PRESENTATION FOLDER

Visual Technic Studio

3D Technical Animation & Simulation
for education and professional training

Document version	2.0
Software status	VTS Studio 2.4.x (development build)
Publisher	DC Software Engineering
Date	August 2026

This is the English edition of the presentation folder. It replaces the 2026 edition and adds the chapters 02 The tool family, 03 VTS Composer, 04 The generators, 05 Kinematics and 06 Standards compliance. Every figure quoted here was counted in the current source; none of the numbers are estimates.

Daniel Zeimentz | DC Software Engineering
visualtechnicstudio.com | dc-software-engineering.com

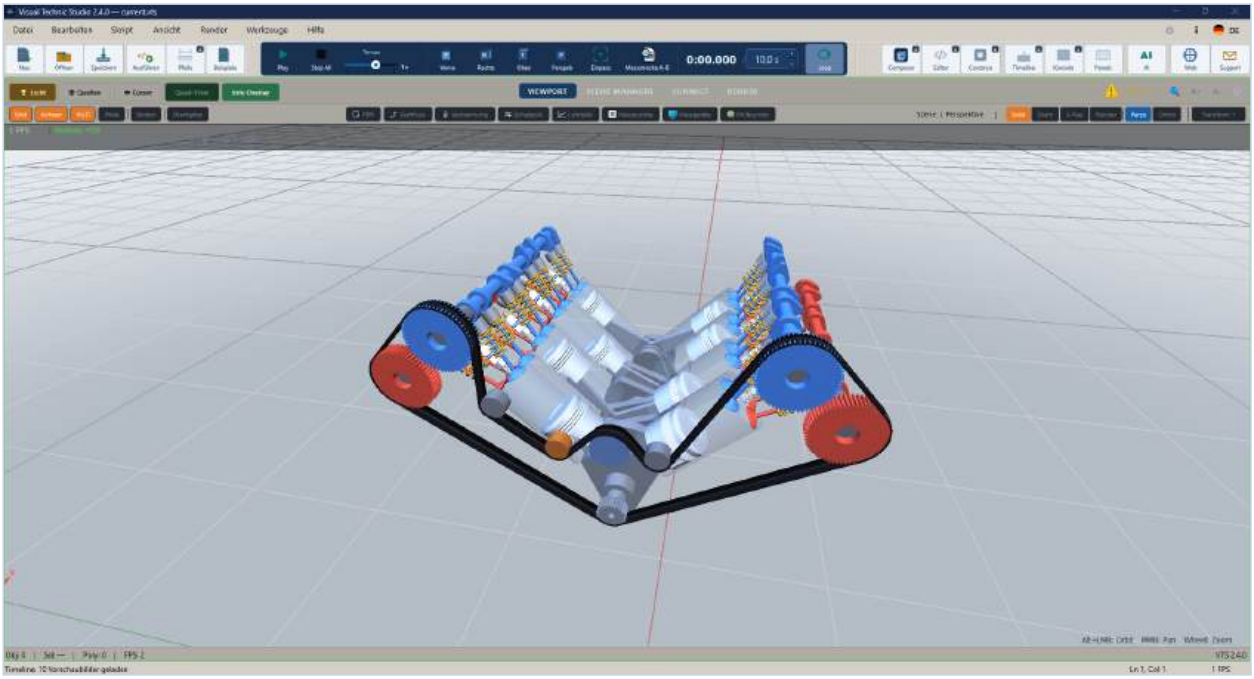
Contents

01

The software

Visual Technic Studio (VTS) is a cross-platform desktop application for 3D technical animation and simulation. Technical processes — engines, gearboxes, hydraulics, pneumatics, electrics, mechanisms — are modelled precisely, animated and prepared for teaching.

> The core idea: make technical relationships visible and understandable — for lessons, for training and for professional visualisation.



VTS Studio — a V8 petrol engine with two toothed-belt timing drives. Crank train, valvetrain and timing drive run in sync; the crankcase is switched to translucent so that pistons, conrods and crankshaft stay visible.

Key features

- Precise 3D modelling with CAD integration (OpenCascade/OCCT, STEP import)
- Professional animation timeline with a keyframe editor
- VisualTechnicScript — a Python API with 644 commands in 54 modules
- Real-time preview and rendering directly in the 3D viewport
- ReactPhysics3D engine with a unit system: every quantity carries its unit, a bare number is rejected
- Interactive control panel: sliders, buttons and gauges on the running model
- Data driven: measurements, CSV files and sensor data become animations
- Didactic tooltip and safety-notice system with context-aware explanations
- Multilingual interface, switchable at runtime without a restart
- Export: .vtsanim, MP4 video, PDF report, CSV/PNG, DXF/SVG (via LineWorks)

The twelve technical disciplines VTS covers — from the engineering drawing to spaceflight.

Technology used

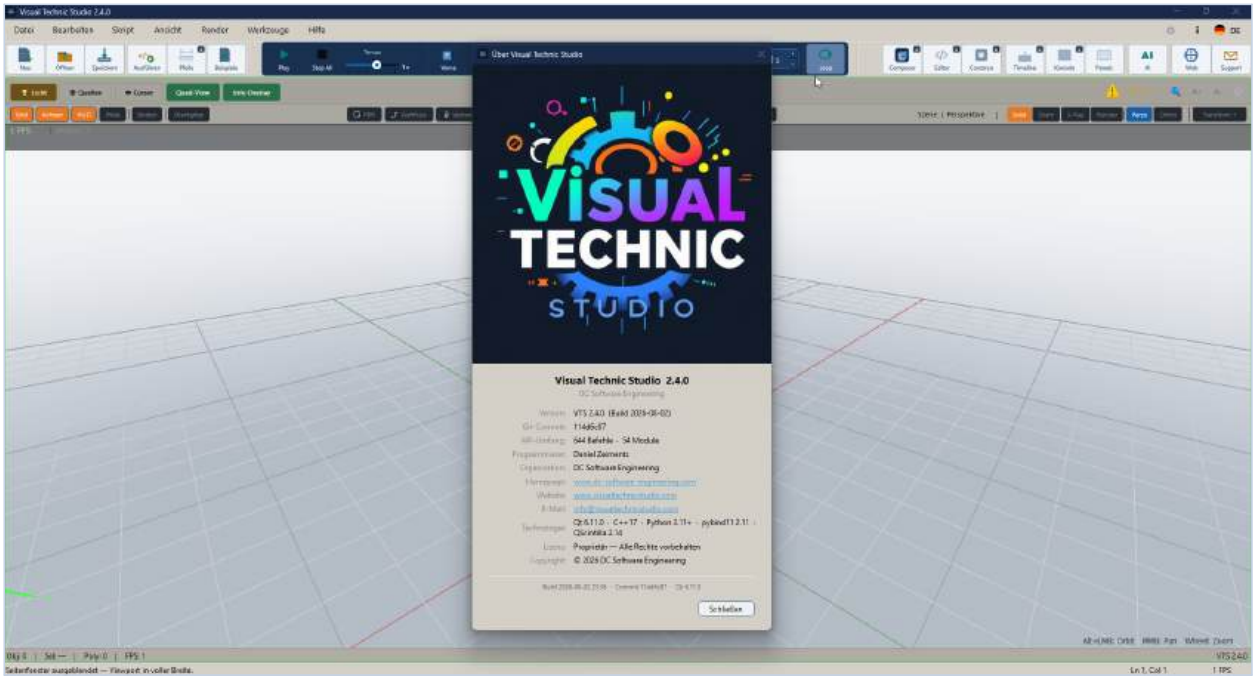
Area	Implementation
Core	Qt 6.11 / C++17 — built to be cross-platform
Scripting	Python 3.11+ via pybind11 2.11 (VisualTechnicScript)
Script editor	QScintilla 2.14: syntax highlighting, auto-completion, live error checking
Physics	ReactPhysics3D (Newton-Euler) with the unit system (pint)
CAD	OpenCascade/OCCT 7.7 — STEP/IGES import, hidden-line removal
FEM	Eigen 3.4+
Rendering	real-time 3D viewport, PBR (Cook-Torrance), HD render path
Asset exchange	glTF 2.0 / Assimp; output format .vtsanim (signed ZIP)
Licensing	RSA-2048 signatures on packages; JWT RS256 with a module bitmask

Platform figures

Every number below was counted in the current source. None of them are targets.

Figure	Value	Where it comes from
Commands in the Python API	644	54 modules
Geometric primitives	67	schema.PRIMITIVES

Figure	Value	Where it comes from
Standard parts in the Composer catalogue	157	machine_elements.CATALOG
Entries in the kinematics library	243	dof.DOF_LIBRARY
Convention domains (DIN/ISO)	23	conventions.CONVENTION_DOMAINS
Test files	565	Animation Core/tests
Demonstration scripts	118	VISUAL TECHNIC SCRIPT
Technical disciplines covered	12	mechanics to medical technology



The About dialogue states the command count, the module count, the build and the commit — the figures in this folder can be read off inside the product itself.

VisualTechnicScript — what a script looks like

Every physical quantity is passed as Q(value, unit). That is not a formality: a module in millimetres and a module in inches look identical as bare numbers, and that particular mistake is expensive in teaching. Forget the unit and you get an explanation, not a crash.

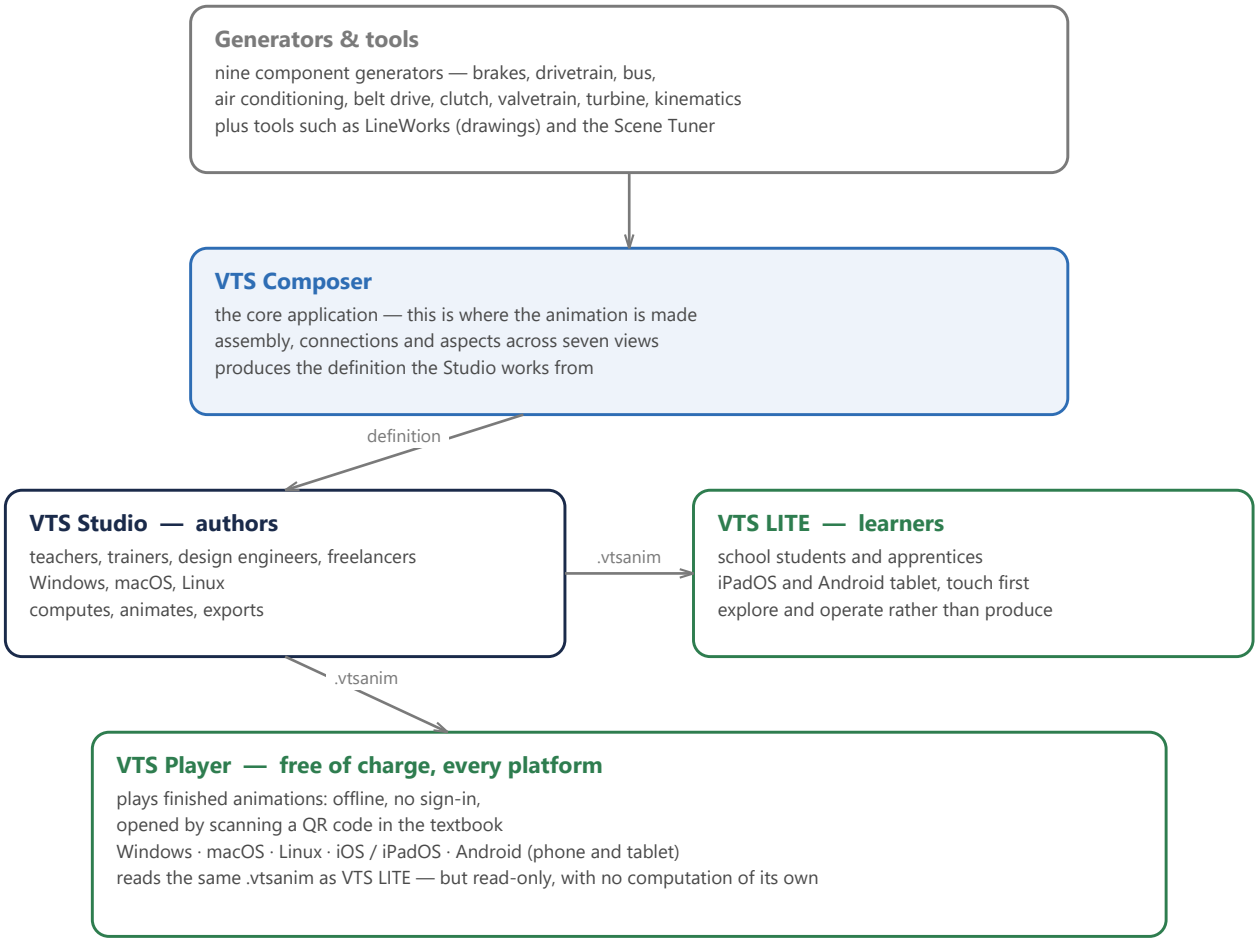
```
from vts.units import Q
from vts.mechanics import CreateGear
from vts.animate import CreateTimeline, AnimateRotation, SyncAnimations

pinion = CreateGear(teeth=18, module=Q(2.5, "mm"))   # true involute, DIN 867
wheel  = CreateGear(teeth=36, module=Q(2.5, "mm"))
wheel.position = (Q(67.5, "mm"), 0, 0)               # centre distance m*(z1+z2)/2

tl = CreateTimeline(duration=Q(10, "s"), fps=60, loop=True)
a1 = AnimateRotation(pinion, axis="z", speed=Q(120, "rpm"))
a2 = AnimateRotation(wheel,  axis="z", speed=Q(-60, "rpm"))   # i = 2
SyncAnimations([a1, a2], mode="parallel")
```

> The tooth profile above is a true involute to DIN 867, not a sine wave with teeth. That distinction runs through the whole product and is written down as a binding rule — see chapter 06.

