

05 Technical Stack

2025-10-19

Contents

1	Technical Stack - Vision-Based Pick and Place System	2
1.1	Overview	2
1.2	1. Architecture Layers	2
1.3	2. Layer 1: Hardware	3
1.3.1	2.1 Robot Manipulator	3
1.3.2	2.2 Sensors	3
1.3.3	2.3 Vision & Lighting	4
1.3.4	2.4 Compute Hardware	4
1.3.5	2.5 Power & Electrical	4
1.3.6	2.6 Networking	4
1.4	3. Layer 2: Embedded / Firmware / Drivers	5
1.4.1	3.1 Operating Systems	5
1.4.2	3.2 Device Drivers & SDKs	5
1.4.3	3.3 Firmware	6
1.5	4. Layer 3: Perception & Sensor Processing	6
1.5.1	4.1 Computer Vision Libraries	6
1.5.2	4.2 Image Processing	6
1.6	5. Layer 4: Robotics Middleware (ROS2)	6
1.6.1	5.1 ROS2 Distribution	6
1.6.2	5.2 Core ROS2 Packages	7
1.6.3	5.3 ROS2 Communication	7
1.6.4	5.4 Custom ROS2 Packages	7
1.7	6. Layer 5: AI / Machine Learning	7
1.7.1	6.1 Deep Learning Frameworks	7
1.7.2	6.2 Pre-Trained Models & Libraries	7
1.7.3	6.3 Training & MLOps	8
1.8	7. Layer 6: Application / Business Logic	8
1.8.1	7.1 Application Frameworks	8
1.8.2	7.2 Task Orchestration	8
1.8.3	7.3 Analytics & Reporting	9
1.9	8. Layer 7: User Interface & Visualization	9
1.9.1	8.1 Robotics Visualization	9
1.9.2	8.2 Dashboards & Monitoring	9
1.9.3	8.3 Web Frontend	9

1.10	9. Cross-Cutting: DevOps & Infrastructure	10
1.10.1	9.1 Version Control & CI/CD	10
1.10.2	9.2 Build & Dependency Management	10
1.10.3	9.3 Testing Frameworks	10
1.11	10. Cross-Cutting: Security	10
1.11.1	10.1 Authentication & Authorization	10
1.11.2	10.2 Encryption & Secure Communication	11
1.11.3	10.3 Security Monitoring	11
1.12	11. Cross-Cutting: Logging & Observability	11
1.12.1	11.1 Logging	11
1.12.2	11.2 Centralized Logging (ELK Stack)	11
1.12.3	11.3 Distributed Tracing	11
1.12.4	11.4 Performance Monitoring	12
1.13	12. Development Tools	12
1.13.1	12.1 IDEs & Editors	12
1.13.2	12.2 Debugging & Profiling	12
1.13.3	12.3 Documentation	12
1.14	13. Hardware Tools & Equipment	13
1.14.1	13.1 Development & Testing	13
1.15	14. Technology Stack Summary Table	13
1.16	15. Technology Selection Rationale	13
1.17	16. Dependency Graph (Simplified)	14
1.18	17. Version Pinning & Compatibility	14
1.18.1	17.1 Critical Version Constraints	14
1.18.2	17.2 Package Managers	15
1.19	18. Deployment Architecture	15
1.19.1	18.1 Single-Machine Deployment (Development/Small-Scale)	15
1.19.2	18.2 Distributed Deployment (Production/Cloud-Connected)	16
1.20	19. Total Cost of Ownership (TCO) Estimate	16
1.21	20. Conclusion	17

1 Technical Stack - Vision-Based Pick and Place System

1.1 Overview

This document provides a comprehensive specification of all **hardware, software, frameworks, libraries, and tools** used in the vision-based pick-and-place robotic system, organized by architectural layer.

