

# Xact-React System Report

---

## 1. Project Overview

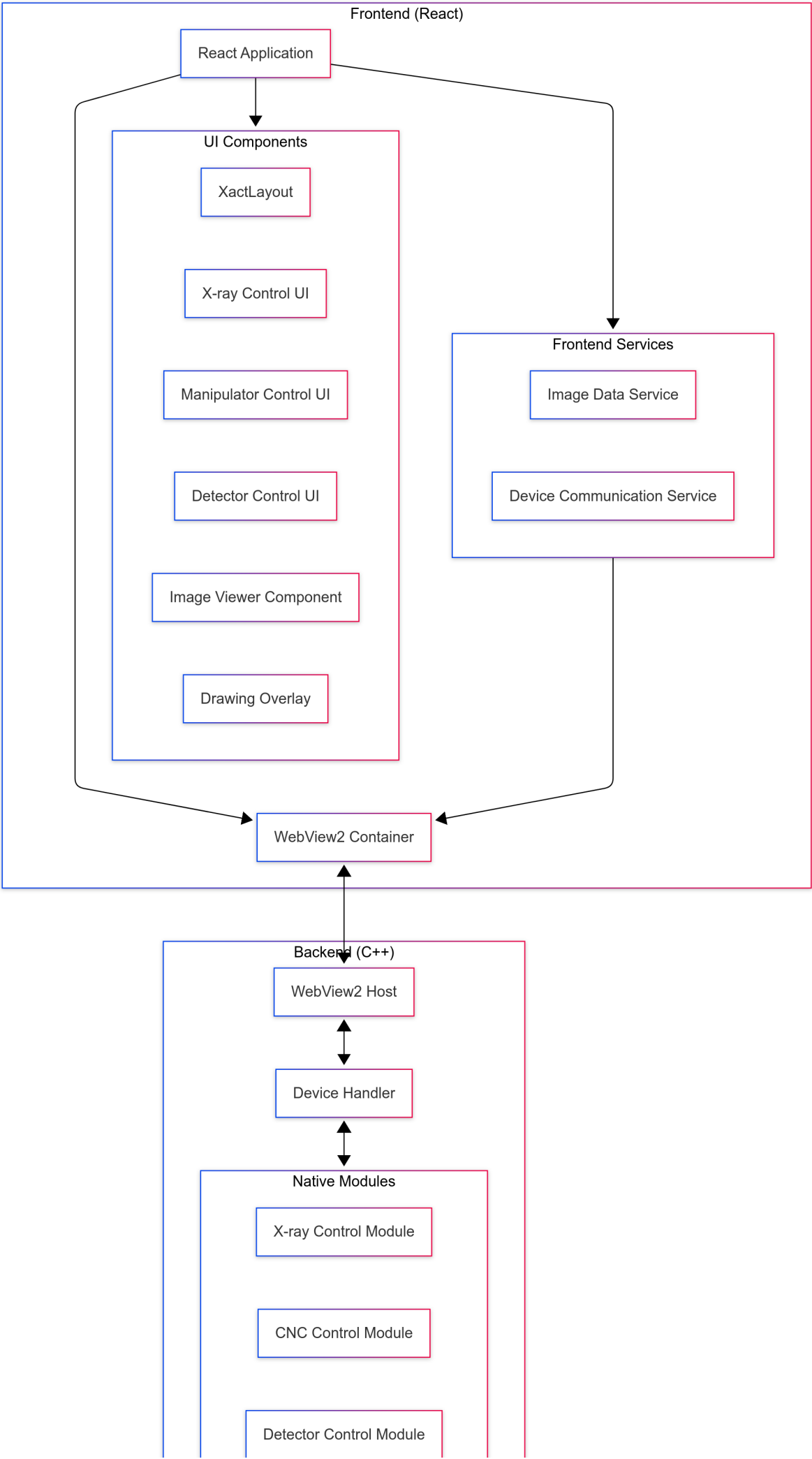
---

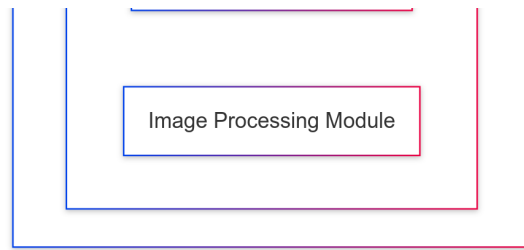
This project is a reconstruction and upgrade of the Xact system, using the **React frontend** framework to replace the original MFC/Qt interface while retaining the original **C++ backend** functional modules. The project aims to provide a **modern user interface and interaction experience** while maintaining full compatibility with the original X-ray control, CNC control, detector control, and image processing functions.

