen geometrischen Fingerabdruck: Typ, Normale, Zentrum, Fläche. Für Blechteile funktioniert das — so werden Bohrungen und Wände bereits gefunden. Universell nicht; das wird anerkannt. Zwei weitere UX-Ideen von Onshape für unseren Viewer: synchrone Anzeige von Abwicklung und gebogenem Teil (Klick auf eine Fläche im einen Fenster hebt sie im anderen hervor) und visueller Vergleich der Teilrevisionen.

RECADA / Airholder — vollständige Projektbeschreibung v50

Deutsche Übersetzung, Teil 5 — Abschnitte 16 bis 21 und die Ergänzungen 15m bis 15u

16. Infrastruktur: Hardware-Entscheidungen

Warum Deutschland die günstigsten Server der Welt hat: Industriestromverträge + Free Cooling, DE-CIX (nahezu kostenloser Traffic → unlimited), Rechenzentren in günstiger Provinz an dicker Glasfaser, Kultur des Preiswettbewerbs (Hetzner/netcup — Familienunternehmen ohne Investoren), vertikale Integration mit Desktop-Hardware. AWS verworfen: 6-8-facher Instanzpreis, Egress ~0,09 \$/GB tötet VDI-Streams (bei Hetzner unlimited), CLOUD Act gegen Datensouveränität, Komplexität ohne Nutzen. Kompatibilität ist kostenlos (Docker zieht an einem Tag um), Abhängigkeit nicht nötig.

Rolle	Lösung	Status / Preis
Zielserver (Kern + VDI 8-10 Sitzungen)	Hetzner EX44: i5-13500 (schneller Einzelkern — Schlüssel zur Reaktionsfreudigkeit), 64 GB, 2x512 NVMe RAID 1, unlimited, FSN1	59 € netto, kein Setup, stundenweise; Limited-Wellen — Verfügbarkeitsabo, 15-Sekunden-Regel bei der Bestellung
Paralleler Kanal	Server Auction (Robot → Server Market): Suche «i5-13», «i5-12», «i7-12»; NVMe, keine HDD; nichts älter als 12. Generation	Kriterium: ≤ 75-80 €
Plan B	OVH RISE-S: Ryzen 9700X (bester Einzelkern), 64 GB, Frankfurt/Straßburg, monatlich kündbar	78 € + 78 € Setup
Verworfen	AX41 (Zen 2 von 2019 zum selben Preis), AX42 (118 € für einen 35-W-Chip), EX63 (177 € — Überzahlung für Kerne), i9-13900 aus der Auktion (165 € — zu früh), Contabo (CPU steal)	—
Skalierung (Etappe 5, 10+ Sitzungen)	OVH RISE-L (9950X, 128 GB, 179 €) — verdrängte AX102 (308 €); i9 aus der Auktion	nach tatsächlichem Wachstum
Bereits vorhanden	ThinkPad i7-8850H / 32 GB / Quadro P1000 — Arbeitsplatz und lokaler Start der Etappe 1 (Git, build123d, FreeCAD); Strato VPS 2/2/90 — Gitea + WireGuard und/oder zweite Backup-Kopie; Hetzner CPX — CI	0 € neu
Zweiter Anbieter	netcup: RS G12 (dedizierte EPYC-Kerne, ECC, günstig) — Backup jetzt, künftig möglicher separater Kern (Gitea + CI); EPYC-Einzelkern ist für VDI schwach	~5 €/Monat

Drei Hetzner-Zugänge: accounts (Ausweis) → Cloud Console (Cloud-CPX) und Robot (dedizierte Server: Bestellung, Rescue, Server Market). EX44 nach Lieferung: rescue → installimage → Ubuntu 24.04 LTS + RAID 1 → SSH-Schlüssel, ufw (nach außen nur SSH + WireGuard), unattended-upgrades, Backup in die Storage Box (~4 €) + zweite Kopie bei netcup. Domain: Subdomains git./work./cdn. der bestehenden Domain (oder eine eigene technische ~15 €/Jahr) — Let's Encrypt, Stabilität bei Umzügen, Trennung der Bereiche; Arbeits-Subdomains können ausschließlich innerhalb von WireGuard auflösen. Autorisierung — drei überschneidungsfreie Kreise: Arbeit (lokale Konten Gitea + Kasm + TOTP-2FA + WireGuard-Perimeter; SSO/Keycloak ab 20 Personen), Kunde (Shopify Customer Accounts / Google — der Konfigurator erbt das), Community (OAuth GitHub/Google in der öffentlichen Gitea). Budget des Gesamtsystems bei 100 Registrierten (Spitze 20-25 Sitzungen): ~150-200 €/Monat.

Betriebskosten: ausschließlich Open Source

Kategorie	Werkzeuge (alle kostenlos, ohne Abo)
CAD und Generierung	FreeCAD + SheetMetal + Add-ons (GPL/LGPL), build123d/CadQuery (Apache)
Versionen und Server	Git, Gitea, Git LFS, zippey; Ubuntu, Docker (self-hosted), WireGuard, ufw; Let's Encrypt; Cloudflare free (CDN)
Berechnungen	CalculiX + Gmsh (Statik, Modal), OpenFOAM (CFD einmalig)
Rendering und Artefakte	Blender headless, SheetNest (Zuschnitt), gltfpack/Draco (GLB)

Kategorie	Werkzeuge (alle kostenlos, ohne Abo)
VDI	Kasm Community Edition (bis 5 gleichzeitige Sitzungen); ohne Limit — Apache Guacamole / webtop-Images von linuxserver.io
Visualisierung	Mermaid/Graphviz (Abhängigkeitsgraph), React Flow/Rete.js (Composer, Etappe 5), Sverchok (Node-Modellierung)
Konfigurator	Unity Personal — kostenlos bis 200 k\$ Jahresumsatz mit dem Produkt; das einzige Werkzeug mit Schwelle; kostenlose Alternative bei Bedarf — three.js/Babylon.js

Bezahlte Posten des Projekts sind nur drei, alle bewusst angenommen: Server (~70–200 €/Monat je nach Wachstum), Claude-/API-Tokens (~20–50 €/Monat zu Beginn), physische Welt (FDM-Drucker einmalig 300–500 €; Prüfungen nach ISO 11154 — einmalig im vierstelligen Bereich). Projektregel: ein Werkzeug kommt nur in den Stack, wenn es Open Source oder für unsere Größenordnung ohne Abo kostenlos ist; bezahlt sein dürfen nur Rechenressourcen und physische Dinge; ein Kandidat für ein kostenpflichtiges Werkzeug erfordert eine ausdrückliche Entscheidung des Inhabers mit einer kostenlosen Alternative in der Tasche.