1.2 1. Architecture Layers

LAYER 7: USER INTERFACE

Web Dashboard, Mobile App, RViz2, Grafana, Foxglove

LAYER 6: APPLICATION / BUSINESS LOGIC
Task Orchestrator, Workflow Manager, Analytics Engine

LAYER 5: AI / MACHINE LEARNING
Object Detection, Pose Estimation, Grasp Planning (AI)

LAYER 4: ROBOTICS MIDDLEWARE (ROS2)
MoveIt2, Nav2, ros2_control, TF2, Image Transport

LAYER 3: PERCEPTION & SENSOR PROCESSING
OpenCV, PCL, RealSense SDK, Image Processing Nodes

LAYER 2: EMBEDDED / FIRMWARE / DRIVERS
Motor Drivers, EtherCAT Master, Camera Drivers, MCU Firmware

LAYER 1: HARDWARE
Robot, Sensors, Actuators, Power, Network

1.3 2. Layer 1: Hardware

1.3.1 2.1 Robot Manipulator

Component	Specification	Vendor/Model	Quantity
Robot Arm (6-DOF)	Payload: 5kg, Reach: 850mm, Repeatability: ± 0.1 mm	Universal Robots UR5e / ABB IRB 1200	1
Gripper	Parallel jaw, 85mm stroke, 100N grip force	Robotiq 2F-85 / Schunk PGN-plus-E	1
Robot Controller	Built-in controller with EtherCAT/Modbus support	UR Control Box / ABB Controller	1

1.3.2 2.2 Sensors

Component	Specification	Vendor/Model	Quantity
RGB-D Camera	1920x1080 RGB @ 30fps, Depth range: 0.3-3m	Intel RealSense D435i	1-2
Force/Torque Sensor	6-axis, $\pm 100\text{N}$, $\pm 10\text{Nm}$, Resolution: 0.1N/0.01Nm	ATI Mini45 / OnRobot HEX	1
Proximity Sensor	Inductive, 8mm sensing distance	Omron E2E-X8ME1	2
Emergency Stop Button	Safety-rated (SIL 2), dual-channel	Schneider XB7NS8445	1

1.3.3 2.3 Vision & Lighting

Component	Specification	Vendor/Model	Quantity
LED Ring Light	5000K, 2000 lumen, dimmable	CCS LDR2-74SW2-WHI	1
Camera Mount	Adjustable angle, vibration-damped	Custom / Manfrotto Magic Arm	1

1.3.4 2.4 Compute Hardware

Component	Specification	Vendor/Model	Quantity
Vision Processing	GPU: 512 CUDA cores, 8GB RAM, Jetson Linux	NVIDIA Jetson Xavier NX	1
Main Controller	x86 CPU (4-core, 3.5GHz), 16GB RAM, SSD	Intel NUC / Dell Optiplex	1
Microcontroller (I/O)	ARM Cortex-M4, 168MHz, 512KB Flash	STM32F407VG	1

1.3.5 2.5 Power & Electrical

Component	Specification	Vendor/Model	Quantity
Power Supply	48V DC, 20A, 960W	Mean Well RSP-1000-48	1
DC-DC Converter (12V)	48V \rightarrow 12V, 10A, 120W	Mean Well SD-100B-12	1
DC-DC Converter (5V)	12V \rightarrow 5V, 5A, 25W	RECOM R-78E5.0-1.0	2
Circuit Breaker	25A, 2-pole	Eaton FAZ-C25/2	1

1.3.6 2.6 Networking

Component	Specification	Vendor/Model	Quantity
EtherCAT Switch	5-port, managed, industrial	Beckhoff EK1100	1
Ethernet Switch	8-port, Gigabit, unmanaged	Netgear GS108	1
WiFi Router (optional)	Dual-band, 802.11ac	TP-Link Archer C7	1

1.4 3. Layer 2: Embedded / Firmware / Drivers

1.4.1 3.1 Operating Systems

Component	Technology	Version	Purpose
Real-Time Linux	Ubuntu 22.04 with RT-Preempt kernel	5.15-rt	Main controller (ros2_control)
Jetson Linux	NVIDIA L4T (Linux for Tegra)	35.3.1 (Ubuntu 20.04)	Vision processing
Bare-Metal RTOS	FreeRTOS	10.5.1	STM32 MCU (I/O)