02

The tool family

VTS is no longer one program but a family with a clear division of roles. The Composer is the core application: that is where the animation is made. The Studio is the authoring application for teachers, trainers and design engineers; VTS LITE is its counterpart for learners on a tablet. Generators and tools — among them the kinematics generator and LineWorks — live inside the Composer and are launched from there. Finally the VTS Player brings finished animations to every platform, free of charge.



The tool family: the tools feed the Composer, the Composer feeds its definition to the Studio — and the Studio produces the signed package that VTS LITE reads on the tablet and the free Player reads on every platform.

Who does what

Tool	Role	Result	Who works with it
VTS Composer	Core application — declare assembly, connections and aspects; launches the generators and tools	definition for the Studio	author, subject teacher
VTS Studio	Authoring application — compute, animate, render, export	scene, MP4, PDF, .vtsanim	teacher, trainer, design engineer, freelancer
VTS LITE	Learner application — explore and operate on a tablet	— (reads .vtsanim)	school students, apprentices
VTS Player	Play finished animations — free on every platform: Windows, macOS, Linux, iOS/iPadOS, Android (phone and tablet)	— (read-only)	learners, textbook readers
Generators & tools	launched inside the Composer: nine component generators, the kinematics structure generator, LineWorks (drawings), the Scene Tuner	parts, topology, drawings	author, subject teacher

Why separate tools

The obvious route would have been to put everything into the Studio. Three reasons argue against it, and practice has borne them out:

- A different way of thinking, a different interface. Describing an assembly means working with lists, diagrams and connections — not with a 3D viewport and a timeline.
- Different users. A subject teacher building a brake diagram does not need a script editor. A publishing editor deriving a drawing does not need a physics engine.
- Different delivery. A tool such as LineWorks needs no interface at all and runs as a pure command-line step inside a publishing pipeline; the Studio cannot do that and should not try.

> What is shared is the core, not the interface. All the tools read and write the same declarative data layers — kinematics, conventions, materials, flow, control panel. Every quantity has exactly one source; a second route to the same answer is explicitly forbidden.

What all the tools share

The shared layers are pure data — no evaluated expressions, no hidden logic. That is precisely why the Player can read them without carrying Python, and why they can be checked without a screen.

Layer	What it records	Who uses it
Kinematics	which movement a part may make, about which axis, within which range, in which direction	Composer, Studio, Player, LITE
Conventions	23 domains of engineering rules — bend radii, bolt patterns, terminals, identification colours	Composer, Studio
Materials	colour, roughness and metalness per material — and, kept separate, the line colours of a drawing	Composer, Studio, LineWorks
Flow	what flows where, how much, how hot — drawn as a band with an arrowhead inside the pipe	Composer, Studio, Player
Control panel	which controls exist, what they set, what is coupled and what is locked	Composer, Studio, Player, LITE
Aspect list	which aspects a document can carry at all	Composer

The route an assembly takes

1. Topology — the kinematics generator fixes links, joints and driving variables before any geometry exists.
2. Assembly — the Composer fills it with parts: catalogue, generator or the customer's own CAD.
3. Aspects — circuit diagram, fluid diagram, measuring points, control panel, sequence and flow are added; the status lamps show what is still missing.
4. Checking — kinematics and conventions are validated, and for machine-generated assemblies a violation is rejected outright.
5. Handover — the Composer writes the sidecar file; the Studio computes, animates and renders.
6. Output — MP4 for the whiteboard, PDF for the worksheet, DXF for the workshop, .vtsanim for the Player.

03

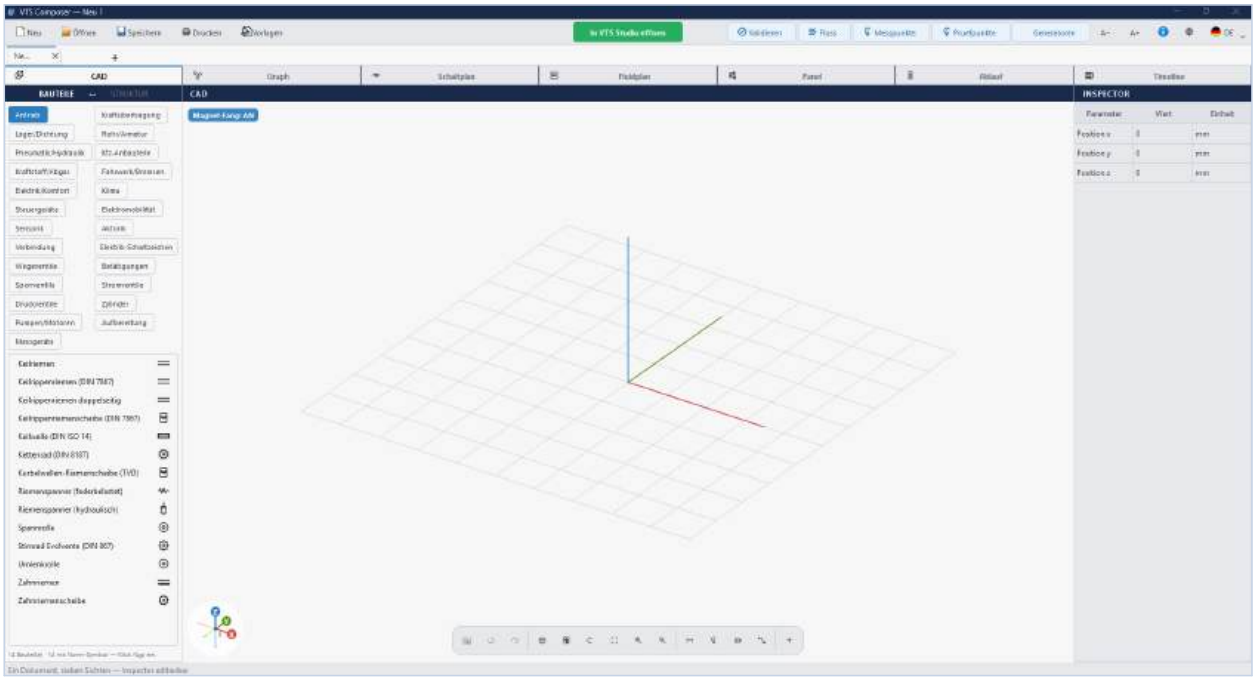
VTS Composer

The Composer is the authoring tool for assemblies. It is where you describe what a system is — which parts, connected how, what may move and in what way, what gets measured, what can be operated from the control panel. The Studio turns that into an animation. The Composer is emphatically not a CAD program: it does not construct geometry, it arranges it and gives it meaning.

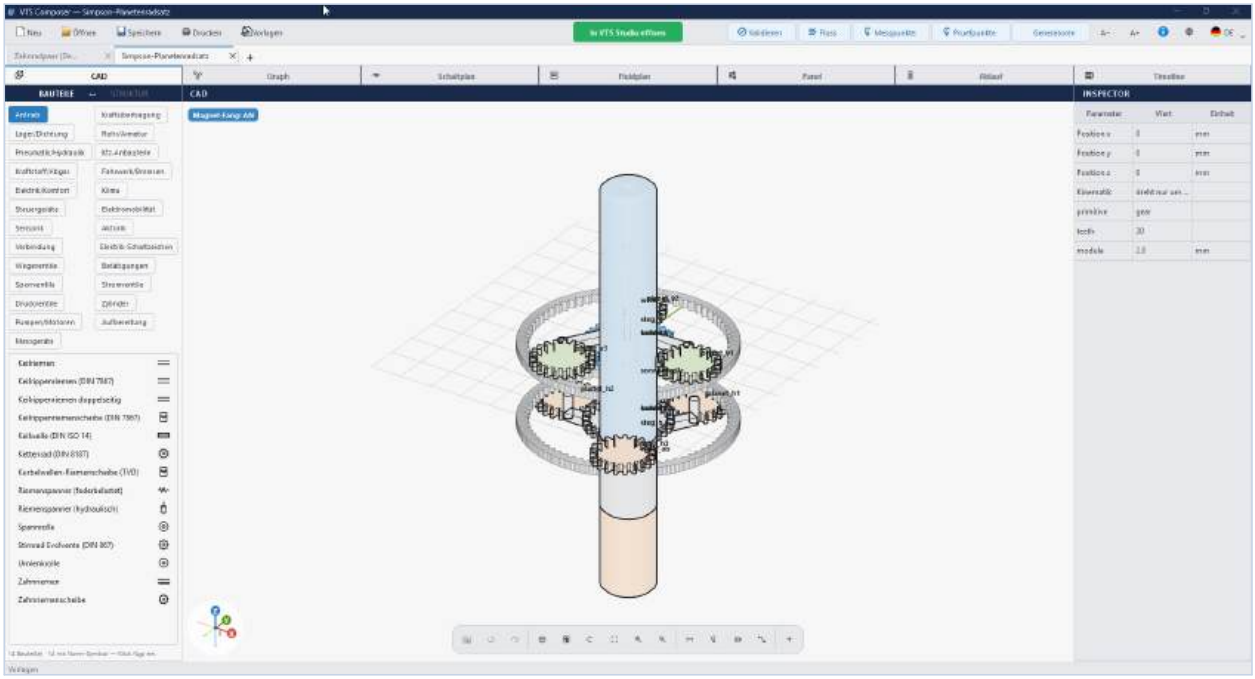
Seven views onto one document

An assembly has several faces. Rather than splitting it into separate files, the Composer shows one document in seven views — each view edits a different aspect of the same thing.

View	What is edited there	Standards
CAD	spatial arrangement of the parts, dimensions, position	—
Graph	cause and effect: what drives what, couplings, flow (matter/current/force)	—
Schaltplan	electrics and electronics as a self-contained 2D diagram with test points and instruments	DIN EN 61082, DIN 72552
Fluidplan	pneumatics and hydraulics as their own diagram with directional valves, cylinders and restrictors	ISO 1219-1
Panel	control panel: sliders, buttons, displays and lamps — placed graphically	—
Ablauf	sequence chart: what happens in which order	—
Timeline	the animation over time	—

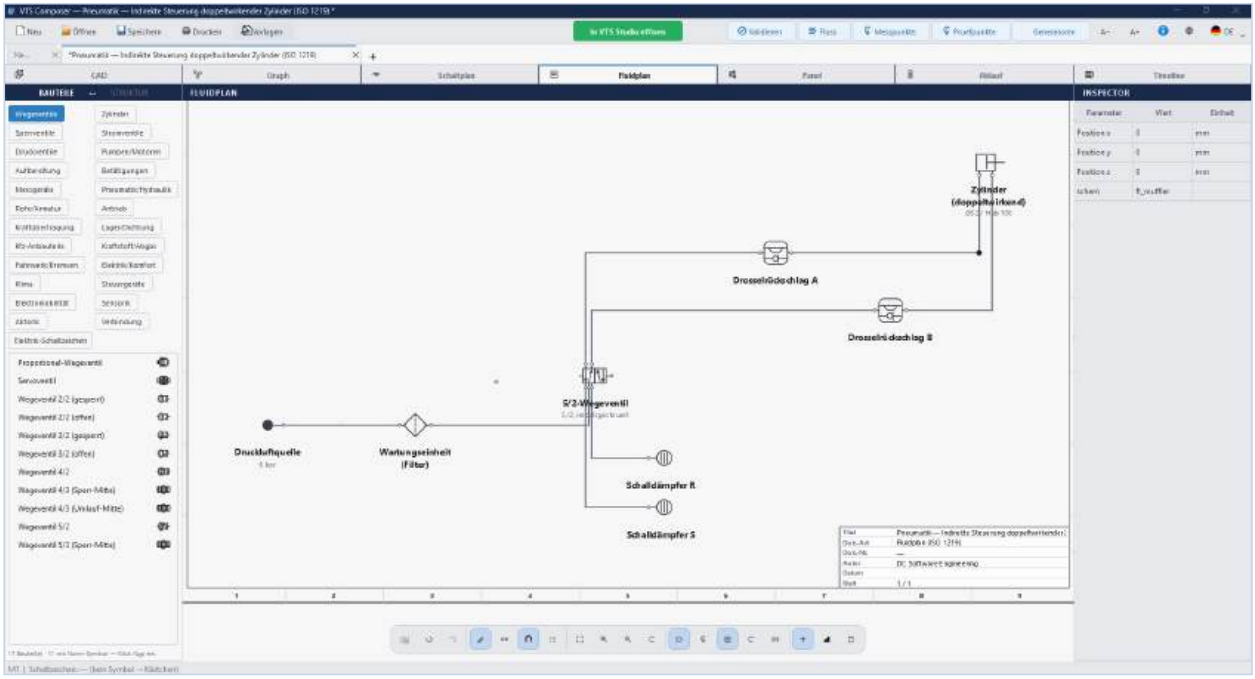


One document, seven views: the tabs along the top switch between CAD, Graph, Schaltplan (circuit diagram), Fluidplan, Panel, Ablauf (sequence) and Timeline. The part catalogue on the left and the inspector on the right stay in place in every view. The interface is shown in German; it switches to English at runtime.