## 2. System Architecture

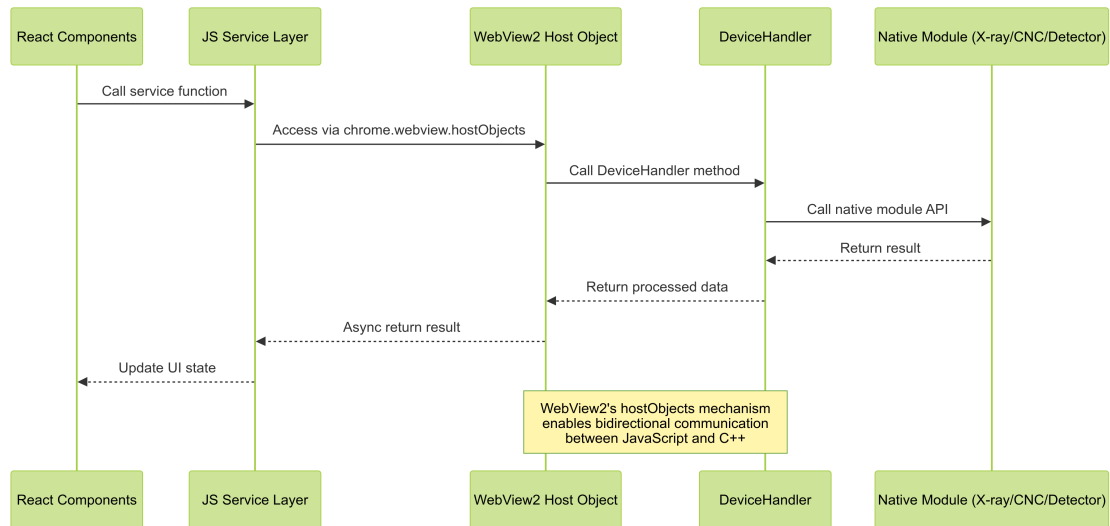
---

### 2.1 Overall Architecture Diagram