1.4.2 3.2 Device Drivers & SDKs

Component	Technology	Version	Purpose
RealSense SDK	librealsense2	2.54.1	Camera interface
EtherCAT Master	IgH EtherCAT Master	1.5.2	Motor driver communication
F/T Sensor Driver	ATI DAQ C Library	2.3.0	Force/torque data acquisition
GPIO Library	libgpiod	1.6.3	Digital I/O (E-stop, sensors)
CUDA	NVIDIA CUDA Toolkit	11.4	GPU acceleration
TensorRT	NVIDIA TensorRT	8.5.1	AI inference optimization

1.4.3 3.3 Firmware

Component	Technology	Development Environment	Purpose
STM32 MCU Firmware	C/C++, HAL, FreeRTOS	STM32CubeIDE	Low-level I/O control
Motor Drive Firmware	Proprietary (Beckhoff TwinCAT)	TwinCAT 3	Servo drive configuration

1.5 4. Layer 3: Perception & Sensor Processing

1.5.1 4.1 Computer Vision Libraries

Library	Version	Purpose	Language
OpenCV	4.8.0	Image processing, calibration, feature detection	C++, Python
Point Cloud Library (PCL)	1.13.1	3D point cloud processing, segmentation, ICP	C++
Open3D	0.17.0	Point cloud visualization, registration	Python
librealsense2	2.54.1	RealSense camera SDK	C++, Python

1.5.2 4.2 Image Processing

Tool/Library	Version	Purpose	Language
NumPy	1.24.3	Array operations, image manipulation	Python
SciPy	1.11.1	Signal processing, filtering	Python
Pillow (PIL)	10.0.0	Image I/O, format conversion	Python
scikit-image	0.21.0	Advanced image processing algorithms	Python

1.6 5. Layer 4: Robotics Middleware (ROS2)

1.6.1 5.1 ROS2 Distribution

Component	Version	Purpose
ROS2 Humble Hawksbill	Humble (LTS)	Base middleware for all nodes
rclcpp	Humble	C++ client library
rclpy	Humble	Python client library

1.6.2 5.2 Core ROS2 Packages

Package	Version	Purpose
MoveIt2	2.5.5	Motion planning, IK, collision checking
ros2_control	2.27.0	Real-time control framework
ros2_controllers	2.27.0	PID, trajectory, admittance controllers
tf2	0.25.2	Coordinate frame transforms
image_transport	3.1.7	Compressed image streaming
cv_bridge	3.2.1	OpenCV ROS message conversion
pcl_ros	2.4.0	PCL ROS message conversion

1.6.3 5.3 ROS2 Communication

Package	Version	Purpose
rmw_cyclonedds_cpp	1.3.4	DDS implementation (middleware)
rosbridge_suite	1.3.2	WebSocket bridge for web UIs
ros2_tracing	4.1.1	Performance tracing (LTTng)

1.6.4 5.4 Custom ROS2 Packages

Package Name	Language	Purpose
vision_pipeline	Python	Object detection, pose estimation nodes
grasp_planner	C++	Grasp synthesis and ranking
task_orchestrator	Python	State machine, task sequencing
hardware_interface	C++	ros2_control hardware interface for robot

1.7 6. Layer 5: AI / Machine Learning

1.7.1 6.1 Deep Learning Frameworks

Framework	Version	Purpose	Backend
PyTorch	2.0.1	Model training, object detection, pose estimation	CUDA 11.4
TensorFlow	2.13.0	Alternative framework for grasp planning	CUDA 11.4
ONNX Runtime	1.15.1	Cross-framework inference	CUDA, CPU
TensorRT	8.5.1	Optimized inference on NVIDIA GPUs	CUDA 11.4