Composer, CAD view: the part catalogue by domain on the left, the assembly in the middle (a Simpson planetary gear set), the inspector on the right — its Kinematik row shows read-only which movement the selected part is allowed to make at all.

Circuit diagram view: a wiring diagram to DIN 72552 with current-path numbering, terminal designations and a title block — a document in its own right, not a projection of the 3D parts.



Fluid diagram view: a pneumatic circuit to ISO 1219 with a 5/2 directional valve, throttle check valves and a double-acting cylinder. The flow paths drawn in the valve follow the port matrix — an arrow floating inside the box is not a flow path.

Twelve aspects — and a display that says what is missing

A document can carry twelve aspects. Below the inspector sits a table with one lamp per aspect: parts, connections, circuit diagram, test points, instruments, measuring points, control panel, panel layout, flow, sequence, timeline, teaching board. Clicking a row jumps to the view responsible for it.

> The lamp is an inventory, not a verdict. It says „present", never „correct". Whether a diagram complies with a standard is what the convention check reports; whether a movement is admissible is what the kinematics check reports. That sounds like a nicety and is the difference between an honest display and a reassuring one.

Parts: a catalogue, not a box of shapes

- 157 standard parts in the catalogue — gears (DIN 867), sprockets (DIN 8187), rolling bearings (DIN 625/720/617/635), belt pulleys (DIN 2211), splined shafts (DIN ISO 14), pipes and flanges (DIN 2633), seals (DIN 3771), circlips (DIN 6799) and more
- 67 primitives as the geometric building blocks the catalogue is made from
- Every part brings its admissible degree of freedom (chapter 05) and its material appearance
- The customer's own CAD geometry can be attached as STEP or glTF

> A part is never assembled from raw boxes and cylinders. A gear that looks like a toothed ring from close up but carries no involute is worse than no gear at all in teaching — it teaches exactly what has to be untaught later.

Further working aids

- Document tabs: several assemblies open at once, each with its own undo stack; assemblies can be merged
- Control-panel designer: place controls by dragging, with grid and snapping — slider, switch, selector, dial gauge, button, indicator lamp, display, text

- Floating tool dock and adaptive level of detail for large assemblies
- Illustrations per variant: every generator variant shows a line drawing derived from the generator data — a picture drawn alongside would go stale unnoticed

Making flow visible

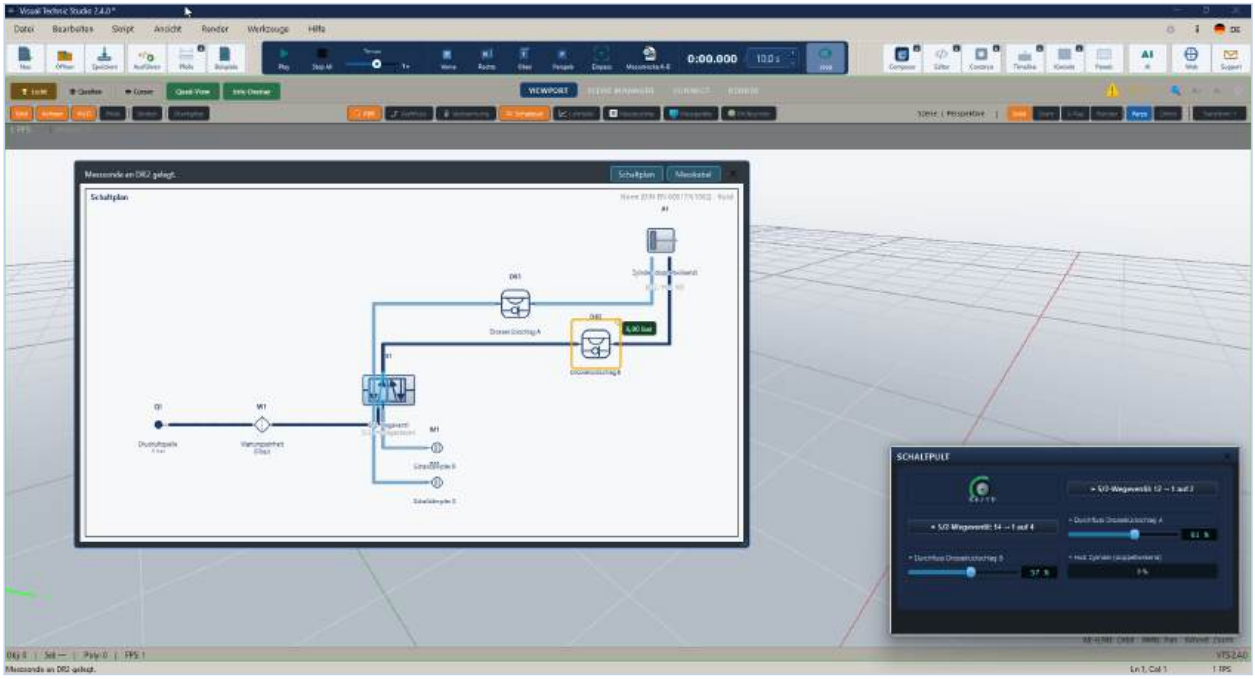
Flow, current and force are shown by one convention across all VTS tools — not reinvented per demonstration. That matters in teaching: a learner should learn to read the picture once and recognise it everywhere afterwards.

Representation	Meaning
a band with an arrowhead inside the pipe, following its shape	direction of flow — in the line, not beside it
width of the band	quantity: volumetric flow or current
colour of the matter band	temperature: cold blue/cyan to hot orange/red
current in two modes	conventional (plus to minus) and physical (minus to plus) — switchable, because both are taught
symbol for the state of matter	three waves liquid, cloud gaseous, solid block solid — the symbol colour is again the temperature
voltage and resistance: not a flow	they do not flow and are therefore not drawn as a band, but read off instruments — voltmeter, ohmmeter, multimeter, oscilloscope

> That last point looks like hair-splitting and is the heart of the matter: an animation that lets voltage „flow" creates exactly the misconception that is hardest to undo in electrical teaching.

Measuring instead of asserting

Test points can be placed in the circuit and fluid diagrams — the places a technician would actually clip onto — and instruments connected to them. The value shown comes from the model, not from a text box. That turns the animation into an exercise: attach the meter, read it, judge it.



Studio: the fluid diagram overlaid on the 3D scene, the probe sitting on the throttle check valve and reading 6.00 bar. The control panel beside it sets valve position, throttle opening and cylinder stroke — the reading follows the operator, it is not written in.

The handover to the Studio

On deployment the Composer writes a companion file — a sidecar. The Studio reads that file and only that file, never the Composer document itself. The boundary is deliberate: it allows the Composer to evolve without touching the Studio, and it makes fault-finding unambiguous. If a change does not show up in the Studio, the first thing to check is whether it was written into the companion file at all.

The route into the Studio runs through one gate that makes sure two tools never write into the same directory at once, and that it stays traceable which build is deployed.

What the Composer deliberately is not

- × Not a CAD program — it does not construct freeform surfaces
- × Not a simulation environment — it describes, the Studio computes
- × Not a drawing program — LineWorks is there for that
- × Not a delivery format — what reaches the learner is the .vtsanim

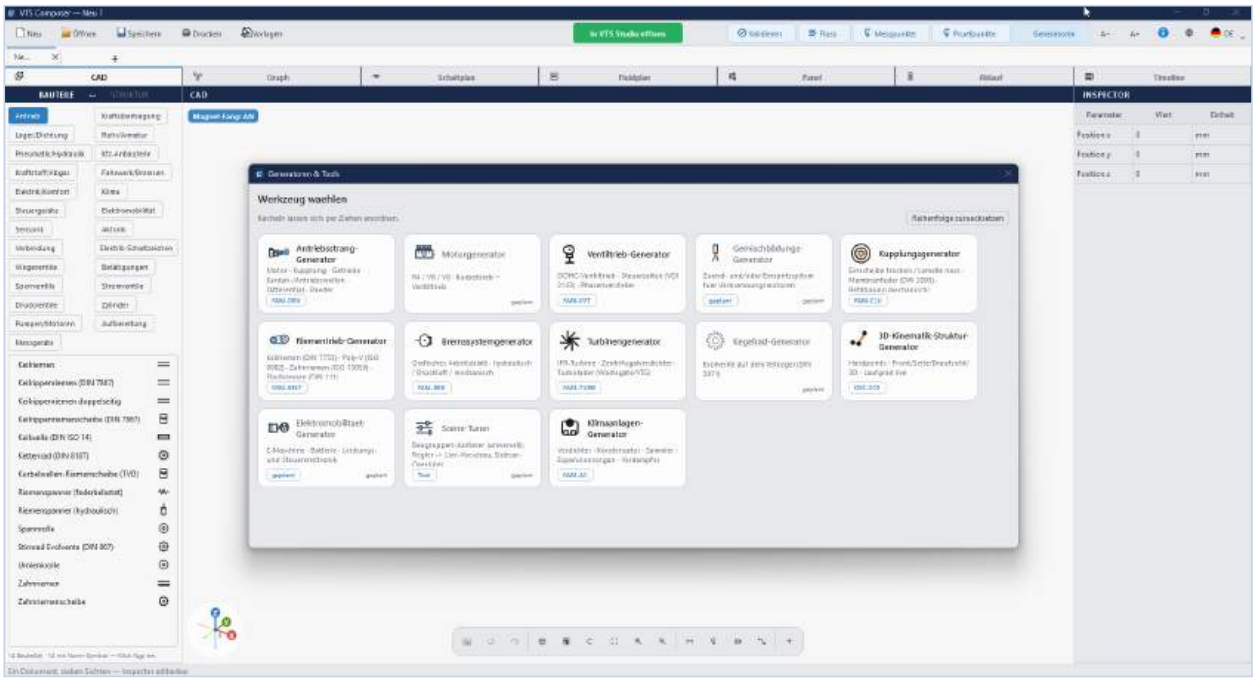
> This keeps the Composer usable. A tool that can do everything demands that a subject teacher learn everything — which is where authoring systems in education routinely fail.

04

The generators in the Composer

A generator builds a complete assembly from a handful of engineering inputs. You do not pick parts out of a box; you answer the questions a design engineer would answer — which brake circuit split, which belt type, which axle layout — and you get the matching, connected, standards-compliant system.

> The difference from drawing: a drawn belt drive looks like a belt drive. A computed belt drive has the right belt length, the right angles of wrap and the right direction of rotation — even after you enlarge one pulley. That is the difference between illustration material and a teaching instrument.