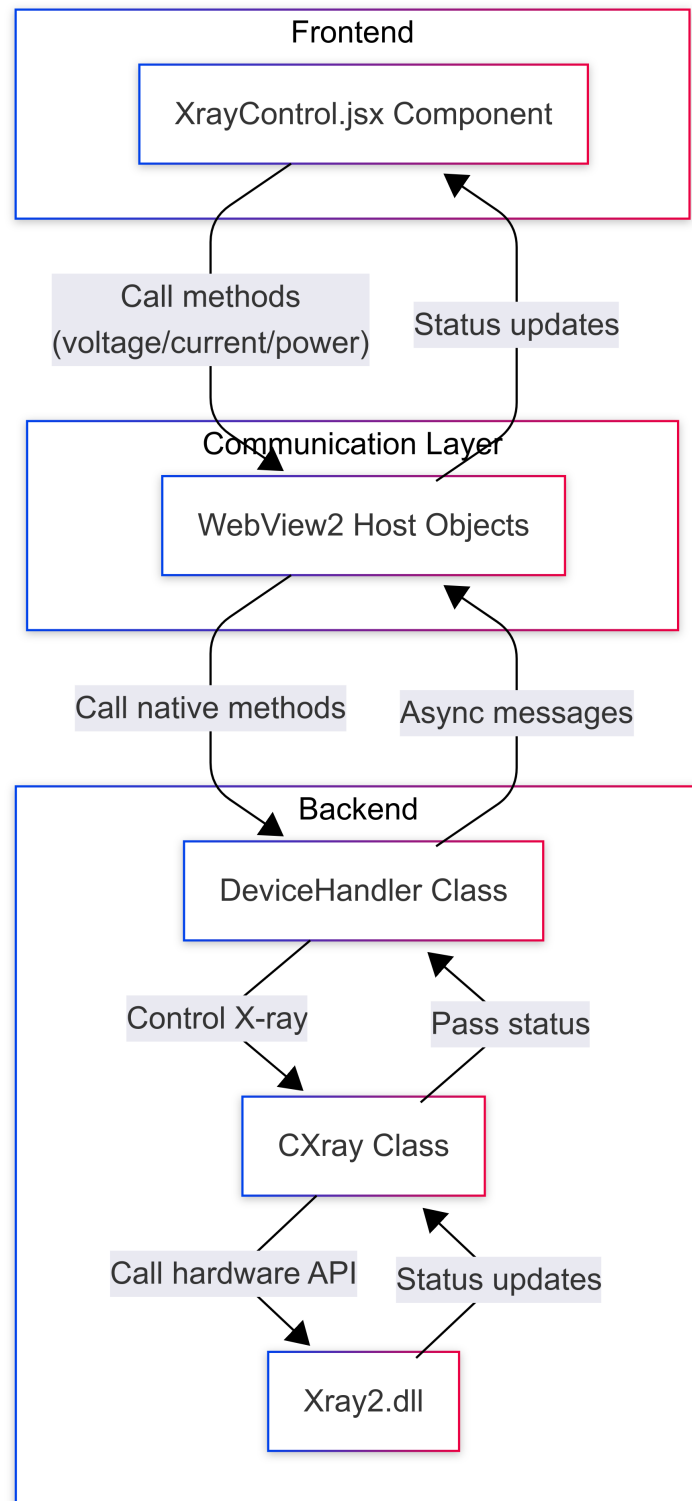




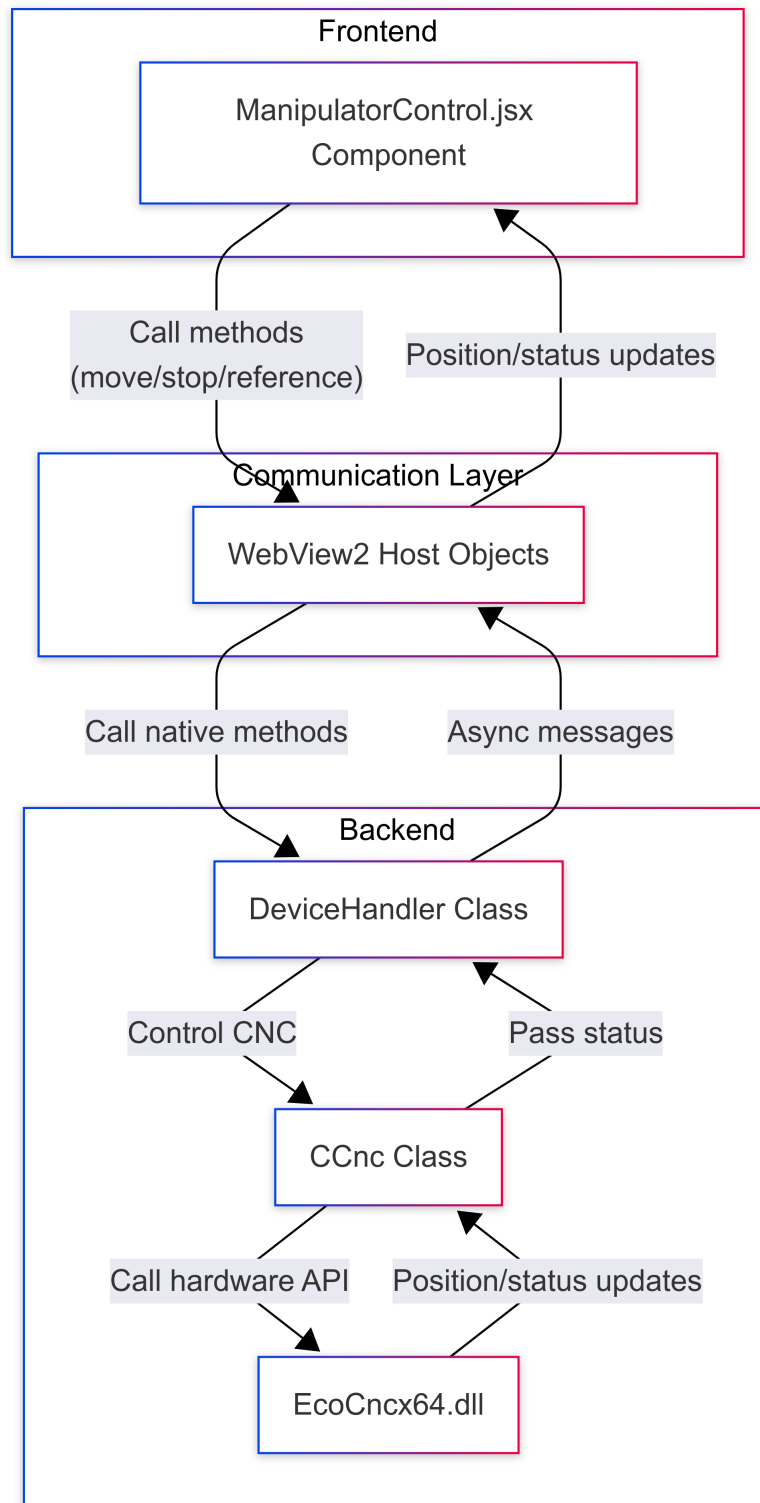
## 2.2 Frontend-Backend Communication Architecture