Claude als Kopilot — vier Schichten: (1) Claude Code auf dem Server — Admin-Dialog statt manuellem Linux; (2) MCP-FreeCAD headless — Motor der Prompt-Schicht; (3) in der CI nicht interaktiv (claude -p): Validator rot → Fix als Commit in einen Zweig → Review durch den Menschen; Übersetzung abgegebener STEP in Generator-Entwürfe; (4) im VDI-Image des Auftragnehmers — separater API-Schlüssel mit Limit und RECADA-Systemprompt (der Kern bleibt unsichtbar). Regel: Claude geht durch dieselben Türen wie Menschen (Zweige, PR, Validator), Commit-Autor «LLM via MCP» mit dem Prompt in der Nachricht.

17. Abhängigkeitsgraph der Konstruktion und Automatisierung

Der Abhängigkeitsgraph ist die visuelle Karte der Konstruktion: Knoten sind Wahrheitsdateien (Pässe, Skelette, Generatoren, Teile, Baugruppen, Konfigurationen), Pfeile bedeuten «liest aus».

Abhängigkeiten sind im System explizit deklariert, deshalb baut die CI die Karte automatisch per Skript (Mermaid/Graphviz, kostenlos) — nichts muss von Hand gezeichnet werden, das Schema ist immer aktuell. Hauptmodus ist die Änderungsanalyse: vor dem Merge beantwortet die CI «was wird neu gebaut, wenn dieser PR angenommen wird» — Liste und Hervorhebung im Schema (rot = betroffen, released-Teile gesondert markiert). Die Kaskade «was zieht was nach sich» wird sichtbar, BEVOR der Knopf gedrückt wird; zugleich ein Onboarding-Werkzeug (ein Neuer versteht das System in einer Stunde anhand der Karte) und ein Diagnosewerkzeug (wo in der Kette ein roter Validator steht). Umsetzung in Etappe 2 neben dem Validator.

Automatisierung von Geschäftsereignissen — mit einfachen Skripten und Webhooks in der CI, ohne eigenen Orchestrator: Lager-Webhook von Shopify (Bestand A = 0 → Run-out-Politik → B aktivieren → Benachrichtigung), Onboarding (Test bestanden → Ausführendenkarte → E-Mail), Monitoring (Release-Tag → Zusammenfassung in den Kanal; CI einen Tag rot → Alarm), Marketing (Release → Rendering → Entwurf der Produktseite per LLM zur Freigabe), Vertrieb als Quelle von Lastenheften (Formular «keine Befestigung?» → Issue + Nachfragezähler → 3 Anfragen = Priorität in der Entwicklung). Die Grenze bleibt unverrückbar: die Automatisierung fasst keine Geometrie an und schreibt nicht in main — alles, was die Wahrheit ändert, geht über PR und Validator. Ein Skript entsteht für ein Ereignis, das dreimal von Hand vorkam. Node-Composer über der Bausteinbibliothek (React Flow/Rete.js — offen) — Etappe 5: Oberfläche für Nicht-Programmierer (Händler-CPQ, Kunde mit Vorlage); ein eigenes Grasshopper wird nicht gebaut (es gibt Sverchok und FreeCAD-Nodes, kostenlos).

18. Finanzmodell — Grundgerüst

Lieferanten und Daten für eine ehrliche Ökonomie: ein strukturiertes Archiv der Blechcon-Rechnungen und des GLS-Tarifs ab der ersten Bestellung + Pre-flight-Nesting (den fairen Schnittpreis vor der Anfrage kennen) + regelmäßiges manuelles Benchmarking von ein bis zwei europäischen Alternativen + Qualifizierung eines zweiten Lieferanten als Monopolversicherung. Eine vollständige API-Auktion mehrerer Dienste wurde verworfen: die Validatorregeln sind die Regeln eines konkreten Herstellers, mehrere Lieferanten bedeuten vielfache Pflege der DFM-Regeln.

CM1 (Marge des Satzes) = Preis – Metall – Blechcon – Profile – Befestigung/Beschichtung – Verpackung/Versand – Kanalprovision. CM2 (mit Design) = CM1 × erwartete Verkäufe des Modells – (Turniere + Stunden des Inhabers × Satz + Prototypen + Pass). Operativ = Σ CM2 – Fixkosten (Infrastruktur, Zeit, Marketing, Lager).

Branchenorientierung (4x4-Zubehör): Rohmarge 45–60 % im Einzelhandel, 30–40 % über Händler; unter ~35–40 % vom Einzelhandelspreis vor der Skalierung reparieren. Kernfrage des Systems: nach wie vielen verkauften Sätzen amortisiert sich die Einführung eines neuen Fahrzeugs. Dafür braucht es acht Zahlen: Endkundenpreis des Satzes und Kanal; Blechcon-Rechnung für einen Beinsatz (oder Masse); €/Meter Profil und Meterzahl; Befestigung/Dichtungen/Beschichtung; Verpackung + Versand; aktuelle und geplante Stückzahlen; interner Stundensatz des Inhabers; heutige Kosten der Fahrzeugeinführung nach alter Methode. Kennzahl Nr. 2 enthält den Metalltransport — die Unit Economics bleibt ehrlich.

19. Managementebene: Rollen im Detail

Rollen sind keine Positionen, sondern Verantwortungsbereiche mit klaren Entscheidungsrechten; in einem kleinen Team vereint eine Person mehrere Rollen, aber die Entscheidungsgrenzen bleiben schriftlich. Prinzip: jede Entscheidung hat einen Eigentümer; «gemeinsam entschieden» heißt, niemand hat entschieden.