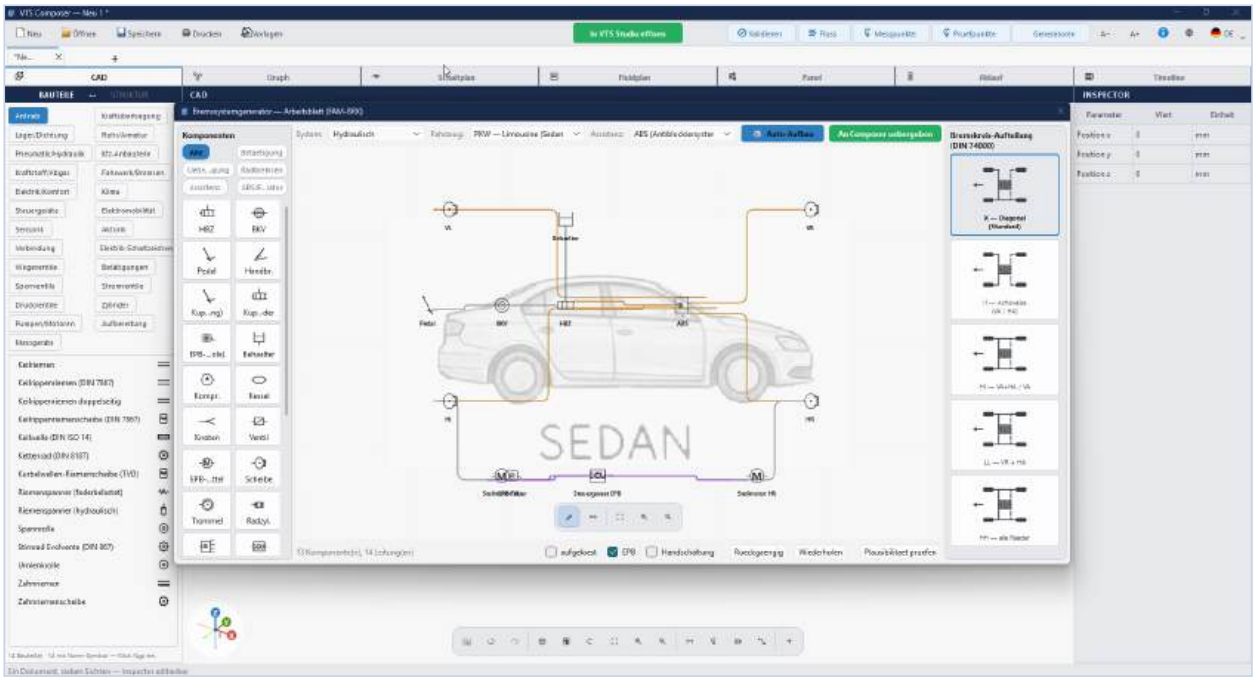


The tool launcher in the Composer: one tile per generator, with its identifier and the governing standards in the subtitle. Announced but not yet released generators are shown as well — marked „planned" rather than left out.

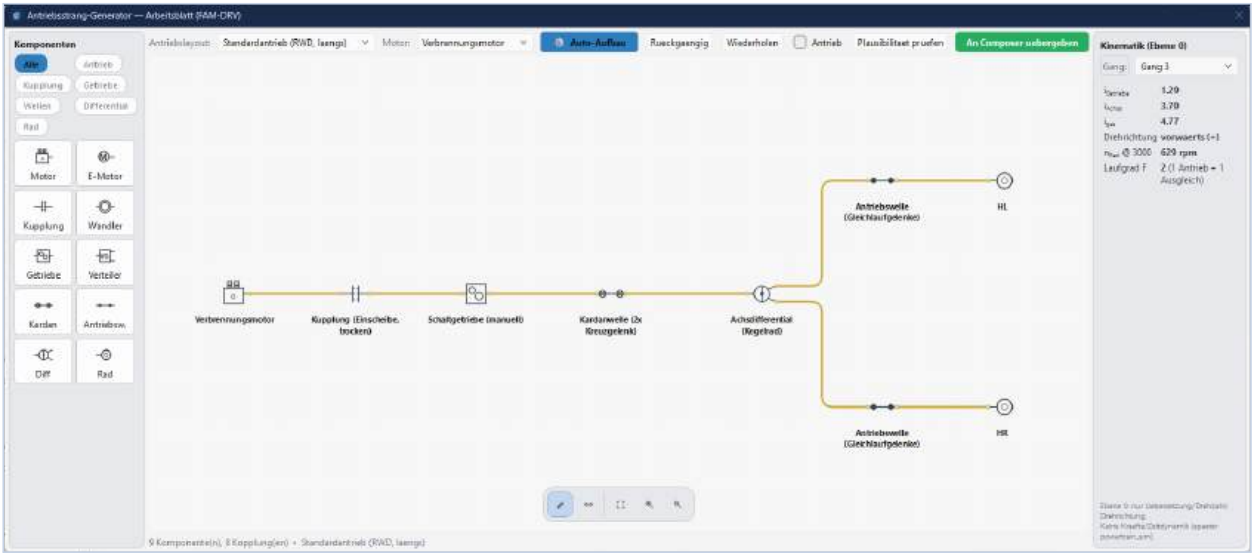
The nine active families

ID	Generator	What it produces	Standards
FAM-BRK	Brake system	the complete system: pedal, booster, master cylinder, ABS/ESP modulator (also resolved into individual valves), disc and drum brakes, parking brake — hydraulic, air or mechanical	DIN 74000, ECE-R13/R13H, ECE-R90, ISO 611, ISO 1219-1
FAM-DRV	Drivetrain	engine, clutch, gearbox, transfer case, propshafts and driveshafts, differential, wheels — with ratio, direction of rotation and mobility per gear	DIN 808 (universal joint), DIN 3971 (bevel gear), Roloff/Matek
FAM-BUS	Bus systems	CAN / CAN FD / CAN XL, LIN, FlexRay, MOST, Automotive Ethernet, diagnostics (OBD/UDS/DoIP) — topology, bus load, arbitration, 3D cable routing through the vehicle	ISO 11898, ISO 17987, ISO 17458, ISO 21806, ISO 14229

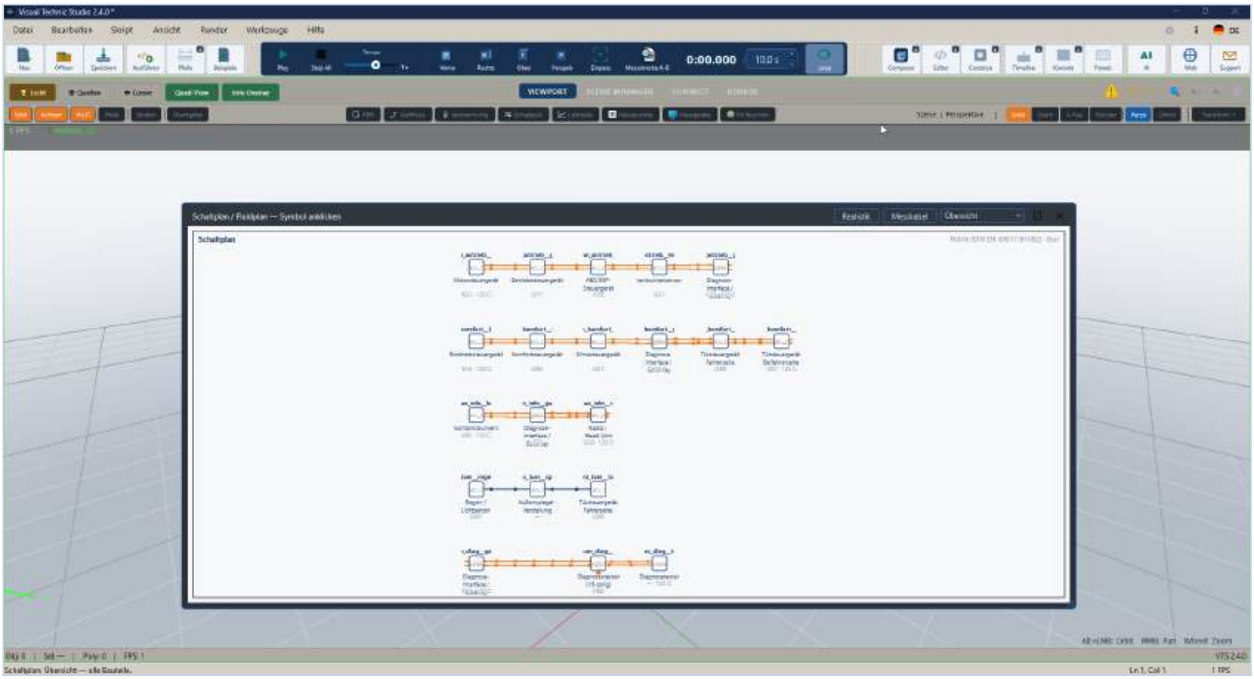
ID	Generator	What it produces	Standards
FAM-AC	Air conditioning	compressor, condenser, receiver, expansion device, evaporator — each port carries its pressure side, so a circuit wired backwards is found even though it looks connected	DIN EN 378, ISO 1219-1
FAM-BELT	Belt drive	from a two-pulley drive to a serpentine accessory drive with tensioner; belt length, wrap and speeds fall out of the geometry	DIN 7753, ISO 9982, ISO 13050, DIN 111
FAM-CLU	Clutch	single-plate dry and wet multi-plate clutch, diaphragm spring, mechanical/hydraulic/concentric actuation, self-adjusting	DIN 2093 (disc spring)
FAM-VVT	Valvetrain	camshaft, valve timing, valve lift; phaser, cam switching and cylinder deactivation	VDI 2143 (motion laws)
FAM-TURB	Turbine / charger	radial turbine, centrifugal compressor, turbocharger with wastegate or variable turbine geometry — blades from the design calculation, not drawn	Euler turbine equation, velocity triangles
DEC-072	3D kinematics structure	the topology of a multibody assembly: hardpoints, joints, links, two-point links — with a live mobility count (see chapter 05)	DIN 808, DIN 71802, DIN ISO 12240



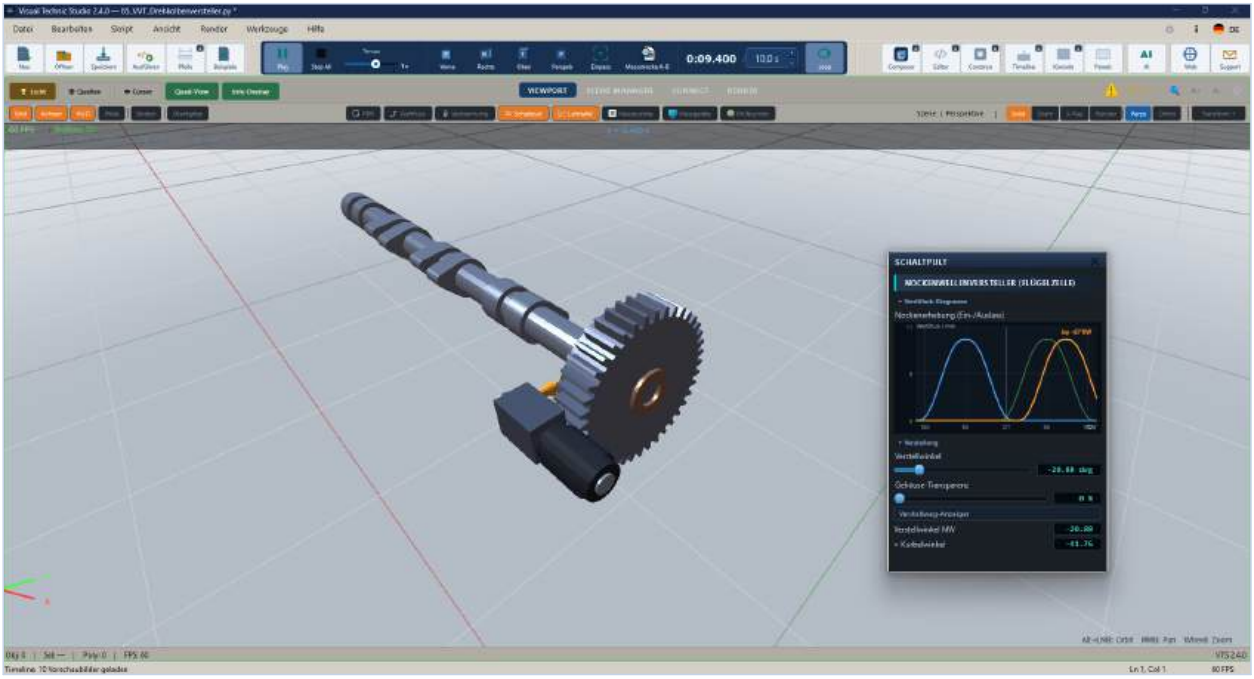
FAM-BRK — the brake system generator's worksheet: the vehicle silhouette as a backdrop, and on the right the five circuit splits to DIN 74000 as pictograms. One click builds the complete, connected system.



