

Platform Software · Optical Tracking

SKADI Desktop App & API

One SKADI desktop app leads through the API to medical integration and an industrial extension.

Period 2023.02 - present | Evidence state Ongoing | Portfolio tier Platform Software

The desktop app handles marker definition, file work, and real-time 6DoF status, while the DtSkadi.dll API connects to surgical-navigation and robot-tracking applications.

Product naming note

In public material, Viewer is the former name of this product, and MarkerEditor is the marker-definition and file-work surface within the same desktop app.

Product problem

Connecting an optical tracker to applications requires predictable product interfaces for marker definition, device state, API-open failures, and delivery of tracked transforms.

SKADI desktop app

One product flow covers marker definition and file work, tracker connection, and 6DoF status inspection.

SKADI API

An application starts the connection through `OpenEx()` in `DtSkadi.dll`, then branches to either a ready state or a handled error.

Medical integration

The same desktop app and API connect to a de-identified tray and 3D Slicer CT workspace to check the surgical-navigation integration boundary.

Industrial extension

The same 6DoF tracking layer was applied to the coordinate relationship between robot-end and target markers to support precision-docking integration.

SKADI desktop app

One product flow covers marker definition and file work, tracker connection, and 6DoF status inspection.

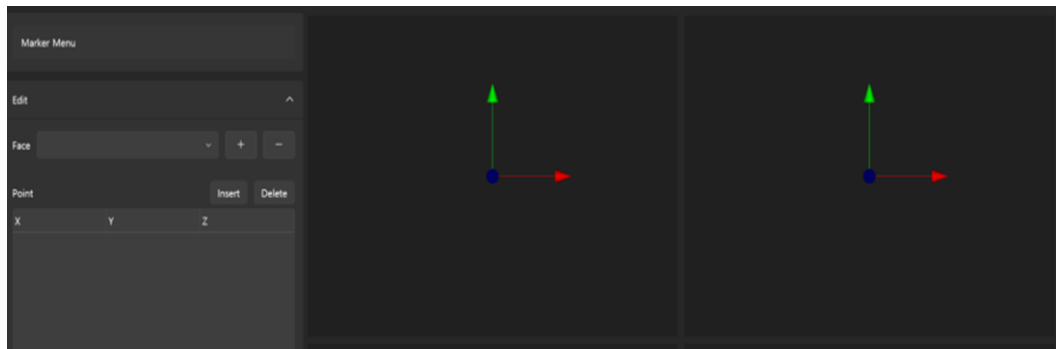


Figure 1. Marker definition and file work in the desktop app

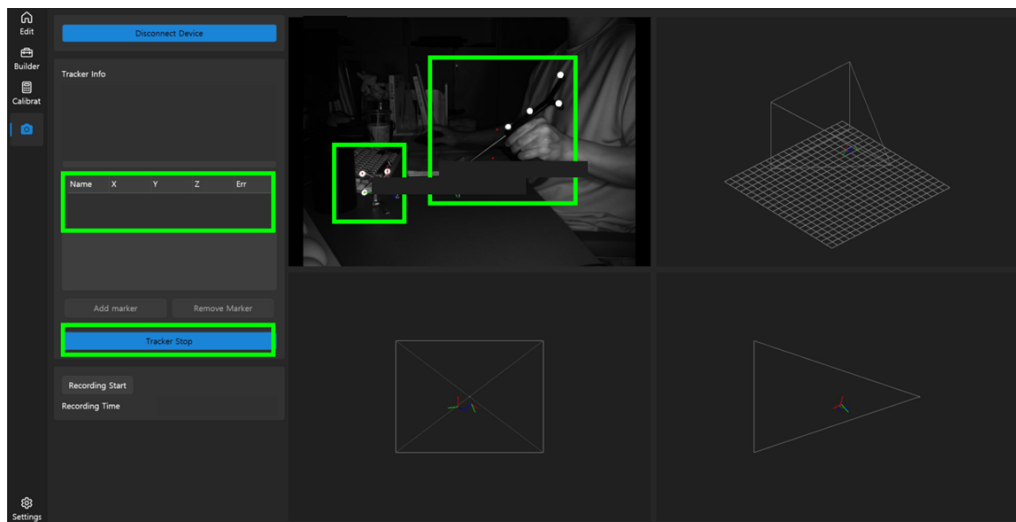


Figure 2. Tracker status and real-time 6DoF view

My role

I implemented MarkerEditor marker-coordinate input validation and own desktop-app maintenance, releases, and user documentation.

Team and partner boundary

The optical tracker and marker or instrument hardware are team results.