Rolle	Wofür verantwortlich	Entscheidungsrechte	Zugänge
Owner (Inhaber)	Strategie und Geld; Prioritäten der Pipeline; Abnahmekriterien; Vertrauen in Menschen; Beziehungen zu Blechcon und Schlüsselpartnern; Entwicklung des Systems (Etappentore)	Allein: Release-Knopf (Tag), Budgets, Beförderungen auf der Leiter, Änderung der Plattformschnittstelle, Änderung released-Teile, Zulassung von Werkzeugen in den Stack (FOSS-Regel)	Vollzugriff: Git-Admin, Server-root, Robot/Cloud Hetzner, Shopify, Bank
Lead Reviewer	Erstabnahme aller PR (Geometrie, Passung, Montierbarkeit, Ökonomie nach BOM, Ästhetik); Führung der Ausführenden; Formulierung der Lastenhefte mit dem Owner; Mentoring; Vorschlag von Regeln in rules/ aus wiederkehrenden Fehlern	Merge in main für Teile (außer Schnittstellen- und released-Änderungen — Eskalation an den Owner); Ablehnung von Arbeiten; Aufgabenverteilung; Eskalation von Beförderungen	Git: Merge-Rechte auf products/accessories; Lesen von generators und rules; Gitea-Kanban; kein Server-root
Senior Engineer	Komplexe Baugruppen und neue Teiletypen (Topologie); Übersetzung «Sandbox → Code»; Ausbau der Bausteinbibliothek; FEM-Kalibrierung an Referenzberichten; komplexe Pässe (Windschildflächen)	Commits in generators/ über PR; Anlegen von Zweigen convert/*; Vorschlag neuer Bausteine; technische Entscheidungen innerhalb des Lastenhefts ohne Abstimmung	Git-Zugang (lokale Arbeit + push), ausgewählte Repositories; VDI nach Wunsch; keine Merge-Rechte in main
Engineer (geprüft)	Direkte Aufgaben nach fertigen Vorlagen und Lastenheften: Adapter, Beinvarianten, Zubehörpässe nach Checkliste; gedruckte Anproben eigener Teile	Arbeit im eigenen Zweig; Abgabe über PR; Rückfragen zum Lastenheft erwünscht (werden als Wachstumssignal protokolliert)	VDI-Sitzung mit erweitertem Repository-Satz; oder lokal + Git nach Entscheidung des Owners

Die erste delegierbare Rolle ist der Lead Reviewer — sie löst den Bus-Faktor: mindestens einmal im Monat führt er die Abnahme vollständig ohne den Owner durch. Die Auswahl erfolgt nach Denkqualität in den Kommentaren, nicht nach Zeichengeschwindigkeit (Peter-Prinzip). Das Ziel des Owners ist die Verdichtung auf die Steuerung der Pipeline; alles, was ihn hineinzieht (zeichnen, CI reparieren), ist ein Signal für eine unbesetzte Rolle.

20. Risiken und Engpässe

Nach der Praxis ergänzte Risiken (September 2026): das Werkzeug ist eine Datei ohne automatische Tests — die Bibliothek als Regression ist das erste Heilmittel. Abhängigkeit von der OCCT-Version: 7.8 in FreeCAD und 7.9 im Kern verhalten sich unterschiedlich — das kostete einen halben Tag; die Kernversion fixieren und nicht vermischen. Das dritte Kriterium wird von Hand über ein fremdes Portal geprüft, das sich ohne Vorwarnung ändern kann — die Portalantworten als Daten führen, damit Änderungen sichtbar werden. Dazu die bekannten: Verstoß gegen die Regel «bearbeitet wird nur die Wahrheit» (manuelles STEP in die Fertigung) als Haupttodesursache des Systems; Bus-Faktor auf dem Owner, bis der Lead Reviewer die Abnahme allein führt; Abhängigkeit von einem Lieferanten; fehlende physische Prüfungen nach ISO 11154 als persönliches Haftungsrisiko und B2B-Sperre.

21. Roadmap

Etappe	Inhalt
0. Fundament	Server aufsetzen (OS, Schlüssel, ufw, WireGuard, restic in die Storage Box), Gitea, Repository-Struktur, CLAUDE.md als Einstiegspunkt
1. Rückgrat	Kern und Oberfläche trennen, Aufgabenwarteschlange, dauerhafte Ergebnisspeicherung; Profilkatalog; erste Generatoren; params.json und Pässe
2. Validator und Graph	dfm_blechcon.json aus der Praxis; features.py (Freischnitte); Abnahmelaufroutine aus spec.yaml; Abhängigkeitsgraph; Bibliotheksscanner; Regression in der CI
3. Arbeitsartefakte	K-Faktor-Kalibrierung über das Portal; Freischnittkatalog → Erzeugung der Biegezone nach Regel; DXF, Zeichnungen aus HLR, BOM, Fertigungspaket; Montageanleitung aus Daten

Etappe	Inhalt
4. Menschen	VDI/Guacamole, Rahmenverträge und Rechte, Wettbewerbe und Karriereleiter, erster Lead Reviewer, hybrider FreeCAD-Workbench
5. Skalierung	Fahrzeugpässe im Produkteditor, Konfigurator, Zubehörkatalog, Node-Composer, Monetarisierungsmodelle

Drei Kontrollpunkte der Sinnhaftigkeit: die Bibliothek mit Regression läuft; der erste Ranger-Haltersatz aus dem Werkzeug besteht das Portal; nach sechs Monaten wurde ein Fahrzeug schneller eingeführt als der Defender. Alles Weitere — Anleitungen, Scans, Streaming, Monetarisierung — erst danach.

15m. Werkstoffe und Technologien

Jede Teilegruppe hat ihren Werkstoff, ihre Gefahr und ihre Technologie. Der Werkstoff ist ein Teileattribut in der BOM mit Dichte, Korrosionsklasse und UV-Klasse; daraus berechnet das System die Masse, prüft galvanische Paare über den Verbindungsgraphen, fängt ein Außenteil ohne UV-Klasse und führt den Werkstoffwechsel als Produktparameter ohne Geometrieneuaufbau.

Teil	Werkstoff	Gefahr	Technologie
Plattformprofile (S13A, B13A)	Aluminium 6060/6063 T6; 6082 T6 bei Festigkeitsbedarf (+30 %, schwerer zu pressen)	Galvanik an der Fuge zum Edelstahl; die Festigkeit bestimmt der Querschnitt, nicht die Legierung	Strangpressen; Eloxal oder Pulver
Beine, Halter, Windabweiser	1.4301 t = 2-3 mm; für Salz und Küste 1.4404 / 1.4571 (+30-40 %, biegt und schneidet sich gleich)	Lochfraß durch Chloride bei 1.4301 nach 2-3 Saisons	Laser + Biegen bei Blechcon; Pulver
Unterlagen	EPDM	NBR reißt durch Ozon und UV in ein bis zwei Jahren	Stanzen
Ecken, Stopfen, Deckel	ASA oder PA6 GF mit UV-Stabilisator; Schwarz hält UV am besten (Ruß als Stabilisator)	ABS und PLA nicht für außen	Spritzguss; ASA/PA12-Druck für Kleinserien
Befestigung	DIN, Klasse A2; A4 für die Küstenversion	Galvanik ohne isolierende Scheiben	Katalog
Prototypen	PA12 (SLS/MJF); Einzelbiegung bei Blechcon	PLA hält nichts	3D-Druck

Galvanisches Paar — der Hauptfeind. Edelstahl auf Aluminium in nasser Umgebung: korrodiert das Aluminium. Drei Schutzmaßnahmen: Isolator zwischen den Metallen (Gummiunterlagen — Validatorregel aus 15i: Kontakte ohne Isolator gleich null); Schrauben A2/A4 mit isolierenden Scheiben; Eloxal oder Pulver auf dem Aluminium als Barriere. Dazu Drainage: im Stoß darf kein Wasser stehen — wird über die Knotengeometrie geprüft. Ultraviolett betrifft Polymere und Beschichtungen, nicht Metalle: Polyesterpulver hält UV, Epoxid kreidet in einer Saison — für außen nur Polyester oder Hybrid mit Polyester außen. Eloxal ist UV-fest, aber kratzempfindlich und farblich begrenzt. Gummi EPDM. Kunststoffe: ASA statt ABS, PA6 GF mit UV-Stabilisator, PA12 im Druck.

