

# TradingSystemAPI — Qt desktop for low-latency market operations

**Domain:** Trading / fintech **Duration:** 2016 – 2018 **Stack:** C++ · Qt · Wt

Native trading platform UI with Qt and Wt — multi-window desks, fast market-data refresh, and tight coupling to proprietary trading libraries for operators who cannot tolerate sluggish or fragile interfaces.

## CHALLENGE

### Performance-critical desktop for trading desks

Build trader-facing tooling with dense operational layouts, low-latency interaction, and integration with native strategy and market-data codepaths — suitable for regulated or high-stakes desks.

## APPROACH

### Qt-native UI with disciplined C++ hot paths

- Qt widgets and views for dense multi-window trading layouts
- C++ integration with proprietary libraries — no thick runtime abstractions
- Wt used where web-facing modules complemented desktop surfaces
- Release discipline aligned to production trading feedback loops

## TECHNOLOGY

### Stack & delivery surface

C++ Qt Wt Market data UI Trading operations

## OUTCOMES

### What the programme proved

- Reference case on [sevendyne.com/projects/qt/](https://sevendyne.com/projects/qt/) — TradingSystemAPI UI
- Demonstrates Qt depth for finance/trading buyers, not only industrial
- Founder-led delivery during Sevendyne's formative 2016–2018 period