Five approved real screens cover marker definition and file work, tracker status and 6DoF, a de-identified medical tray, the CT workspace, and the robot-tracking field setup.

SKADI API

An application starts the connection through `OpenEx()` in `DtSkadi.dll`, then branches to either a ready state or a handled error.

SKADI API connection flow



My role

I structured the `DtSkadi.dll` integration and `OpenEx()` error branch so failure stops before an invalid state, and I own public API documentation and integration support.

Team and partner boundary

Final medical or industrial applications and product performance belong to the team and partners.

OpenEx() stability

In the `DtSkadi.dll` integration, `OpenEx()` success and error results are separated so failure stops before an invalid state and crash.

The single entry point makes the success and error outcomes explicit.

[Public API change note](#)

Medical integration

The same desktop app and API connect to a de-identified tray and 3D Slicer CT workspace to check the surgical-navigation integration boundary.

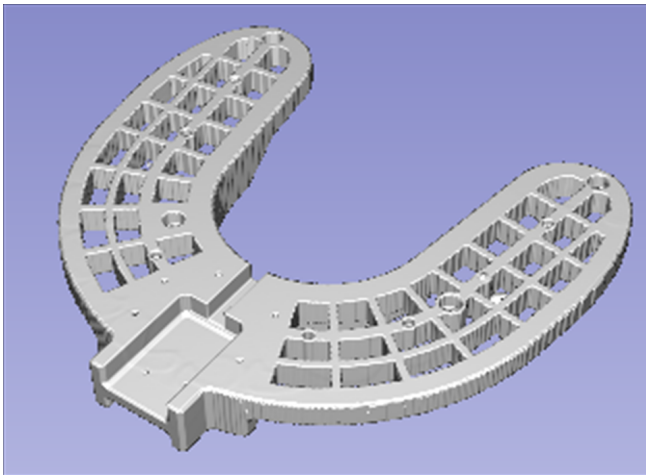


Figure 3. De-identified tray used in the medical integration

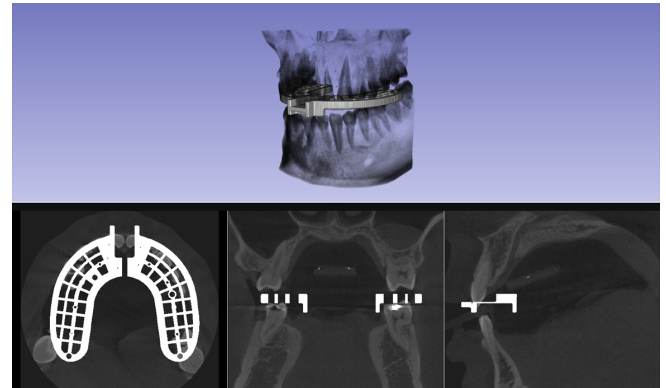


Figure 4. De-identified CT workspace in the medical integration

My role

I own the tracking API and desktop-app connection, public documentation, and Slicer integration support.

Team and partner boundary

The clinical workflow and final surgical-navigation application are team and partner results; no clinical effect is claimed as my result.

Industrial extension

The same 6DoF tracking layer was applied to the coordinate relationship between robot-end and target markers to support precision-docking integration.

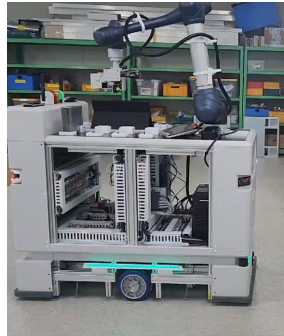


Figure 5. Robot-tracking field setup for the industrial extension

My role

I own tracking-API and coordinate-integration support.

Team and partner boundary

Robot control and the final docking behavior are team and customer application results; product performance or customer outcomes are not claimed as mine.

Public documentation and product information

DOCUMENTATION

SKADI Viewer documentation

Explains device connection, marker editing, real-time tracking, and the Viewer workflow.

PRODUCT

SKADI Viewer product page

Public overview and application scope for the Viewer product family.

DOCUMENTATION

SKADI API documentation

Explains the public API entry point and integration flow used by tracking applications.

PRODUCT

SKADI API product page

Public product overview and application boundary for the SKADI API.

Limits and team result

The public material covers only product interfaces and integration boundaries. Faces, device IDs, coordinate values, logos, versions, licence paths, and performance assessments are excluded.

Device hardware, optical and mechanical design, final medical and industrial applications, robot control, sales, and customer outcomes belong to the team and partners. Product performance, clinical effects, and sales outcomes are not claimed as my results.