Produktfolgen: die «Küstenversion» ersetzt 1.4301 durch 1.4404 und A2 durch A4 ohne Geometrieänderung — ein Produktparameter in product.json, eine andere BOM-Zeile, ein anderer Preis. Es wird nicht geschweißt — und das ist richtig: Zerlegbarkeit, keine Gefügeveränderung im Edelstahl, einfache Logistik im Karton. Wettbewerber verwenden dieselben Werkstoffe: das Wettbewerbsfeld ist nicht der Werkstoff, sondern Masse je m² und Beständigkeit der Beschichtung. Werkstoffkarten in der Bibliothek: Dichte, Streckgrenze, Modul, Korrosionsklasse, UV-Klasse, Preis je kg — dieselben Zahlen gehen in BOM, Validator und CalculiX.

Zeiterfassung über das Journal

Jede Operation in project.json erhält Zeit und Autor. Ohne Zusatzeingaben berechnet sich: Durchlaufzeit vom Laden bis «tauglich»; reine Arbeitszeit über die Intervalle zwischen Operationen mit Leerlaufschwelle; Anzahl der Iterationen über das Portal; Zeit nach Typen — Reverse, Korrekturen, Warten auf den Lieferanten. Git liefert dasselbe für die Generatoren. Mit wachsender Bibliothek folgt die Schätzung neuer Arbeit über ähnliche Teile: «ein räumlicher Halter mit sieben Biegungen — drei Stunden, ein L-förmiger mit zwei — vierzig Minuten». Nötig für den Einführungsplan eines neuen Fahrzeugs in Stunden, den Preis von Auftragsarbeiten, Abschnitt 8 (Aufstieg nach Logs) und den Kontrollpunkt «Ranger schneller als Defender». Einschränkungen: die Denkzeit in FreeCAD sieht das Journal nicht — Handmarkierung oder Commits; und die Daten dienen der Planung und Schätzung, sind für den Ingenieur selbst einsehbar und sind keine Stoppuhr zum Vergleich von Menschen.

15n. Externe Ressourcen: was wir nehmen und was nicht

Übersicht vom 6. September 2026. Geprüft: Stand des OCCT-Kerns, offene Abwicklungsbibliotheken auf GitHub, eigene Kerne Chinas und Russlands. Ergebnis: zu nehmen ist eines — und es deckt sich bis ins Detail mit unserem Ansatz.

Aktuelle Kernversion ist 8.0.1 vom 30. Juli 2026; davor der Patch 8.0.0p1 im Juni (Neubau der Gordon-Flächen: die experimentelle Implementierung wurde durch eine neue mit ausdrücklichem Status und Näherungsmodus als Rückfall ersetzt). Die Acht lebt drei Monate und wurde zweimal stabilisiert — der Major-Release ist nicht mehr riskant. Über das in v31 Festgehaltene hinaus: Umstellung auf C++17, Standardausnahmen, BRepGraph als Ersatz für die Baumtraversierung über TopExp_Explorer mit Indexkarten — genau die langsamste Stelle unserer Vermessung. Für uns ändert sich heute nichts: unser Kern kommt über cadquery-ocp, gebaut für 7.9.3; solange OCP nicht für die Acht neu gebaut ist, gibt es kein Ziel. Alle paar Monate prüfen. Reihenfolge unverändert: zuerst Bibliothek und Regression, dann der Wechsel — die API ist gebrochen, das Verhalten der Booleschen Operationen kann sich geändert haben. FreeCAD wie bisher: stabiler Zweig 7.8, dev 7.9.

YouCanNotUnfold — das Einzige, was es zu nehmen lohnt

Autor Alexander Neufeld, Lizenz LGPL-2.1, in die Haupt-Sheet-Metal-Workbench von FreeCAD integriert. Abwicklung von Blech, gebaut auf networkx — wie unsere Vermessung; der Autor weist darauf hin, dass dies gegenüber der Standardabwicklung spürbar schneller ist und Formen abwickelt, die jene nicht bewältigt.

Was dort ist	Wofür es uns nützt
unfold.py auf networkx	ihren Graphen mit unserem FTG vergleichen; möglicherweise ist ein Teil dessen gelöst, wo wir bei den Freischnitten steckenblieben
test_cases + test_unfold.py	ein fertiger Satz Testteile mit Tests — genau die Regression, die wir bauen wollen
svg_writer.py	Ausgabe der Abwicklung als SVG; Ergänzung zu unserem DXF
LGPL-2.1	mit der Nutzung in unserem Projekt vereinbar

Chinesische und russische Kerne — geschlossen und schwächer: Overdrive (ZWSOFT, 2010 von der amerikanischen VX gekauft), DGM (Huayun / CrownCAD), CRUX IV (Shandong Huatian), AMCAX (Universität für Wissenschaft und Technik Chinas), CASGEO (Akademie der Wissenschaften); dazu das russische C3D (ASCON). Alle sind geschlossen und kommerziell. Chinesische Übersichten räumen selbst ein, dass Overdrive gegenüber CGM, ACIS und Parasolid bei Synchronstechnologie, Freiformflächen und Verrundungen zurückliegt — selbst bei Verfügbarkeit wäre das kein Fortschritt gegenüber OCCT. C3D ist stark, mit Blechmodul, aber kommerziell und für ein deutsches Unternehmen in der aktuellen Lage nicht praktikabel. Unerwartete Bestätigung der Wahl: in der akademischen Arbeit P3D-Bench (Juni 2026) zählt CadQuery auf OCCT zu den Referenzwerkzeugen für die Generierung parametrischer CAD-Modelle — die Forschung nutzt genau unseren Stack.

15o. Geschwindigkeit: was die Einführung zusätzlich beschleunigt

Ausgewählt mit demselben Filter, der die «fundamentalphysikalischen» und «Gamedev»-Ideen aussortiert hat: nur was die Produkteinführung beschleunigt, nicht was das System verziert. Neun Ideen; die drei mit dem größten Hebel sind der Sprint als Protokoll, Prüfungen nach Familie und der Preis im Kreislauf.