1.7.2 6.2 Pre-Trained Models & Libraries

Model/Library	Version	Purpose	Source
YOLOv8	8.0.20	Real-time object detection	Ultralytics
Mask R-CNN	-	Instance segmentation (if needed)	Detectron2
PVNet	-	6DoF pose estimation	Research repo
GraspNet	-	Grasp pose prediction	Research repo
Segment Anything (SAM)	1.0	Zero-shot segmentation (optional)	Meta AI

1.7.3 6.3 Training & MLOps

Tool	Version	Purpose
MLflow	2.5.0	Experiment tracking, model registry
Weights & Biases (W&B)	0.15.8	Experiment tracking, collaboration
DVC (Data Version Control)	3.15.0	Dataset versioning
Label Studio	1.8.2	Data annotation (bounding boxes, keypoints)
Roboflow	-	Dataset management, augmentation

1.8 7. Layer 6: Application / Business Logic

1.8.1 7.1 Application Frameworks

Framework	Version	Purpose	Language
FastAPI	0.103.0	RESTful API backend	Python
gRPC	1.57.0	High-performance RPC	C++, Python
Redis	7.0.12	In-memory cache, pub/sub	-
PostgreSQL	15.3	Relational database (tasks, logs, configs)	SQL
InfluxDB	2.7.1	Time-series database (sensor data)	-
MongoDB	6.0.8	Document database (AI inference logs)	-

1.8.2 7.2 Task Orchestration

Tool/Library	Version	Purpose
BehaviorTree.CPP	4.5.1	Behavior tree execution engine
SMACH (ROS)	2.5.0	State machine library (deprecated, use BT.CPP)
Celery	5.3.1	Distributed task queue (if async tasks needed)

1.8.3 7.3 Analytics & Reporting

Tool	Version	Purpose
Pandas	2.0.3	Data analysis, reporting
Matplotlib	3.7.2	Data visualization (plots)
Seaborn	0.12.2	Statistical visualization
Jupyter Notebook	7.0.2	Interactive data analysis

1.9 8. Layer 7: User Interface & Visualization

1.9.1 8.1 Robotics Visualization

Tool	Version	Purpose
RViz2	Humble	3D robot visualization, TF, point clouds
Foxglove Studio	1.68.0	Modern ROS visualization (alternative to RViz)
Gazebo (Classic)	11.13.0	Physics simulation
Gazebo (Ignition/Harmonic)	Garden	Next-gen simulation
RobotStudio (ABB)	2023.2	ABB-specific simulation (if using ABB robot)

1.9.2 8.2 Dashboards & Monitoring

Tool	Version	Purpose
Grafana	10.0.3	Real-time dashboards, time-series visualization
Prometheus	2.45.0	Metrics collection, alerting
Node Exporter	1.6.1	System metrics (CPU, RAM, disk)
ROS2 Diagnostics	Humble	Robot health monitoring

1.9.3 8.3 Web Frontend

Framework	Version	Purpose
React	18.2.0	Web UI framework
Next.js	13.4.12	React framework with SSR
TypeScript	5.1.6	Type-safe JavaScript
TailwindCSS	3.3.3	Utility-first CSS framework
rosbridge	1.3.2	WebSocket connection to ROS2
roslibjs	1.3.0	JavaScript library for ROS communication

1.10 9. Cross-Cutting: DevOps & Infrastructure

1.10.1 9.1 Version Control & CI/CD

Tool	Version	Purpose
Git	2.40.1	Source code version control
GitHub / GitLab	-	Code hosting, issue tracking
GitHub Actions	-	CI/CD pipelines
Docker	24.0.5	Containerization
Docker Compose	2.20.2	Multi-container orchestration
Kubernetes (optional)	1.27.4	Container orchestration (for cloud deployment)