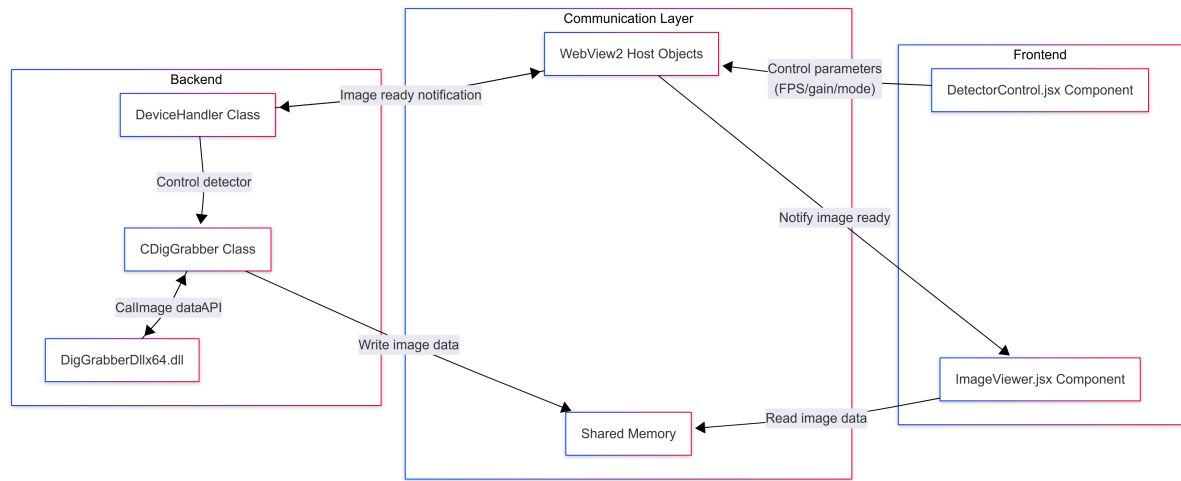
## 2.3 X-ray Control Architecture



## 2.4 CNC Control Architecture

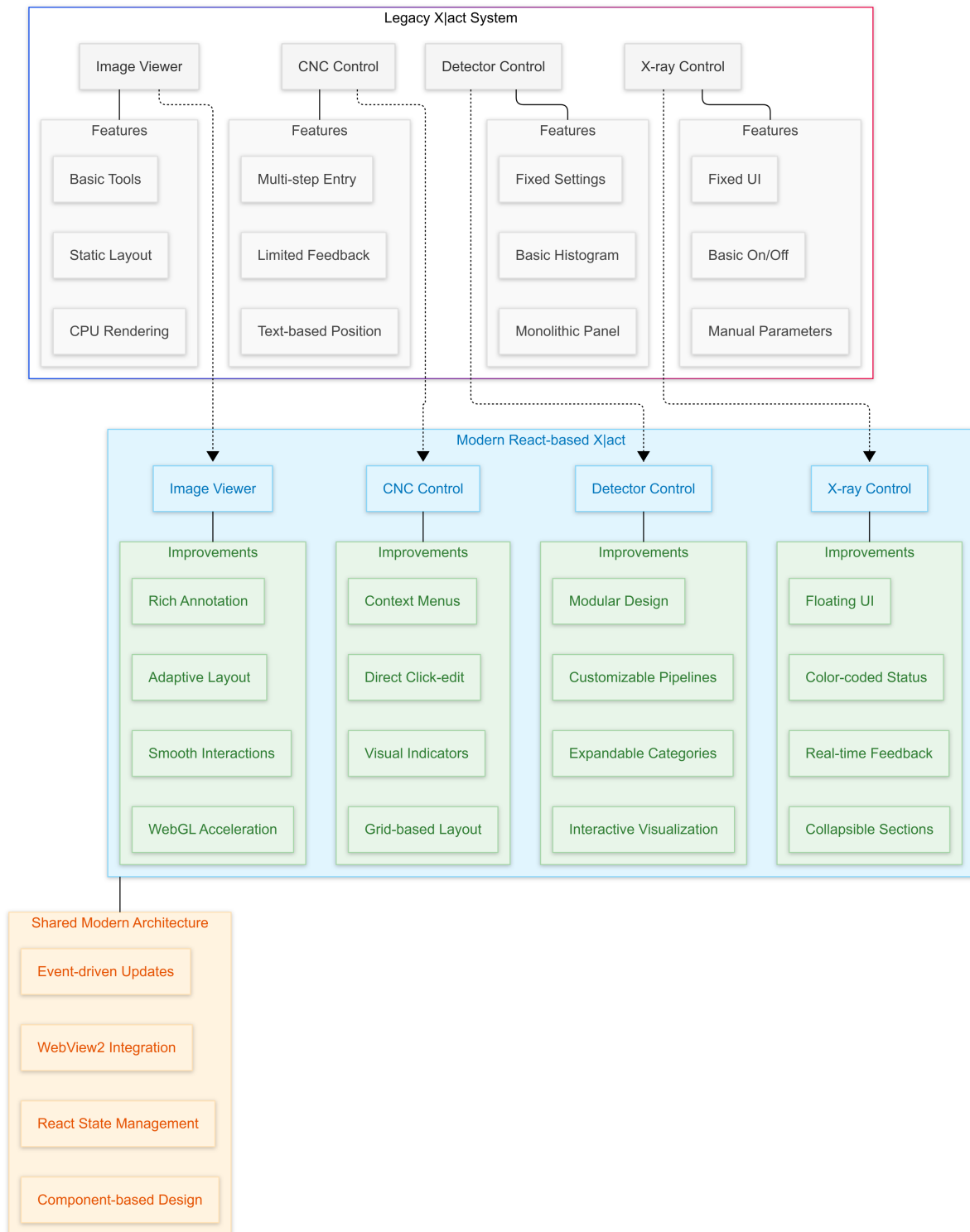


## 2.5 Detector Control Architecture



### 3. Core Functional Modules

---



## 4. Comparison Between Old and New Systems

### 4.1 User Interface Comparison

Feature	Old System	New System	Advantage
Overall Layout	Fixed layout, non-adjustable functional areas	Responsive layout, supporting collapse/expand and floating windows	Improved space utilization, enhanced user experience
Control Panel	Fixed position, non-movable	Draggable floating window, position adjustable	Higher flexibility, adapts to different workflows
Visual Style	Traditional Windows style, gray tone	Modern design, clear color coding and icons	More intuitive and aesthetic, lower learning curve
Functional Area Organization	Deep hierarchy, requires multiple clicks	Flat design, quick access with collapsible panels	Improved operation efficiency, reduced click count
Image Processing	Fixed toolbar, occupies interface space	Floating toolbar, displayed as needed	Focus on image observation, reduced distraction