#	Idee	Warum schneller	Priorität
1	Sprint für ein neues Fahrzeug: Pass aus Aufbaurichtlinien und OEM → Beinauswahl aus der Familie über die Bibliothek → product.json v0 → Prüfungen → Paket → Portal → Preis. Kennzahl: Stunden von den Daten bis zum Blechcon-Angebot	die Fahrzeugeinführung wird vom Projekt zur Prozedur; Ranger ist der erste Durchlauf	hoch
2	Prüfungen nach Familie statt je Fahrzeug: CalculiX ist am ISO-11154-Prüfstand kalibriert; ein neuer Halter innerhalb des geprüften Bereichs wird rechnerisch mit Verweis auf den geprüften Verwandten begründet	eine Prüfung je Beinfamilie statt je Fahrzeug — die größte Ersparnis der Kette	hoch
3	Preis im Kreislauf: Kosten bei Blechcon als Funktion von Schnittlänge, Biegungen, Werkstoff, Losgröße; kalibriert an realen Angeboten	jede Konstruktionsänderung zeigt sofort Euro	hoch
4	Begrenzter Durchlauf: nachts Dicken, Flanschbreiten, Bohrbilder durch CalculiX und Validator — morgens die Pareto-Front Masse / Steifigkeit / Preis	der Mensch wählt aus geprüften Varianten	mittel
5	Zuschnitt als Hinweis: Verhältnis der Zuschnittfläche zur konvexen Hülle	«Flansch drehen — 8 % weniger Material», eine Validatorzeile	mittel
6	Auftragsadapter nach Kundenupload: STEP oder Scan des Zubehörs → nächste Schnittstelle → Adapter durch Strecken eines bewährten Teils → Preis und Termin	individuelle Lösungen zum Serienpreis, neuer B2B-Kanal	mittel
7	Produktseite als Artefakt von product.json: Maße, Masse, Höhe, Zubehör, Anleitung → Shopify-Seite	weicht nie vom Produkt ab	mittel
8	Upgrade-Satz als Differenz der Revisionen: Diff zweier product.json → BOM «was nachzukaufen ist»	Verkauf an Bestandskunden ohne Neukonstruktion	niedrig
9	Zweites Lieferantenprofil: zwei DFM-Regelsätze im Validator	das Teil hängt nicht von einer Werkstatt ab	niedrig

15p. Remote-Desktops: Wege, Auswahl, Geografie

Weg	Was es ist	Plus	Minus
xrdp	Linux-Desktop über RDP, Client in Windows enthalten	sparsamste Variante, latenztolerant, keine Einrichtung beim Client	3D kommt als Bild, Drehen zäh
Guacamole (Apache)	Gateway RDP/VNC/SSH über den Browser	offen, kostenlos ohne Limits, Sitzungsaufzeichnung, 2FA, Sperre von Zwischenablage und Dateien	startet selbst nichts — ein Desktop wird benötigt
Kasm Workspaces	Desktop im Einweg-Container je Nutzer, im Browser	vollständige Isolation, Wasserzeichen, Sperren eingebaut	schwer; kostenlos bis 5, danach Lizenz
Selkies	WebRTC-Desktop im Browser, offen	Videostream, flüssiges 3D	Einrichtung, Kodierung belastet die CPU
Xpra	eine Anwendung als eigenes Fenster beim Client	sparsamer als ein voller Desktop	—
NoMachine	NX-Protokoll	sehr effizient	geschlossen, Lizenz für Unternehmen
Sunshine + Moonlight	Spiele-Streaming	minimale Latenz	für GPU gedacht; ohne sie CPU-lastig
Viewport-Streaming OCCT	nur das 3D-Fenster (15c)	am leichtesten und sichersten	nur Ansicht und Auswahl

Guacamole gegen Kasm: Guacamole ist die Tür zum Desktop, Kasm der Desktop im Container. Für ein vertrauenswürdiges Team bis fünf Personen — Guacamole mit xrdp: einfacher, leichter, kostenlos. Für nicht vertrauenswürdige Auftragnehmer — Kasm: Isolation je Container. Empfohlenes Schema: WireGuard (persönlicher Schlüssel) → Guacamole (2FA, Aufzeichnung, Zwischenablage aus) → xrdp-Desktop mit FreeCAD, Werkzeug und Blender. Alles offen, alles auf dem EX44, Kosten null. Für interaktives Blender mit schweren Szenen ohne GPU wird es eng — Renderings nachts im Stapel.

Geografie und Latenz: die Physik bestimmt, nicht die Software — Osteuropa und Türkei 30–50 ms, Nordafrika und Naher Osten 60–90, Indien 130–180, Südostasien 180–220, Lateinamerika 200–250. Bis 80 ms wie lokale Arbeit; 100–150 erträglich; über 150 verzögert jeder Klick, 3D wird zur Qual. Software trägt Millisekunden bei, das Protokoll spürbar: RDP und Xpra übertragen Oberflächenbefehle und tolerieren Latenz, Kasm und Selkies streamen Video und werden auf entfernten Kanälen unscharf. Für ferne Länder Guacamole mit RDP. Wichtiger als beides ist der Kanal des Auftragnehmers: kabelgebunden ab 10 Mbit mit gleichmäßiger Latenz, vor der Einstellung testen. Ein zweiter Server näher am Auftragnehmer hilft nicht: die Daten bleiben in Deutschland, sie zu verlagern wäre genau das ausgeschlossene Leck.

15q. Rekrutierung: drei Profile und Tests durch das System selbst

Profil	Was er tut	Niveau	Wo und wie
Systembediener	treibt Teile durch Reverse und Portal, führt Bibliothek und Pässe, überträgt Aufbaurichtlinien ins JSON, packt Pakete für Blechcon, korrigiert Anleitungen	Junior oder Mittelstufe Blech, mit Biegepraxis von Hand	fest, Geografie bis 80 ms; günstig — die Regeln stecken im Validator, nicht im Kopf
Zubehörkonstrukteur	schafft, was es nicht gibt: die Schnittstelle zwischen fremdem Objekt (Box, Zelt) und Plattform; Konzeptvarianten mit Kompromissen; Lasten, Vibration, Wind, Zugang mit den Händen	Mittelstufe oder Senior mit in Serie gegangenen Produkten (Aftermarket, Outdoor, Befestigungssysteme); Blech und Kunststoff	fest bei Zubehörfluss oder projektweise; Polen, Tschechien, Türkei, Serbien — Aftermarket-Büros
Entwickler Python/OCCT	erweitert das Werkzeug: Regeln, Freischnittkatalog, Streaming, Integrationen	Mittelstufe; kann aus dem Bediener wachsen	einer, nicht sofort

Was zu bieten ist: ein Arbeitsplatz ohne Kosten auf ihrer Seite — Desktop über Guacamole, keine Lizenzen und keine Hardware außer Laptop und Kabelinternet (für ein Land, in dem SolidWorks ein Jahresgehalt kostet, ein Argument). Ein verständliches System mit Regeln: produktiv ab der ersten Woche. Aufstieg nach Journalzahlen, nicht nach Sympathie. Moderner Stack: OCCT, Python, Git, Parametrik. Stabile Bezahlung in Euro, langfristig, mit Eigenverantwortung für den eigenen Bereich.