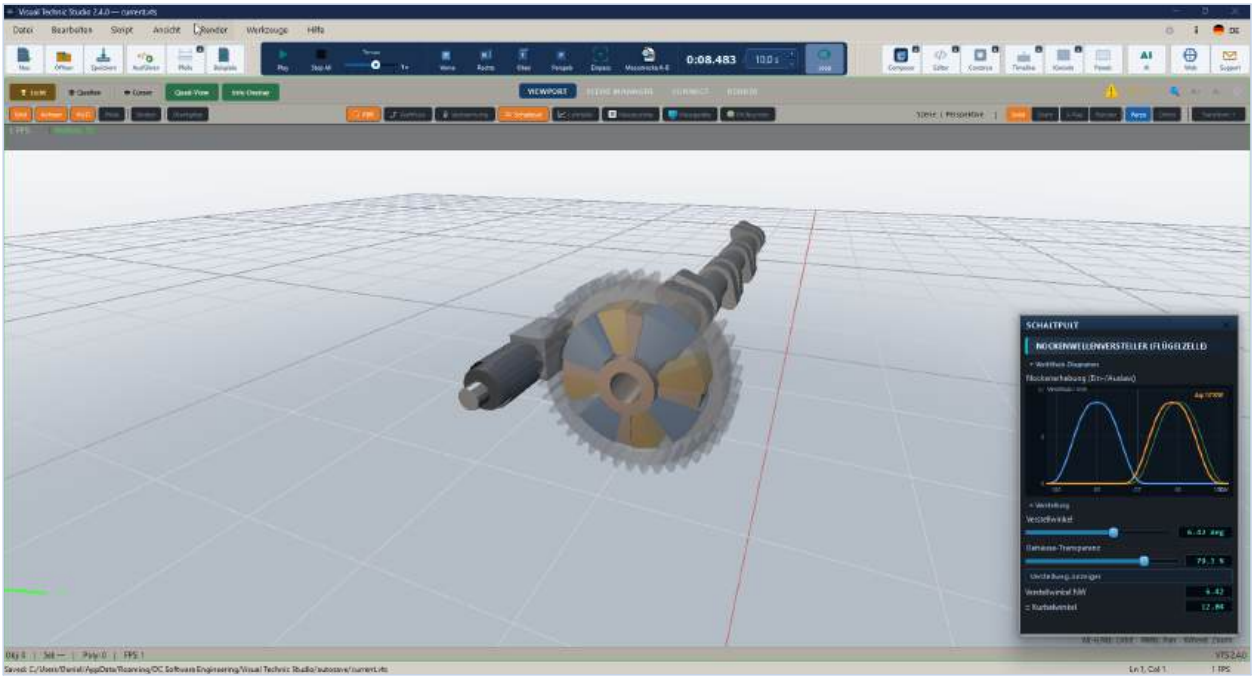
FAM-DRV — drivetrain: the kit on the left, the assembled chain engine→clutch→gearbox→propshaft→differential→wheels in the middle, and the live kinematics on the right. For the selected gear it shows ratio, direction, wheel speed and mobility — and the note that this is model level 0: no forces, no time-domain dynamics.



FAM-BUS — the same vehicle network in the Studio: control units on powertrain CAN, comfort CAN, infotainment CAN and LIN, joined through the diagnostic interface; the 120 Ω terminations sit at the ends of the line, where they belong.



FAM-VVT — a vane-type phaser with the housing closed: this is what the part looks like on the engine. Phase angle −20.88° camshaft, shown in the valve-lift diagram as Δφ −42° crank angle.



The same assembly with the housing at 79 % transparency and adjusted the other way (+6.42° cam, Δφ +13° crank): stator, rotor and the four vane chambers are now visible. The valve-lift diagram follows the slider — it is computed from the same lift curve as the cam contour, not drawn beside it.

Where this is going

The range of generators will be extended and adjusted up to the final 3.0 release — which families are added is decided by demand from practice. Currently in progress or planned:

- Mixture formation — ignition and injection systems for combustion engines
- Electric mobility — e-machine, battery, power and control electronics

- Engine — inline-four, V6 and V8 as a crank train with valvetrain
- Bevel gear — involute on the pitch cone to DIN 3971

> Other occupations get their own generators. Today's family grew around automotive engineering because that is where the first demand was. For metalworking, electrical engineering and further occupational fields, dedicated generators will follow — on the same data layer and with the same standards binding, but with the parts, diagrams and vocabulary those trades expect.

How a generator comes about

Before the first line of code there is a one-page brief: purpose, explicit non-goals, model level (what is modelled and what deliberately is not), the single source of truth, the invariants together with the regression test planned for each, and the boundary to the neighbouring tools. That is not bureaucracy but the lesson from a concrete case: a fluid animation grew purely reactively and had to discover its modelling rules one mistake at a time.

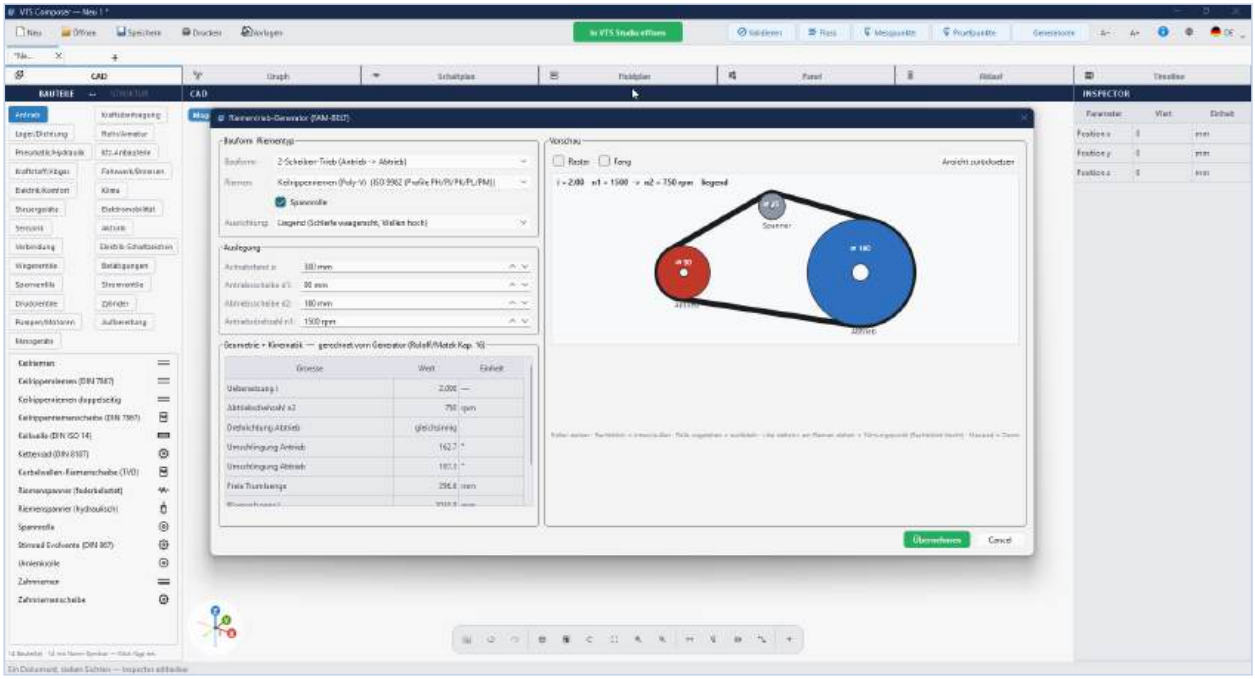
Only then is it built — and the conformance tests are written with the code, not afterwards. Every requirement taken from a standard has a test that checks it by an independent route.

Example: a belt drive

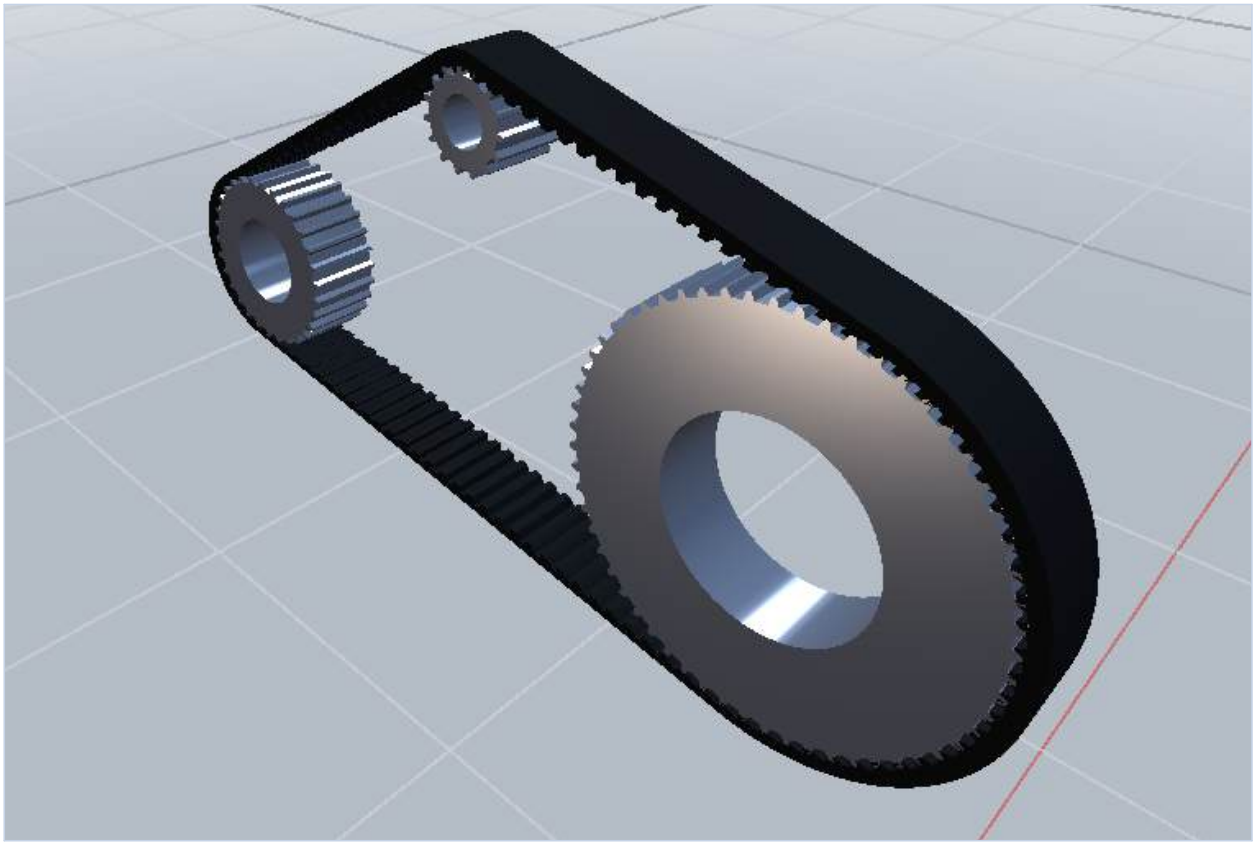
What the author supplies is a handful of engineering quantities. What comes out is a complete, self-consistent assembly — and it stays consistent when one quantity changes.

The author supplies	The generator computes from it
belt type (V-belt, poly-V, toothed, flat)	profile and pulley shape to the relevant standard
pulley diameters and centre distance	angles of wrap, span lengths, total belt length
input speed	ratio, output speed and direction of rotation per pulley
tensioner yes/no, position	belt path around all pulleys as outer tangents with wrap
orientation, flat or upright	spatial position, so the drive fits a horizontal crankshaft

> This is verified against an independent oracle: the computed belt length must equal the perimeter of the convex hull of all pulley circles, and the span length must satisfy Pythagoras. A test that simply evaluates the same formula again would repeat every mistake.



FAM-BELT — the belt drive generator: the inputs on the left (belt type, diameters, centre distance, tensioner), the computed preview on the right with belt length, wrap and speeds.

