01 / REFERENCES AND GENERATION

From generation to a 3D design studio.

Ferrari livery on the production car, captured directly in the studio.

Start with references

F1 measurement references and car photographs guided the shape and proportions of the base car. The build combined 3D generation, mesh refinement and vector graphics to create a customizable racing studio.

Generate with Astra and Scenario

Base-model generation used Meshy Multi Image and Rodin 2.5. The editor runs the Scenario Rodin Gen-2.5 High reconstruction, prepared in Blender.

Build for customization

Blender organized the geometry into editable regions. Scenario vectorization prepared graphics for the interface and liveries. Three.js brought the model, paint and stickers together in an interactive browser editor.

02 / PREPARE THE GEOMETRY IN BLENDER

Turn the generated car into editable parts.

Refine the reconstruction

Astra refined the car in Blender: reshaped mirror pods and supports, recessed reflective glass, smooth halo roots, a rounded airbox and refined rear and sidepod contours.

Organize the editable surfaces

Orientation and scale were normalized while preserving mesh normals. The car was partitioned into nine groups: wheels, mechanical parts, sidepods, engine cover, rear wing, body, nose, front wing and halo.

Prepare clean paint surfaces

Generated material maps contained embossed logos. Removing those maps created clean surfaces for paint and stickers. The surface groups let the editor address each region independently.

Prepared geometry with editable paint surfaces and separately positioned graphics.

03 / CREATE USABLE GRAPHICS WITH SCENARIO

From raster marks to clean vectors.

Team selector with Scenario-vectorized Alpine and Haas marks.

Vectorize with Scenario

Astra used Scenario to vectorize UI components and sticker packs. Scenario Vectorize converted the Alpine and Haas team tiles and a Hard Rock sponsor mark into SVG paths.

Prepare the input, then clean the result

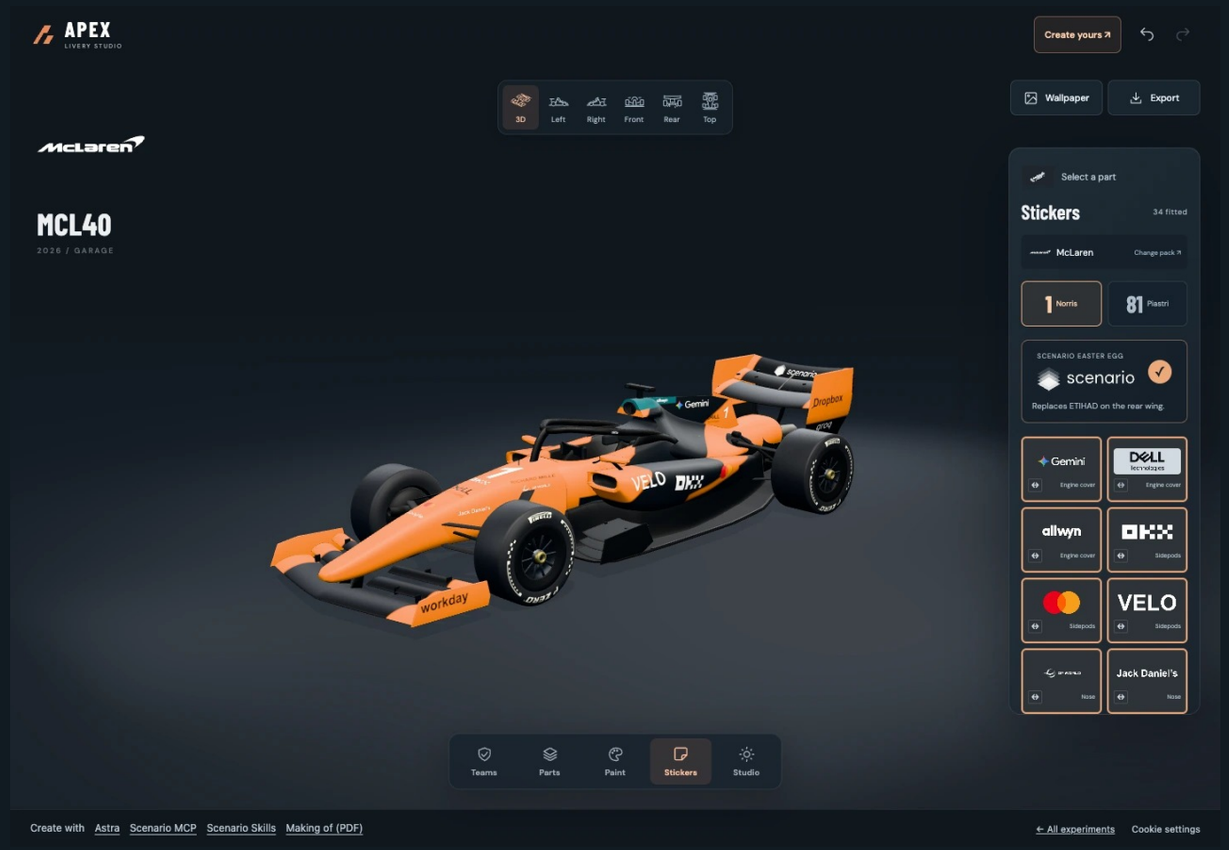
Alpine's white lettering used a temporary dark matte for tracing; the matte was removed from the SVG. The Haas wordmark was cropped and enlarged before tracing, then displayed as a complete vector mark.