Test	Was er prüft
Reverse mit Falle	STEP → Reverse → Urteile erklären; ein Teil wurde real vom Portal abgelehnt: findet er die Ursache und schlägt eine Korrektur vor, statt Knöpfe zu drücken
Blechkompetenz	Zeichnung mit absichtlichen Fehlern: Bohrung an der Biegung, fehlender Freischnitt, Edelstahl auf Aluminium ohne Isolator — wie viele findet er in 20 Minuten
Lesen der Aufbaurichtlinien	Abschnitt → Zonen und Lasten in den Pass-JSON mit Seitenangabe: Sorgfalt, Sprachen, Umgang mit der Primärquelle
Ehrlichkeit	eine per Bedingung unlösbare Aufgabe: sagt er «geht nicht, weil ...» oder zeichnet er etwas, um abzugeben
Git-Disziplin	Parameter ändern, mit sinnvoller Nachricht committen, den Diff erklären
Kanal und Zeitzone	Latenzmessung zum Server vor dem Gespräch; über 100 ms für eine 3D-Rolle nein
Probemonat	an realen Teilen; Bewertung nach denselben Journalzahlen, die später den Aufstieg bestimmen
Zubehörkonstrukteur	bezahlte Aufgabe über ein bis zwei Wochen mit Ihrer Box: zwei bis drei Varianten mit Kompromissen (Masse, Preis, Montagezeit); die gewählte besteht den Validator; Preis gerechnet; Anleitungsschritt geschrieben; Bibliotheksteile verwendet
Entwickler	eine Aufgabe: eine Validatorregel mit Test hinzufügen (z. B. Randabstand einer Bohrung) als Pull Request

15r. Rechte an den Ergebnissen und Geschäftsgeheimnis

Vorbemerkung: keine Rechtsberatung; die Vertragsmuster kommen vom Anwalt für deutsches und grenzüberschreitendes Recht. Das Gerüst sieht so aus. Die Hälfte dieses Systems ist bereits gebaut — wir nannten sie Datenschutz.

Juristische Schicht — einmalig, beim Anschluss: In Deutschland ist das Urheberrecht unübertragbar, übertragen werden ausschließliche Nutzungsrechte; Erfindungen folgen einem eigenen Regime. Für einen externen Auftragnehmer: ein Rahmenvertrag, einmal unterzeichnet — alle im System entstandenen Ergebnisse mit ausschließlichen Rechten bei RECADA; Erfindungen werden übertragen und gemeldet; Vertraulichkeit; deutsches Recht, Gerichtsstand Augsburg. Unterschrift qualifiziert elektronisch nach eIDAS (Skribble oder ähnlich) — grenzüberschreitend wirksam. Ein Klick beim Login als einziger Vertrag ist eine schwache Position; ein Klick als Bestätigung des bereits Unterzeichneten eine starke.

Technische Schicht — Bestätigung bei jedem Login: Guacamole zeigt die Bedingungen vor der Sitzung («die Arbeit im System fällt unter den Rahmenvertrag vom ...») und schreibt ins Log: wer, wann, mit welchem Schlüssel zugestimmt hat. Beweis, dass die Person die Bedingungen im Moment der Entstehung kannte.

Herkunft des Ergebnisses — stärker als der Vertrag: das Projektjournal schreibt Autor und Zeit jeder Operation; Git den Autor jedes Commits; die Sitzungsaufzeichnung von Guacamole, was auf dem Bildschirm geschah. Drei unabhängige Logs: in einem Jahr muss nicht bewiesen werden, wer den Halter konstruiert hat. Die Arbeit hat den Server physisch nie verlassen — also ist sie im System entstanden; der Streit «zu Hause gemacht, später gebracht» ist unmöglich.

Geschäftsgeheimnis — unerwartete Folge des Datenschutzes: das GeschGehG schützt ein Geheimnis nur bei «angemessenen Geheimhaltungsmaßnahmen». Pixel nach außen, persönliche Schlüssel, Downloadverbot, Sitzungsaufzeichnung, Vertrag beim Login — genau diese Maßnahmen. Der für anderes gebaute Datenschutz verschafft der Bibliothek, den Pässen und den Regeln zugleich den Rechtsschutz als Geschäftsgeheimnis. Ohne diese Maßnahmen kann ein Gericht feststellen, dass es kein Geheimnis gab.

Zur «Idee»: eine Idee als solche ist nirgends geschützt. Geschützt ist ihre Verkörperung: Zeichnung und Code durch das Urheberrecht, die Form durch das Geschmacksmuster, der Mechanismus durch das Patent, alles Übrige als Geschäftsgeheimnis bei eben jenen Maßnahmen. «Eine Idee unterschreiben» funktioniert nicht; «alles, was im System entsteht, gehört uns» funktioniert. Reihenfolge: Rahmenvertrag mit elektronischer Signatur → Bestätigung bei jedem Login mit Log → Arbeit nur im System → Ergebnisse nur im Git des Unternehmens → Bezahlung an die Abnahme gekoppelt. Für einen Auftragnehmer in der Türkei oder Polen ergänzt der Anwalt ein bis zwei Seiten zum lokalen Recht. Zugang des Zubehörkonstruktors: ein Desktop nur mit den Daten seines Zubehörs und der Schnittstelle, nicht die ganze Bibliothek; hier ist Kasm passender als Guacamole.

15s. Monetarisierung: drei Modelle

Gemeinsames Prinzip: monetarisiert werden weder Code noch Daten als Dateien, sondern der Zugang zu einem System, aus dem nichts herausgetragen wird. Der externe Konstrukteur arbeitet innen — Pixel nach außen, Rechte per Vertrag beim Login, Herkunft in drei Logs. Die Fahrzeugdaten — OEM-CAD, Richtlinien, Scans — bleiben auf dem Server; sie werden nicht verkauft, sie werden zur Nutzung gegeben.