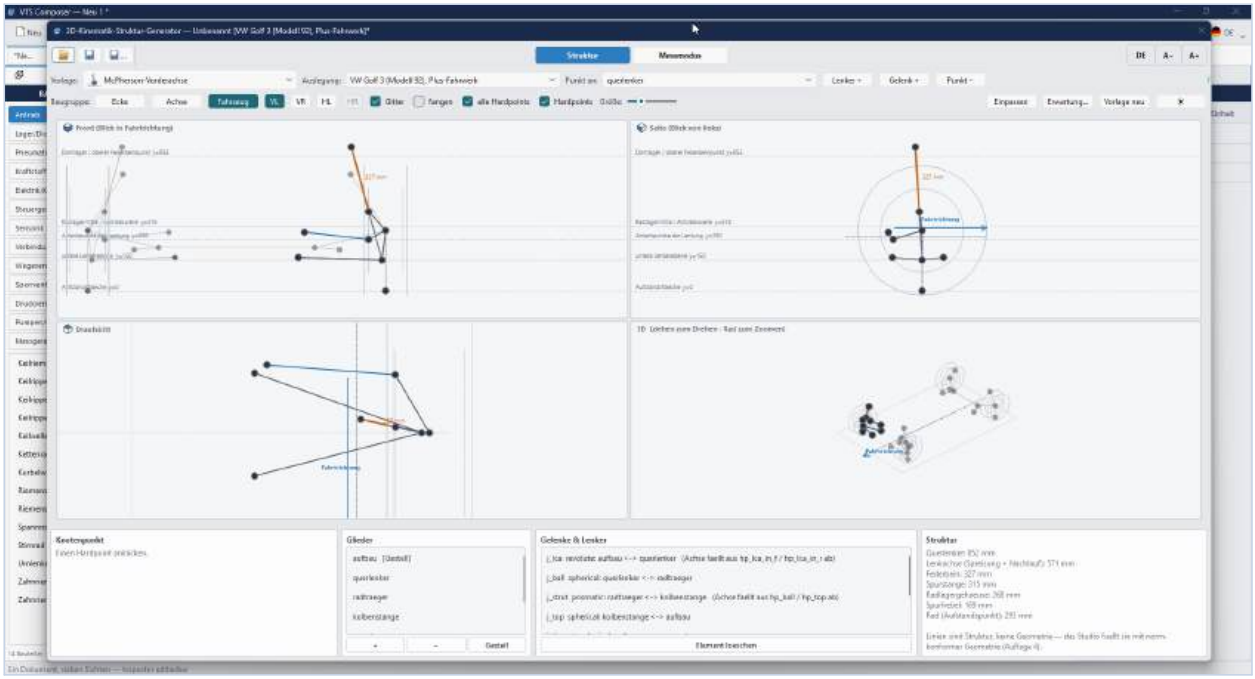


The result of the same input in the Studio: toothed pulleys to ISO 13050, the belt running as an outer tangent with real wrap around every pulley — no parallel spans, which is what a drawn representation would give you.

05

Kinematics

An animation in which a part performs a movement the real part cannot perform is not a teaching instrument — it is a wrong idea planted in the learner's head. VTS therefore treats kinematics as its own, checked data layer.



The 3D kinematics structure generator on a MacPherson front axle: front, side, top and spatial views all show the same hardpoints; below them the links, joints and two-point links as a list — and the computed mobility against the declared expectation. No geometry is involved yet.

Level 1 — what a single part may do

Every part carries a declared, technically admissible movement: type, axis, permitted degrees of freedom, driving variable, range and direction of action. The library holds 243 entries.

Joint type	Meaning	Example	Standard
revolute	one rotation about one axis	shaft, camshaft, gear, control arm	DIN 71751/71752
prismatic	one translation along one axis	pneumatic/hydraulic piston, steering rack	ISO 6020/15552
spherical	three rotations, no translation	ball joint, track rod end	DIN 71802, DIN ISO 12240
universal	two rotations across its own axis	universal joint in a propshaft	DIN 808
helical	rotation and translation coupled	trapezoidal lead screw	DIN 3975
fixed	no movement	housing, engine block	—

> The direction is declared as well. A valvetrain that closes the valve instead of opening it passes every length and interference check: a sign error is invisible in the design position, because there every direction gives zero travel.

Level 2 — what the assembly does

A permission holds nothing together. For assemblies, five obligations apply — born of a case that was proven in the mathematics and wrong nonetheless: a strut sat inside the wheel, the track rod was 52.9 mm too short.

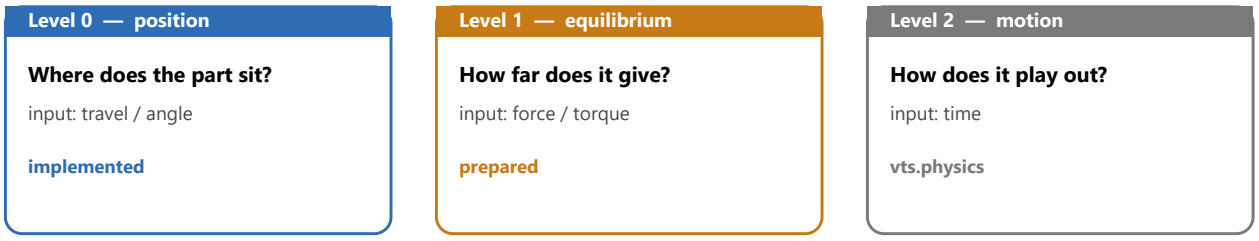
	Obligation	What it prevents
R1	Topology before geometry — links, joints, loops and driving variables are fixed before the first part	posing a closed loop as if it were a family tree
R2	Mobility proof — the degree of freedom is computed and checked against the expected value	mistaking „movable somehow" for „movable correctly"
R3	Invariants instead of poses, across the whole operating range: lengths, interference, ground contact, direction	checking only at the design value — where all errors are zero
R4	No second route — if a solved source exists, it is used	two truths that drift apart
R5	Visual inspection before sign-off — „mathematically verified" is not a sign-off	green tests waving through visible nonsense

Level 3 — the mechanism solver

A joint connects two bodies; from all the constraints comes a system of equations for the position of every link — purely kinematic, with no dynamics. Three statements carry it: the connection is primary, the degree of freedom is what remains (a bolted joint never comes undone), a two-point link is a distance constraint, not a part with a degree of freedom — and convergence is not a sign-off: on a wrongly declared assembly any solver returns convincing nonsense.

Name the model level

What the input is determines what is computed — a force slider on a pure position calculation is a design error. And whatever is not modelled gets said.



What the kinematics generator proves

The tool at the head of this chapter builds the topology before any geometry exists: four views, drag the hardpoints, wire up the joints, live mobility. Six axle layouts are available as templates — from MacPherson to push/pull rod; all axes and lengths fall out of the hardpoints. The expected value never updates itself — otherwise it would always be right and the check would be worthless. So the learner sees that a wheel is under-located before anything is drawn.

06

Standards compliance & quality

The biggest difference between VTS and a general 3D tool is not the graphics but a rule: what merely looks plausible is not admissible. Geometry has to follow the calculation prescribed by the relevant standard.

The rule and where it comes from

A sine wave with teeth looks like a gear from two metres away. It does not roll, it has no transmission behaviour, and anyone learning gearing from it learns something false. Binding therefore:

- The relevant standard is named in the source of the part — not in a handout beside it
- The geometry follows the calculation the standard prescribes: true involutes, true trapezoidal threads, true sprocket pocket profiles, true outer tangents on belts and chains, solid bodies with bores rather than hollow shells
- Every requirement from a standard has a regression test that checks it by an independent route
- A binding mapping table part family → standard records what applies

The convention aspect: 23 domains

Beyond the geometry of a part, 23 domains bind generation to real engineering rules — from the bend radius of a hose to the terminal designation in a vehicle.

Domain	Content (selection)	Standards
A / B / C	hoses and lines, pipes and flanges, cables — bend radii, straight run at the fitting, bolt patterns, strain relief, EMC separation	DIN 20066, ISO 4413/4414, DIN EN 1092-1, DIN VDE 0298-3, EN 60204-1
D	fundamentals: fits, geometrical tolerancing, cross-pattern tightening, pipe identification colours	ISO 286, ISO 1101, VDI 2230, DIN 2403
E	vehicle electrics: terminal designations, wire colours and cross-sections, high-voltage orange, functional safety	DIN 72552, DIN 72551, ISO 6469-3, ECE-R100, ISO 26262
F / G / H / I	joining technology, circuit diagrams, control and automation, machine safety	DIN EN ISO 2553, DIN EN 61082, ISO 13849
J / K	materials technology, technical drawing	DIN ISO 128, ISO 5456
L / M / N / O	physics and SI, thermodynamics, fluid mechanics, chemistry and process engineering	SI/CODATA
P / Q	electrical fundamentals, spaceflight	—
T / U / V / W	refrigeration, HVAC and building services, manufacturing and machining, drive-technology detail, metrology	—

Two levels — and a stricter one for machine output

Level	Effect	Applies to
Level 1	a warning — the assembly stays valid, the violation is named	hand-built assemblies (the default)

Level	Effect	Applies to
Level 2	outright rejection — the assembly is not built	machine-generated assemblies: generators and digital-twin import

The distinction has a simple reason: a warning nobody reads is not a check. For a person at a screen a warning is appropriate — they can deviate with good reason. For a machine producing a thousand parts it has no effect at all.

The test net

- 565 test files, all runnable headless — no screen, no clicking
- Independent oracles: belt length is checked against the perimeter of the convex hull, hatch area against line length times spacing, the analytic derivative against a numerical one. A test that walks the same path twice checks nothing
- Sweeps instead of set points: checks run across the whole operating range — in the design position every error is zero
- Counter-tests: every statement about direction is accompanied by proof that the wrong combination yields the opposite sign
- Visual inspection is mandatory — green tests are no substitute for looking

What this means in practice

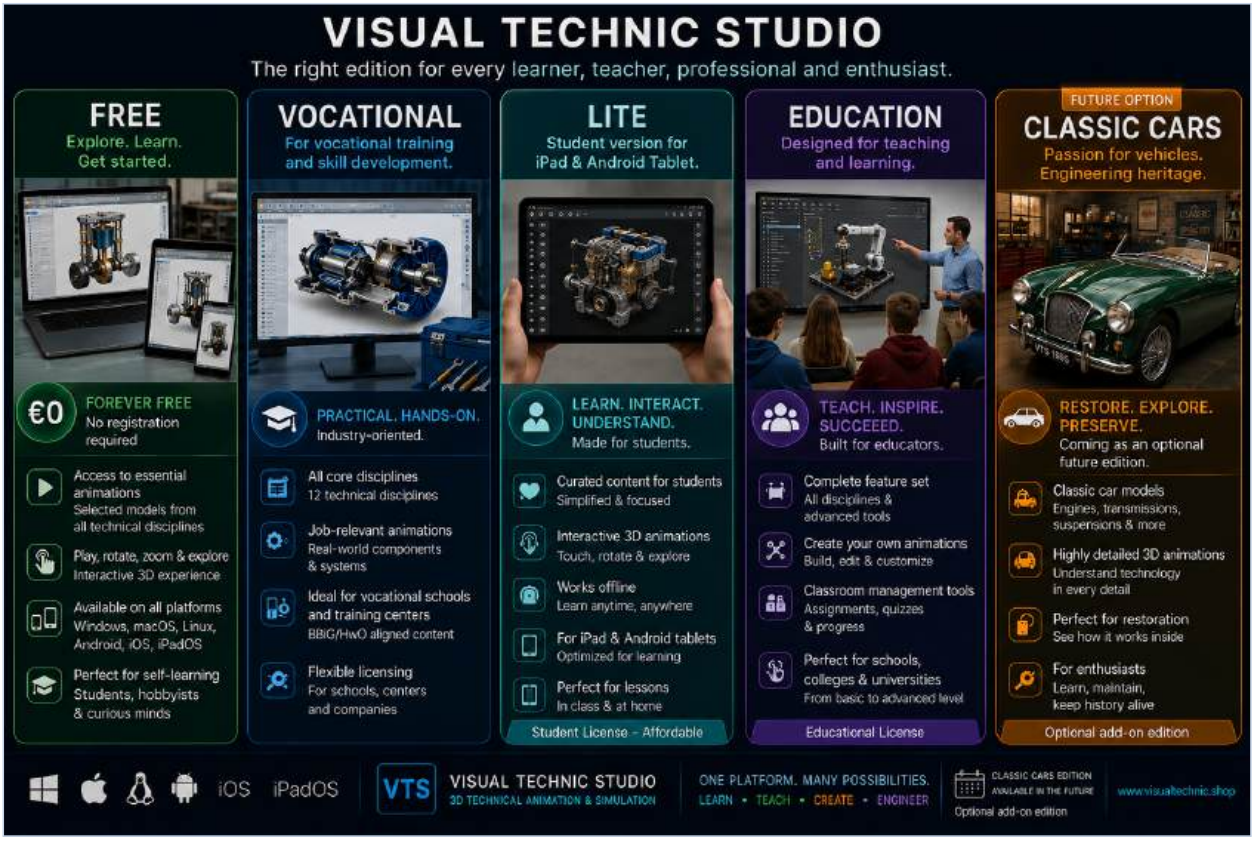
For whom	What follows from it
Teacher / trainer	what the animation shows can be checked against a standard — it can be named in the lesson, not merely admired. The standard is stated at the part, not in a footnote.
Exam preparation	designations, symbols and identification colours match what the examination expects — terminals, schematic symbols, wire colours.
Textbook publisher	the standards binding is a verifiable commitment, not a marketing claim. Where something is simplified it says so; where something is approximated it is counted and reported.
Company	the customer's own CAD geometry can be attached and is subjected to the same checks as the supplied parts.