Inspect and iterate

The first Hard Rock trace lacked contrast. A black-on-white input produced a usable trace; light background paths were removed and the surviving mark recolored white. The result was checked against the brand reference.

04 / ASSEMBLE THE BROWSER EDITOR

Make one model support eleven liveries.



Surface-based stickers in the finished editor, shown with the later Scenario sponsor option.

Load the prepared car into Three.js

Three.js loads the GLB, identifies paintable parts and applies editable materials. Canvas textures and paint shaders supply the livery, allowing the same geometry to support different team identities.

Keep the design in data

Team definitions supply palettes, car names, drivers and sticker packs. Surface attachments place the graphics on named parts. The reference workbench aligns comparison views using wheel centres and a nose marker.

Connect the controls to the scene

Teams, Parts, Paint and Stickers update a shared design state. Part inspection, saved designs, undo and redo connect the prepared geometry and graphics to a reversible editing workflow.

05 / EXPORT THE DESIGN BEYOND THE BROWSER

Bake the livery into a portable model.

01 PREPARE IN BLENDER

Build a dedicated, non-overlapping UV layout from the source car.

02 BAKE IN THE EDITOR

Render the configured paint and tyre shaders into material textures.

03 PACKAGE AS GLB

Embed the textures with the assembled car for an external 3D viewer.

Convert shaders into material textures

The live editor uses procedural shaders. The exporter bakes their base color and roughness/metalness into standard embedded textures so the configured design can be opened in Blender or another 3D viewer.

Keep geometry and UVs together

Blender generated a dedicated UV layout with an atlas lookup tied to the car's vertex data. The exporter reuses this layout while preserving the configured geometry and paint coordinates.

Package the configured car

The GLB contains the assembled car and its graphics. Studio lights, floor and inspection highlights stay in the editor. Mirror glass exports as polished metal, using the destination scene for reflections.

From Blender UV preparation to a self-contained, textured GLB.

06 / TOOLS AND RESOURCES

The tools behind the studio.

Astra / Build the experience

Astra combined the generated assets, Blender refinements and editor code into the working studio. Team data, editable materials and surface attachments connect the car to its customization tools.

Scenario / Generate and vectorize

Scenario supplied 3D generation and graphic vectorization. Meshy Multi Image and Rodin 2.5 supported the base-model work; Scenario Vectorize turned raster team and sponsor marks into SVG assets.

Blender / Prepare and export

Blender refined the geometry, organized editable surface regions and generated the export UV layout. The browser exporter then bakes the configured livery into textures and packages the car as a GLB.

START HERE

Scenario

Scenario MCP

Original studio

Build discussion (team access)

Workflow references

Explore Scenario and its agent resources with the links above. Team and sponsor marks belong to their respective owners.