Modell A. Zubehör-Ökosystem — wie bei Front Runner, aber mit offenem Eingang

Front Runner verkauft Hunderte Zubehöerteile für die eigene Plattform — Zelt, Tische, Kanister, Halter — und das ist der größere Teil des Umsatzes je Kunde nach dem Trägerkauf. Dort entwickelt ein eigenes Team. Bei uns ist der Eingang offen: externe Konstrukteure entwerfen im System für unsere Schnittstelle (T-Nut, Airline-Schiene, Halter — als Daten beschrieben), der Validator prüft Kompatibilität, Preis und Fertigbarkeit, wir fertigen über Blechcon und verkaufen über unseren Kanal, der Konstrukteur erhält einen Anteil an jedem verkauften Stück. Was über das Vorhandene hinaus nötig ist: die Schnittstellenspezifikation als öffentliches Dokument; ein Einreichungsportal mit Prüfung; Vertrag mit Anteil und Rechteübergang; Abnahmeregeln für Zubehör — Validator plus physischer Test; Verkaufserfassung je Zubehör für die Auszahlungen. Risiko Qualität und Haftung — ein Zubehör mit dem Namen Airholder versagt auf der Straße; geschlossen durch verpflichtende Abnahme und Prüfung vor dem Katalog, nicht danach.

Modell B. Konstruieren im System auf unseren Fahrzeugdaten

Das Wertvollste, was sich ansammelt, sind die Fahrzeugpässe: aus den Aufbau Richtlinien extrahierte Regeln, Scans realer Dächer, Befestigungszonen, Streuung zwischen Exemplaren, Lieferantenkalibrierungen. Als Dateien verkaufen geht nicht: OEM-Daten stehen unter NDA, und eigene Scans, nach außen gegeben, hören auf, ein Vermögenswert zu sein. Aber innen konstruieren lassen geht: ein externer Konstrukteur oder ein Unternehmen erhält einen Desktop im System, sieht den Pass des gewünschten Fahrzeugs, baut sein Produkt gegen unsere Regeln und Prüfungen, und die Daten verlassen den Server nicht. Adressaten: Aufbauhersteller, Camper, Regalsysteme in Kastenwagen, Leitersysteme, Anhänger — dieselbe Physik, und sie haben weder Scans noch extrahierte Regeln noch einen Validator; unabhängige Konstrukteure; Zubehörhersteller zur Kompatibilitätsprüfung. Bezahlung: Arbeitsplatz je Monat plus Zugang zu ausgewählten Fahrzeugpässen.

Juristische Grenze vor dem Start zu prüfen: NDA mit Herstellern beschränken die Nutzung der OEM-Daten meist auf das eigene Unternehmen. Die Arbeit Dritter in unserem System mit OEM-Geometrie auf dem Bildschirm kann die Zustimmung des Herstellers erfordern — auch ohne Export. Solange keine Zustimmung vorliegt, baut Modell B auf eigenen Daten auf: Scans, aus öffentlich zugänglichen Richtlinien extrahierte Regeln, Kalibrierungen, Bibliothek. Das genügt für die meisten Aufgaben; die OEM-Schicht wird je Fahrzeug zugeschaltet, wo die Zustimmung vorliegt.

Modell C. Offenes Ökosystem: Hersteller, Selbermacher, Händler

Der Markt für Lampen und LED-Bars für Geländewagen ist groß und fragmentiert — Dutzende Hersteller (darunter australische, wo diese Kultur stark ist), und jeder sucht, wohin sein Produkt gehört. Unsere Plattform mit beschriebener Befestigungsschnittstelle kann für sie der Ort werden, an dem ihre Lampe besser verkauft wird: «kompatibel mit Airholder, Adapter mit einem Klick». Vier Beteiligungsebenen: Hersteller melden die Schnittstelle ihres Produkts — Lochbild, Masse, Abmessungen, drei Zahlen und ein Bild, kein CAD — und das System erzeugt den Adapter zur T-Nut automatisch aus dem Generator für gebogene Teile. Selbermacher geben Lochbild und Masse ihrer Lampe ein, wählen den Ort auf der Plattform und erhalten eine Datei für die eigene Fertigung: STL für den Druck (PA12, ASA), DXF für den Laser in der nächsten Werkstatt oder die Bestellung bei uns über Blechcon, dazu den Anleitungsschritt und das Validatorurteil. Externe Konstrukteure wie in Modell A. Und Händler und Installateure — kleine Händler, Werkstätten, Ausrüstungsläden, die Panels und LED-Bars verkaufen und dem Kunden montieren: sie haben dieselbe Not, mit der Zahl der Kunden multipliziert, aber weder CAD noch Konstrukteur noch Zeit. Für sie ist das System ein Verkaufs- und Montagewerkzeug: Träger des Kunden wählen, Panel aus dem Katalog wählen, Kompatibilitätsurteil, Adapter, Preis und Anleitung erhalten — und die Montage in einer Stunde verkaufen, statt den

Kunden zum Nachdenken nach Hause zu schicken. B2B, aber ein anderes: nicht Konstruktion, sondern Service am Tresen.

Was niemand sonst hat — die Prüfung vor dem Download: nicht «lade das STL», sondern ein Validator, der Masse des Zubehörs, Werkstoff des Adapters und Ort auf der Plattform kennt: gedruckter Kunststoff hält eine 400-g-Lampe und keinen 20-kg-Kanister — das sagt das System, statt dass der Mensch rät. Daraus die Verantwortungsklassen: zertifiziert (unser, geprüft), Partner (der Hersteller haftet für sein Produkt, wir für den Adapter), selbstgebaut (der Käufer, mit Haftungsverzicht beim Download; nur leichte und nicht sicherheitsrelevante Teile, harte Massengrenze im Validator, visuell und rechtlich von «Airholder» getrennt). Händler arbeiten nur mit den Klassen zertifiziert und Partner.

15t. Programm «Airholder Ready»

Zusammenfassung der Modelle A, B und C in einer Konstruktion: kein Katalog und kein Service, sondern eine lizenzierbare Schnittstellenspezifikation mit Marke. Vorbild ist das Programm «Made for iPhone» (MFi): Apple fertigt fast nichts vom Zubehör, besitzt aber den Standard, prüft die Konformität und vergibt das Zeichen. Ein Ökosystem aus Tausenden Herstellern entsteht um eine Spezifikation, nicht um eine Fabrik.

Lizenziert wird die Plattformschnittstelle als Dokument: Profilquerschnitt, Raster und Geometrie der T-Nut, zulässige Befestigung und Nutensteine, Zonen und zulässige Punktlasten, Höhenbeschränkungen, Anforderungen an die galvanische Isolation, Massenklassen. Bei uns existiert sie bereits als Validatordaten; für das Programm wird sie als öffentliche Spezifikation mit Version formuliert. Dazu das Zeichen selbst — die eingetragene Marke «Airholder Ready».