> Whatever is not modelled gets named. A belt drive in VTS computes geometry and speeds — no preload, no slip, no service life. The tool says so. In teaching it is not the simplification that causes harm, but the unmentioned one: it creates confidence in a statement the model never made.

07

The editions

VTS comes in four editions with clearly separated audiences — plus a fifth planned as an add-on. At the centre of this folder are the two education editions: Education for schools and higher education, Vocational for occupational training.



The editions at a glance: FREE to get started, VOCATIONAL for occupational training, LITE as the student version for tablets, EDUCATION for teachers — and CLASSIC CARS as a planned add-on edition.

> Important: „Education" and „Vocational" are two different editions — Education teaches the why, Vocational trains the how. LITE is not a cut-down full version but the learner's side: it runs on iPad and Android tablets and is built for exploring, not for producing.

Education — for teaching, learning and inspiring

Audience: general education (lower and upper secondary), grammar schools, universities and universities of applied sciences — teachers and lecturers.

- Complete feature set: all disciplines and the advanced toolset
- Create, edit and adapt your own animations
- Classroom management: assignments, quizzes and progress
- Didactic tooltip and safety-notice system for lessons
- Export as a PDF report and MP4 video
- Licence: single teacher, class or site licence

VTS Education: didactic preparation, interactive 3D simulations and integrated exercises — shown here on robotics, mechanics, thermodynamics and spacetech. Product illustration; the interface shown is a designed view, not a screenshot of the current build.

Vocational — for occupational training

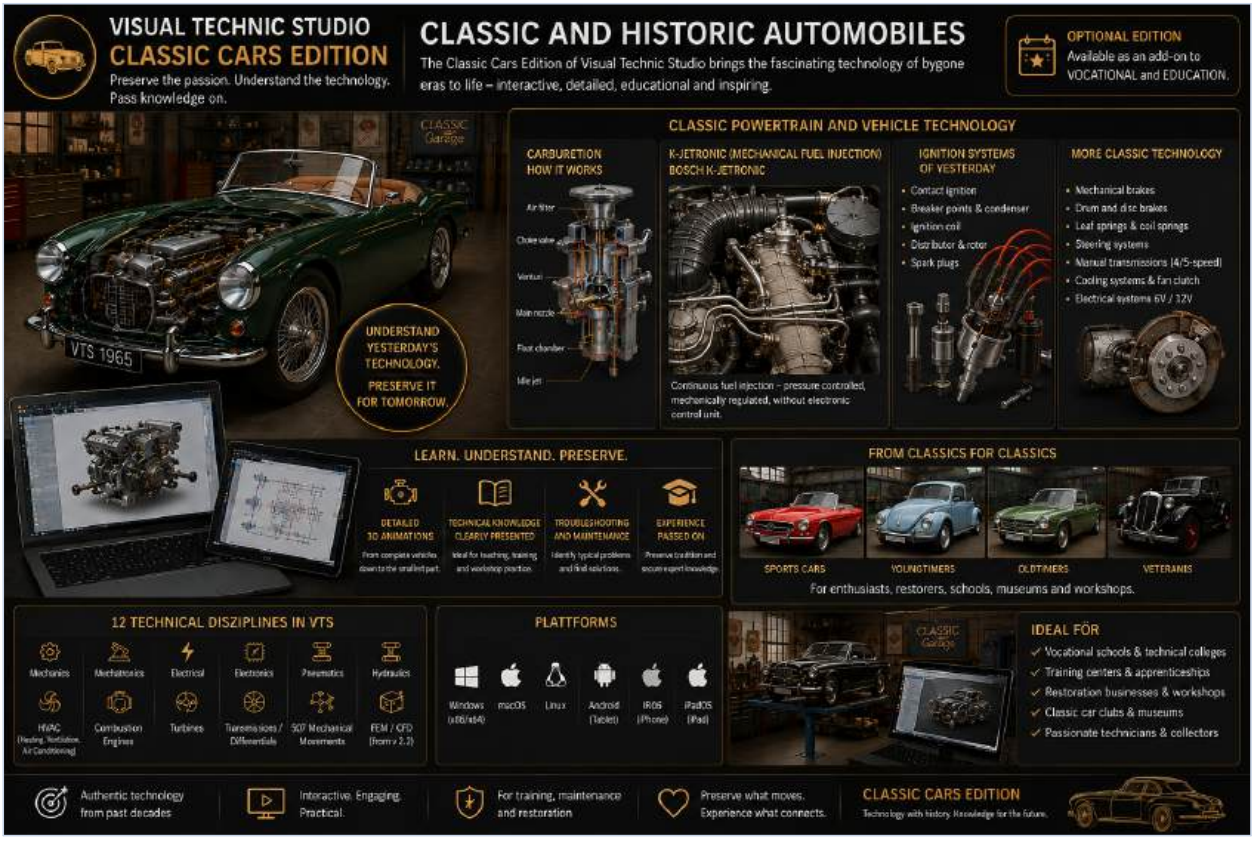
Audience: vocational education, both dual-system and school-based — vocational colleges, master-craftsman and technician schools, inter-company training centres and in-house training workshops.

- All twelve technical disciplines
- Job-relevant animations on real components and systems
- Composer generators for the brake system, drivetrain, bus systems, air conditioning, belt drive, clutch and valvetrain (chapter 04)
- Content aligned with the German vocational training regulations (BBiG/HwO) and relevant to examinations
- CAD import (STEP) for the company's own components
- Flexible licensing for schools, training centres and companies

Vocational covers vehicle technology in full breadth — automotive, commercial-vehicle and agricultural-machinery technology, each with its training occupations. Further occupational fields will follow.

Classic Cars — a planned add-on edition

An edition for classic and historic vehicles is planned as an add-on to Vocational and Education: carburettors, mechanical fuel injection (K-Jetronic), ignition with breaker points and distributor, drum brakes, rigid axles and leaf springs, 6-volt electrics. The audience is restoration businesses, classic car clubs, museums and technical colleges — and vocational schools that want to show how the same function was solved before electronics.



Classic Cars: the technology of past decades in the same presentation as the modern kind. A planned option, not yet available.

Edition overview

Edition	Audience	Scope	Platform
FREE	private, to get started	selected animations, play and explore	all
VOCATIONAL	vocational college, training centre, company	all disciplines, generators, CAD import	desktop
LITE	school students and apprentices	curated content, interactive, works offline	iPad, Android tablet
EDUCATION	school, higher education	complete feature set, own animations, classroom management	desktop
CLASSIC CARS	restoration, museums, clubs	add-on to Vocational and Education — planned	desktop

On sequence: development is concentrating first on completing Vocational and Education. A professional edition for industrial use is explicitly not being pursued at present — it would only get in the way of the two education editions.

08

Platforms & ports

VTS is built to be cross-platform (Qt 6 / C++17). Windows is the reference platform; the remaining desktop systems are part of the market-entry gate. For tablets the route is the Player, not the full application.

Platform	Product	Status	Technology
Windows (x64)	Studio + Composer + kinematics generator + LineWorks + Player	available	Qt 6 / C++17, OpenGL
macOS	Studio + Player	planned for market entry	Qt 6 / C++17
Linux	Studio + Player	planned for market entry	Qt 6 / C++17, AppImage
Android	Player	in progress	shared Qt RHI engine → Vulkan / OpenGL ES 3
iOS / iPadOS	Player	in progress	shared Qt RHI engine → Metal
iPadOS	VTS LITE (education version)	concept approved, implementation after the Studio gate	touch-first, Metal
Browser	QR preview	feasibility demonstrated	loads the real .vtsanim without installation

The target picture of platform coverage after the market-entry gate — the table above is what holds today.

One interface, five actively maintained languages (English, German, French, Spanish, Chinese), switchable at runtime; seven more are prepared. The AI assistant runs locally via Ollama — no cloud. The node-flow and block editors shown are development stages; today the text editor is the productive route.

> Why the Player and not the full application goes onto a tablet: the full application needs an embedded Python runtime, the script editor and the physics engine — several hundred megabytes together, and an interface that assumes a mouse. The Player reads a finished, baked package and stays under 100 MB. That is not a compromise but the appropriate split: the tablet is where you learn, the computer is where you build.

VTS LITE for the iPad

For mobile use in the classroom and the training workshop there is a dedicated iPad edition — in the two education variants, kept strictly separate and with different content at the same scope.

- Touch-optimised, adaptive tablet layouts
- Focused on playback, exploration and guided learning paths — not on full production
- Its own control panel: LITE computes the coupling and validity of the controls itself, unlike the Player, which only plays back
- Accessibility to WCAG 2.1 AA as a binding requirement — information is never carried by colour alone
- No backend of its own: teacher content arrives via device management, cloud, AirDrop, QR code or files — friendly to examination rules and data protection
- Well suited to bring-your-own-device settings in schools and colleges

Accessories — an outlook

Two extensions are being considered, both optional. An OBD2 adapter feeds readings from a real vehicle into the animation — so the measured value stands next to the simulated one. And an external control panel (ESP32-P4 with a 10.1-inch touchscreen) takes over the sliders, buttons, readings and teaching board: a pure input and display device that frees the iPad screen for the animation.

VTS LITE on the iPad, beside it the external control panel with sliders, readings and status display, and in front of it the OBD2 adapter. This leaves the iPad screen entirely to the animation — the operating controls move to the second device.

> Both accessories are an outlook, not a commitment. Their realisation is not yet assured, and the core function of VTS LITE explicitly does not depend on them — it runs on the tablet alone.

09

A concept for textbook publishers

For the publishing channel VTS follows a B2B production model — not a marketplace. Animations are embedded in textbooks like licensed picture content and opened through a QR code in the free VTS Player.

How an animation gets into a textbook

1. VTS produces the animation as a production package (set-up fee plus a per-book licence).
2. A QR code is printed in the textbook.
3. Teacher or learners scan the code with the free VTS Player.
4. The signed .vtsanim animation opens — offline capable, no sign-in required.

Business model

- Publishers treat animations like pictures in the book — they buy production packages
- Revenue model: a set-up fee per animation plus a licence/royalty per book
- Tenant separation: access across publisher boundaries is cryptographically impossible by construction
- Every .vtsanim is RSA-2048 signed; a watermark (publisher, book, figure) cannot be switched off

What the Composer changes for publishers

Until recently an animation was a script — changing it meant being able to program. With the Composer an editorial team describes the assembly itself, and the generators produce variants from engineering inputs rather than from program code.

- Variants instead of one-offs: one brake system becomes five circuit splits without any of them being redrawn
- An illustration per variant shows the editor immediately which version they are looking at — a name alone does not
- Your own drawings via LineWorks from the same geometry — legally clean and with demonstrable provenance
- Standards binding as a commitment: what the illustration shows follows the standard — verifiable, not asserted (chapter 06)

10

VTS Player

The VTS Player is a lightweight, freely distributable application for playing prepared 3D animations in the .vtsanim format — a product line of its own alongside the Studio, as easy to operate as a PDF viewer.

What the Player can do

- Interactive playback of prepared animations and simulations

- Opened by scanning a QR code straight from the textbook
- Section navigation (e.g. intake / compression / power / exhaust)
- Control panel: the controls authored in the Composer are baked into the package and available in the Player
- Kinematics in the package: the admissible degree of freedom of every part ships with it — the Player needs no Python for that
- Offline capable: no internet connection, no sign-in, no cloud access
- Compatible with school and examination IT: no telemetry, no background connections
- Download under 100 MB

What the Player deliberately is not

× No script editor, no Python API, no asset creation

× No write access to animations — a pure read-only renderer

× No live physics — animations are pre-computed

× No watermarks that can be switched off

> The restriction is the feature. A player that cannot write anything can be approved in an examination environment. A player with no network connection is unremarkable under data protection law. Neither is a cost-saving measure; both are what makes a school IT department willing to roll it out at all.