## 4.2 Technical Architecture Comparison

Feature	Old System	New System	Advantage
Frontend Framework	MFC/Qt (C++)	React (JavaScript)	Higher development efficiency, component reuse, rich ecosystem
Rendering Engine	GDI/DirectX	WebGL (PixiJS)	High-performance image rendering, GPU acceleration
UI Components	Custom controls	React components + Tailwind CSS	Rapid development, consistent visual style
Frontend-Backend Communication	Tightly coupled C++	WebView2 bridging	Decoupled frontend and backend, independent development and testing
Code Organization	Weak modularity	Fully componentized	Better maintainability and scalability
State Management	Distributed management	Centralized management with React Hooks	Predictable state changes, easier debugging

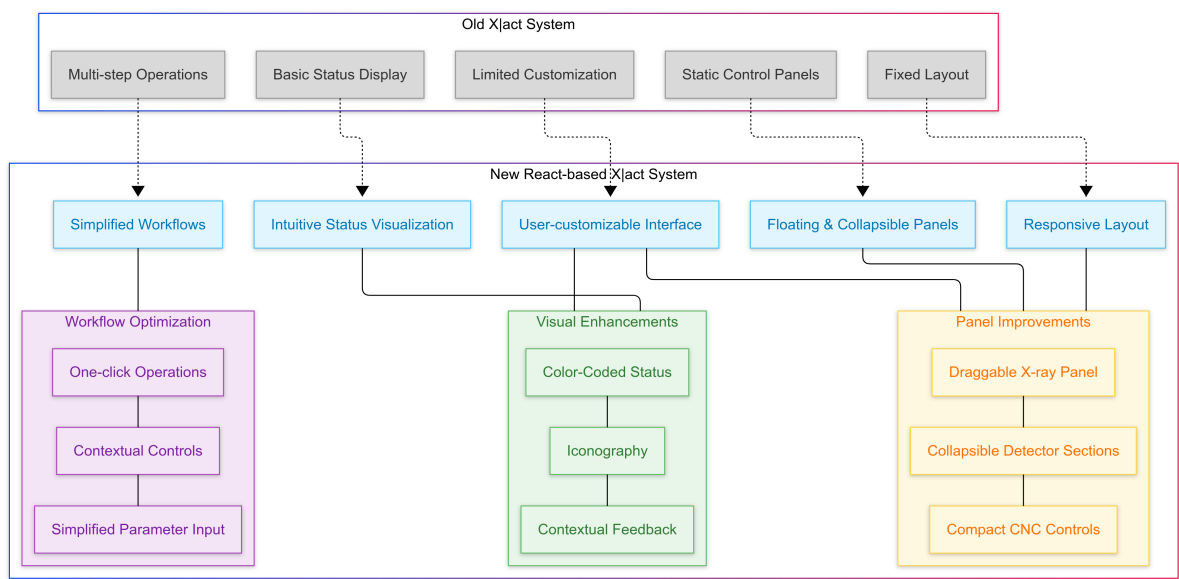
## 4.3 Performance Comparison



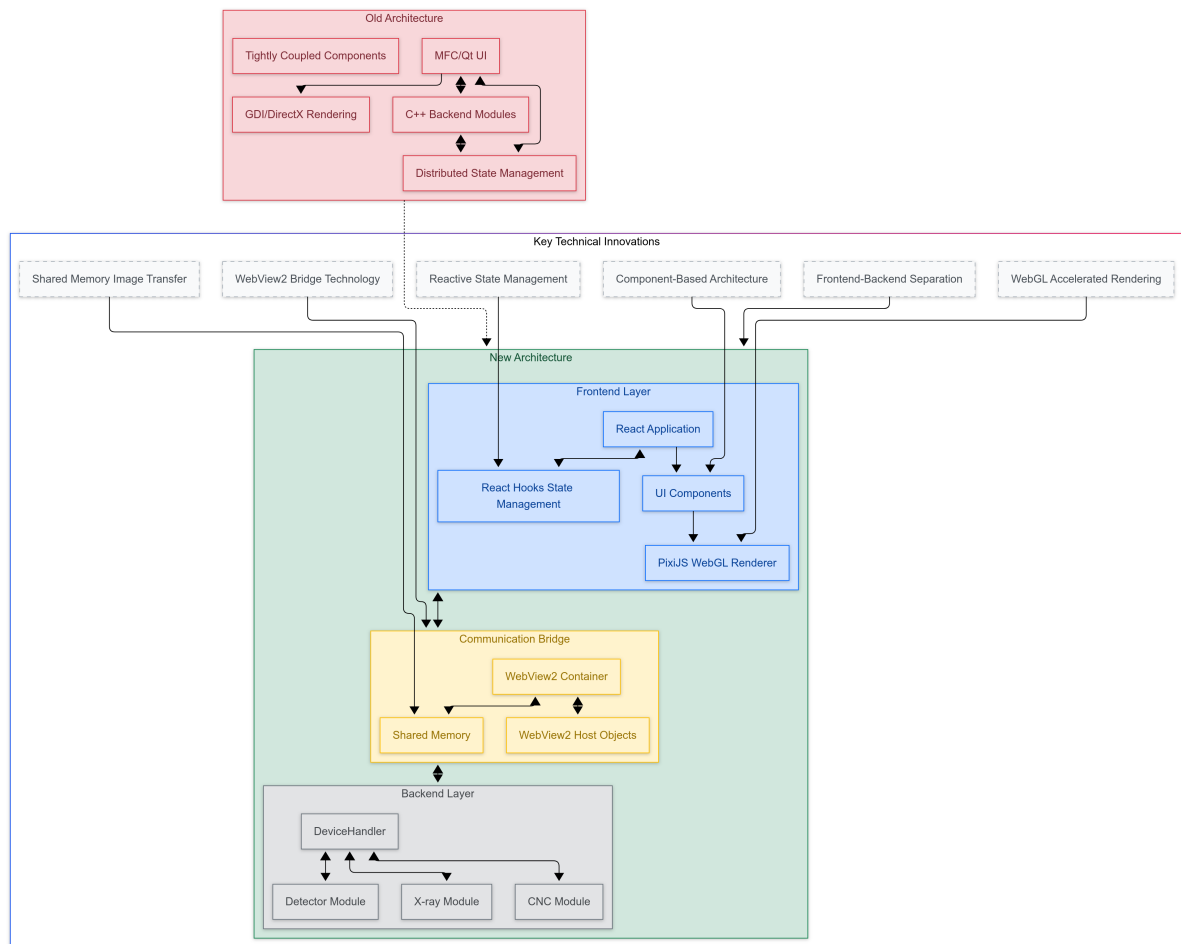
Feature	Old System	New System	Advantage
Startup Time	Faster	Slightly slower but acceptable	Higher feature integration
Image Rendering	Good	Excellent (WebGL acceleration)	Smoother large image processing
Memory Usage	Lower	Slightly higher but stable	Richer features, better experience
Response Speed	Good	Good	Maintained system responsiveness
Image Processing	CPU intensive	GPU accelerated	Faster large image processing

## 5. Improvements and Innovations

### 5.1 UI/UX Improvements



### 5.2 Technical Innovations



## 6. Conclusion

This project has successfully achieved the modernization upgrade of the Xact system. By adopting the React frontend framework and WebView2 technology while retaining the original C++ backend functionality, it provides a more intuitive, aesthetically pleasing, and efficient user interface. The main achievements include:

1. A completely new user interface, providing better interaction experience
2. Maintaining full compatibility with original functionality
3. Frontend-backend separation architecture, improving system maintainability