Stufe	Wer	Was geprüft wird	Was er erhält
Compatible	Hersteller erklärt die Kompatibilität selbst	Schnittstellenprüfung durch den Validator anhand dreier Zahlen: Lochbild, Masse, Abmessungen	Eintrag im Katalog, Adapter aus dem Generator
Ready	Hersteller durchläuft die Abnahme bei uns	Validator + physische Anprobe an der Plattform + Prüfung der Montageanleitung	Zeichen auf Verpackung und Website, Platz im Händlerkatalog, Co-Branding
Certified	Produkte mit Sicherheitsrelevanz: Zelte, Leitern, große Panels	alles Obige + Prüfung nach Familie mit Berechnung; bei Vorliegen das Zeichen «geprüft»	Recht auf Vertrieb über zertifizierte Installateure und an Fuhrparks

Ökonomie: dem Hersteller kostet das Zeichen Geld — Jahresbeitrag je Stufe plus Abnahmegebühr je Produkt; uns kostet es fast nichts, weil die Abnahme aus Validator und einem physischen Test besteht, nicht aus Konstruktion. Der Hauptpreis sind Dachzelte: das teuerste Zubehör auf dem Dach, und ihre Hersteller suchen Plattformen für Kompatibilität. Danach Markisen, Leitern, Duschen, Panels, LED-Bars. Zweite Einnahmequelle: Verkauf von Adaptern und Partnerprodukten über das Händlernetz. Dritte, indirekte: jeder Programmteilnehmer wird zum Vertriebskanal für den Träger — «für unser Zelt braucht es eine Airholder-Ready-Plattform».

Warum es gerade mit unserem System funktioniert: MFi trägt, weil Konformität geprüft und nicht erklärt wird. Bei uns ist die Prüfung ein Validator mit Masse, Ort, Höhe, Schwerpunkt und Galvanik; die Adapterabnahme über drei Kriterien; die Anleitung als Artefakt für jedes Paar «Plattform + Zubehör». Ein Wettbewerber kann ein Zeichen drucken, aber kein Prüfsystem dahinterstellen. Und die Schnittstelle ist konstruktionsbedingt stabil: versioniert, Änderungen rückwärts kompatibel — sonst verliert das Programm seinen Sinn. Grenze: das Programm wird nicht eröffnet, bevor die Plattformschnittstelle für mehrere Jahre fixiert ist — ein Wechsel des Profilquerschnitts entwertet das Zeichen bei allen Teilnehmern. Reihenfolge: nach Modell C mit zwei bis drei Lampenherstellern als Piloten der Stufe Ready; erstes Certified — das Dachzelt.

15u. Fluss statt Codieren: messen und Iterationen verringern

Studie DX (September 2026, 500+ Teams): die KI schreibt mehr als die Hälfte des Codes und spart Stunden, aber Features erscheinen nicht früher — Codieren war nie der Engpass; der Engpass sind Abstimmungen, Reviews, Warten auf Nachbarteams, Nacharbeit nach «wir haben es uns anders überlegt». Bei uns von innen sichtbar: der Reverse-Kern in zwei Tagen, gebremst haben das Warten auf die Ranger-Baugruppe, die Portalantworten, die Entscheidungen zum Datenmodell, das Verschieben von Dateien. Und dieselbe Nebenwirkung: drei Metrologiefehler, schnell geschrieben und von der Realität gefunden.

Messmethode — das Programm Ranger als Pilot: Tore mit Zeitstempeln: Fahrzeugdaten erhalten → Pass fertig → erstes STEP des Halters → drei Kriterien bestanden → ins Portal geladen → Portalantwort → Bestellung → Teil in Händen → geprüft → im Shop. Das Projektjournal schreibt die Stempel für Operationen im System; für externes Warten genügt eine Handzeile. Je Stufe zwei Zahlen: Arbeitszeit und Wartezeit; für das Programm der Anteil der Arbeit an der Gesamtdauer (in der typischen Entwicklung 10–20 %). Vor der Messung festgehaltene Prognose: die Zeit im Werkzeug ist ein kleiner Anteil; das Meiste sind Portalantworten, Warten auf Daten, Entscheidungen, Fertigungsdauer. Bestätigt sich das, muss nicht das Reverse beschleunigt werden, sondern diese Stellen. Zum Vergleich den Defender aus E-Mails und Dateien rekonstruieren: die Differenz ist die nach einer Methodik gemessene Antwort auf die Frage der Sinnhaftigkeit.

Mechanismus	Wie er wirkt	Kennzahl
1. Prüfungen nach links verschieben	eine Iteration über das Portal dauert einen Tag, dieselbe Prüfung im Validator eine Sekunde; jede Antwort «nicht erkannt» wird zur Regel (Spline in der Biegezone, Freischnitt nicht nach Typ, Dicke nicht streng); nach zwanzig Teilen kennt der Validator die Ablehnung vor dem Portal	Anteil der Teile, die das Portal beim ersten Mal bestehen: derzeit 2 von 4, Ziel 9 von 10
2. Vorhersage aus dem Journal	welche Parameter nach der ersten Baugruppe am häufigsten geändert werden (Höhe, Position, Querträgerraster) — sie werden Parameter erster Ebene, unveränderte werden fixiert; Operationsketten (nach «anheben» in 90 % «Abstand» und «Anleitung») schlägt das System vor oder führt sie als Makro aus	Anteil der durch Makros abgedeckten Änderungen
3. Varianten im Voraus statt einzeln	Nachtlauf: Höhe 20/30/40, Flanschbreite — morgens eine Tabelle mit Masse, Preis, Urteilen; die Entscheidung wird zur Auswahl aus Fertigem; ans Portal im Stapel — parallele Aufklärung statt sequentieller Iterationen	Anzahl der Operationen je Teil bis «tauglich»
4. Reserve dort, wo Änderungen von außen kommen	die Streuung zwischen Fahrzeugen aus Scans bestimmt den Verstellbereich des Langlochs; Regel «Verstellweg $\geq 2 \times$ Streuung» — Vorhersage durch Konstruktion statt Nacharbeit	Feldmeldungen «passt nicht» je Serie

Das Billigste sind Entscheidungen vor der ersten Linie: die meisten Nacharbeiten entstehen durch «wir haben es uns anders überlegt» bei Entscheidungen, die vor der Geometrie zu treffen waren — Befestigungspunkte, Schnittstelle, Höhe über Dach, Länge und Breite, Fahrzeug und Variante. Eine Liste von fünf Punkten, unterschrieben, bevor FreeCAD geöffnet wird. Eine spätere Änderung ist dann ein bewusster Akt mit einem Preis, den das System zeigt, kein Zufall. Alle Kennzahlen ergeben sich aus dem bereits vorhandenen Journal — nichts Neues muss erhoben werden, man muss anfangen hinzusehen.