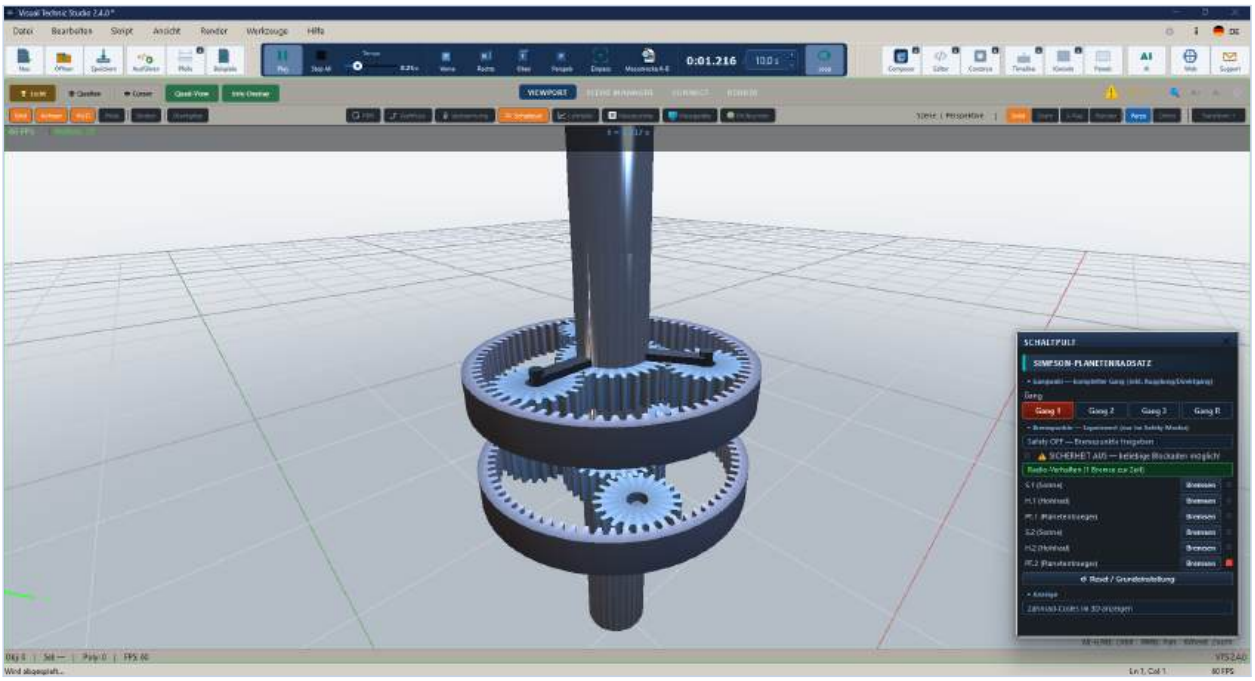
11

VTS in general education

In general education and at universities, VTS Education makes abstract technical and scientific relationships visible — from the gearbox to celestial mechanics.

Educational value

- Complex processes are shown in the 3D viewport clearly and repeatably — as often as needed, in slow motion, taken apart
- The didactic tooltip system explains parts and processes in context
- Learners can build animations themselves with VisualTechnicScript — STEM engagement on a genuine technical subject
- Didactic demonstrations across mechanics, thermodynamics, robotics and spaceflight
- Export as a PDF report and MP4 for worksheets and presentations



A Simpson planetary gear set: gear selection from the control panel. In the unlocked experiment mode each element can be braked individually — the relationship between the held member and the resulting ratio is tried out rather than memorised.

What the kinematics brings to the lesson

The mobility of an assembly is a topic that is hard to convey at the blackboard — it is a number that emerges from the interplay. In the kinematics generator you watch it live: remove a link and the number changes; attach it differently and the wheel is no longer located. The relationship between construction and mobility becomes tangible before anything is drawn.

Classroom scenarios

- Whole-class teaching at the interactive whiteboard with a guided animation
- Student workstations with their own edition or the Player
- University lectures and tutorials in the engineering foundation years
- Technical drawing on the 3D model — choose a view, place a section, dimension it (LineWorks)

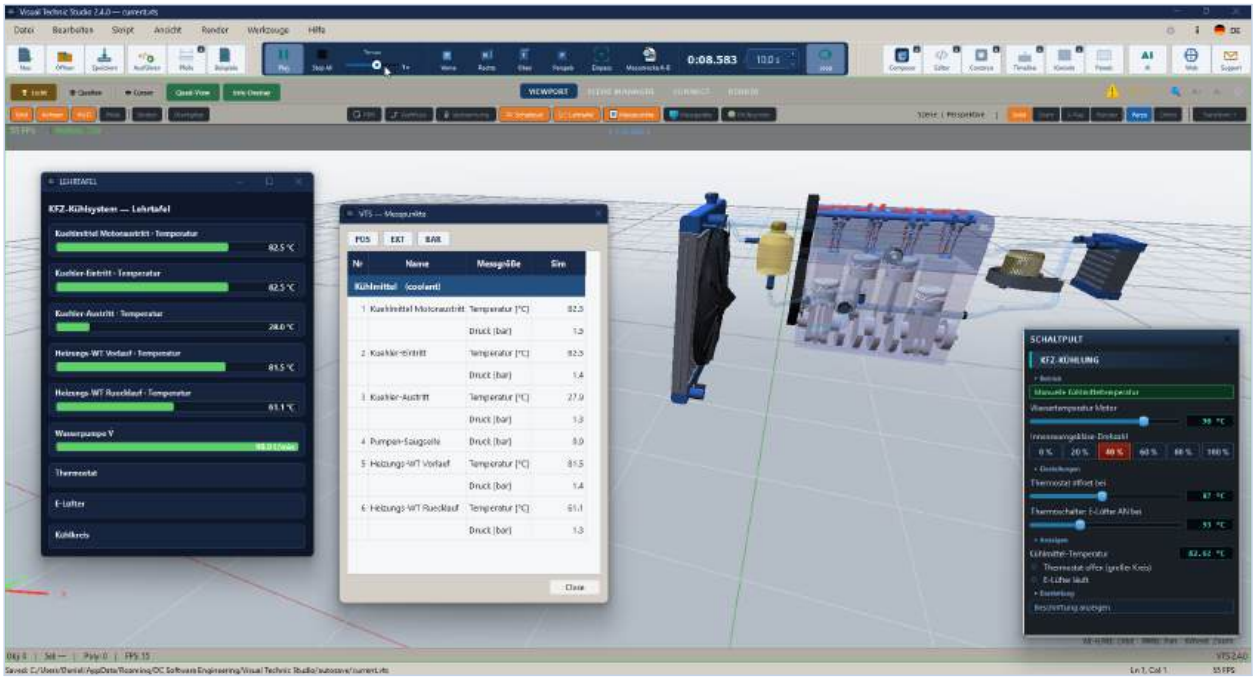
12

VTS in vocational training

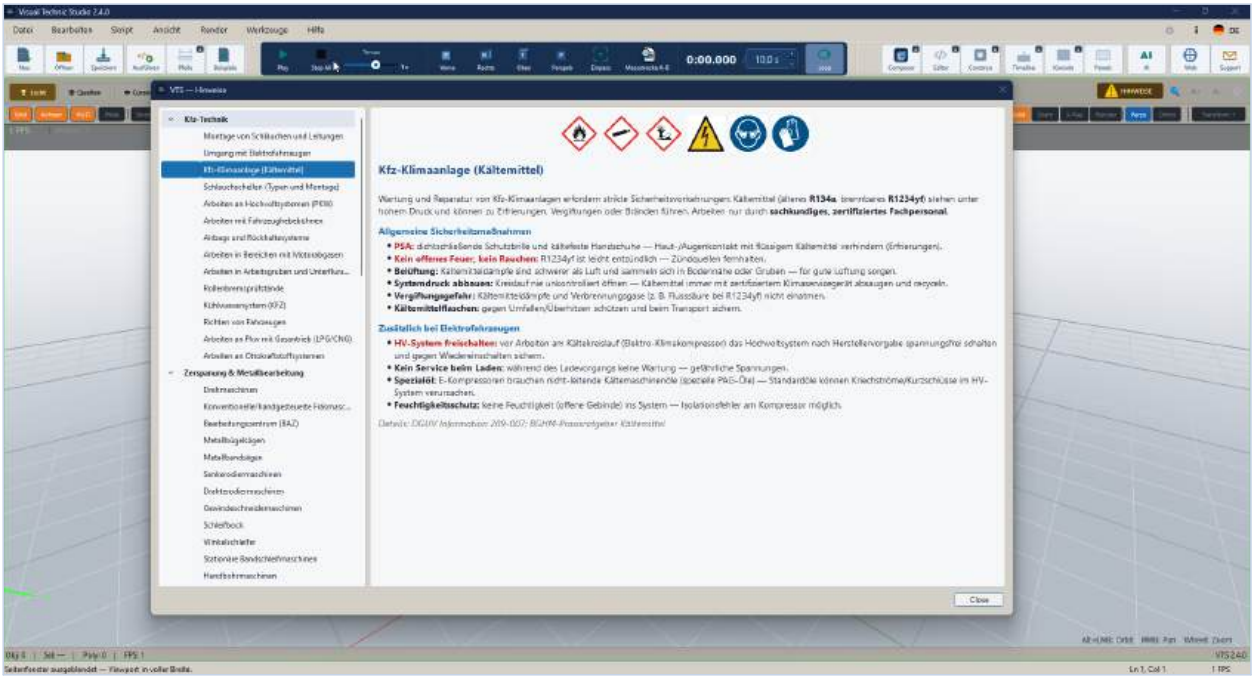
In vocational colleges, training companies and inter-company training centres, the Vocational edition combines the didactic base with job-related practice modules — above all for automotive, mechatronics and electrical trades.

Practical value

- Industry practice modules: combustion engine, gearbox, hydraulics, pneumatics, electrics, electronics, mechatronics, control engineering
- Generators instead of stills: a brake system is not shown but built — with the chosen circuit split, with or without ABS, hydraulic or air-operated
- Real components and processes become separable and traceable in slow motion
- Circuit and fluid diagrams as separate, standards-compliant views with test points — measuring is practised, not merely described
- CAD import (STEP) for the company's own components; preparation for chamber examinations through repeatable procedures
- Linked to textbook content via QR code and the VTS Player



An automotive cooling system: the teaching board shows the temperature at every measuring point, alongside the measuring point table and the control panel. The main circuit opens when the set threshold is reached — not on a stopwatch.



The safety-notice system supplies the relevant warnings for each work area with GHS and mandatory-action symbols — here vehicle air conditioning with R134a/R1234yf refrigerant, including the additions for high-voltage vehicles.

Example: a brake system in the lesson

The trainer selects a diagonal split, ABS resolved into individual valves, disc brakes front and rear. The generator builds the complete system: pedal, booster, master cylinder with reservoir, the individual modulator valves, return pump, low-pressure accumulator, four wheel brakes and the parking brake — connected, checked, drawn with standard symbols. Then the split is switched to front/rear, and the difference is not described but seen.

Where it is used

- Vocational college teaching in the subject classroom
- In-house training workshops and inter-company training
- Exam preparation and self-study phases (bring-your-own-device via the Player or an iPad)
- Instruction on plant-specific equipment via the company's own CAD import

13

Company & founder

Visual Technic Studio is developed by DC Software Engineering.

Item	Detail
Company	DC Software Engineering
Founder / developer	Daniel Zeimentz (solo founder)
Experience	Senior systems engineer with over 30 years of programming experience (trained as a state-certified IT assistant for production informatics, 1989–1991) Automotive electrician; master craftsman in automotive technology (1999, 2004) BOSCH systems technician (diagnostics specialist, 2004) Teaching qualification with a pedagogical diploma (Ministry of Education, Luxembourg, 2014) as a state-certified instructor for automotive technology, with more than 12 years in the profession
Product	Visual Technic Studio (VTS) — VTS Studio 2.4.x (development build)
Primary market	German-speaking Europe and Luxembourg, vocational and general education
Secondary markets	European Union, the Americas, Asia
Web	visualtechnicstudio.com | dc-software-engineering.com

> Run solo and technically self-sufficient: architecture, implementation, documentation and business development from one hand. The dual qualification — master craftsman in automotive technology and software developer and certified instructor — is the reason the standards binding described in chapter 06 is not an afterthought in this product but the building specification.

Contact & next steps

Interested schools, vocational colleges, publishers and education partners are invited to get in touch about pilot projects and pre-orders.

Enquiry	Suitable offer
School / university	Education — class or site licence, pilot phase with support
Vocational college / training company	Vocational — including the Composer and generators for your own equipment
School students and apprentices	LITE on the tablet, plus the free Player on every platform
Textbook publisher	production package .vtsanim + QR workflow, drawing derivation via LineWorks
Restoration / museum / club	Classic Cars — planned add-on edition, registrations of interest welcome