1.10.2 9.2 Build & Dependency Management

Tool	Version	Purpose
colcon	0.14.1	ROS2 build tool
CMake	3.26.4	C/C++ build system
pip	23.2.1	Python package manager
conda / mamba	23.5.0	Python environment manager
rosdep	0.22.2	ROS dependency management

1.10.3 9.3 Testing Frameworks

Framework	Version	Purpose
pytest	7.4.0	Python unit testing
Google Test (gtest)	1.13.0	C++ unit testing
ros2 launch_testing	Humble	ROS2 integration testing
unittest (Python)	Built-in	Python standard testing library
Locust	2.15.1	Load testing (API endpoints)

1.11 10. Cross-Cutting: Security

1.11.1 10.1 Authentication & Authorization

Tool	Version	Purpose
OAuth2	-	Authentication protocol
Keycloak	22.0.1	Identity and access management
JWT (JSON Web Tokens)	-	Stateless authentication tokens
bcrypt	4.0.1	Password hashing

1.11.2 10.2 Encryption & Secure Communication

Tool	Version	Purpose
OpenSSL	3.0.9	TLS/SSL, cryptography
Let's Encrypt	-	Free SSL certificates
mTLS (mutual TLS)	-	Bidirectional authentication (gRPC)

1.11.3 10.3 Security Monitoring

Tool	Version	Purpose
Fail2ban	1.0.2	Intrusion prevention (ban failed logins)
Snort / Suricata	3.1.65.0	Network intrusion detection
OSSEC	3.7.0	Host-based intrusion detection
Wireshark	4.0.8	Network traffic analysis

1.12 11. Cross-Cutting: Logging & Observability

1.12.1 11.1 Logging

Tool	Version	Purpose
syslog-ng	3.38.1	System logging
Logrotate	3.20.1	Log file rotation
Python logging	Built-in	Application-level logging
rclepp logging	Humble	ROS2 C++ logging

1.12.2 11.2 Centralized Logging (ELK Stack)

Tool	Version	Purpose
Elasticsearch	8.9.0	Log indexing and search
Logstash	8.9.0	Log ingestion and transformation
Kibana	8.9.0	Log visualization and dashboards
Filebeat	8.9.0	Log shipping agent

1.12.3 11.3 Distributed Tracing

Tool	Version	Purpose
Jaeger	1.47.0	Distributed tracing
OpenTelemetry	1.20.0	Observability framework (metrics, traces, logs)
Zipkin	2.24.2	Alternative to Jaeger

1.12.4 11.4 Performance Monitoring

Tool	Version	Purpose
Prometheus	2.45.0	Metrics collection
Grafana	10.0.3	Metrics visualization
Node Exporter	1.6.1	Hardware/OS metrics
cAdvisor	0.47.2	Container metrics
NVIDIA-SMI	530.30.02	GPU monitoring

1.13 12. Development Tools

1.13.1 12.1 IDEs & Editors

Tool	Version	Purpose
VS Code	1.81.1	Primary IDE (C++, Python)
CLion	2023.2	C++ IDE (JetBrains)
PyCharm	2023.2	Python IDE (JetBrains)
Vim / Neovim	9.0	Terminal-based editor

1.13.2 12.2 Debugging & Profiling

Tool	Version	Purpose
GDB	13.2	C++ debugger
Valgrind	3.21.0	Memory leak detection
perf	6.3	Linux performance profiling
NVIDIA Nsight Systems	2023.2.3	GPU profiling
ros2 topic / service / action CLI	Humble	ROS2 debugging tools

1.13.3 12.3 Documentation

Tool	Version	Purpose
Doxygen	1.9.7	C++ API documentation
Sphinx	7.1.2	Python documentation
MkDocs	1.5.2	Markdown-based documentation
Mermaid	10.3.1	Diagrams as code (in markdown)
PlantUML	1.2023.10	UML diagrams

1.14 13. Hardware Tools & Equipment

1.14.1 13.1 Development & Testing

Tool	Purpose
Oscilloscope	Signal analysis (motor drivers, sensors)
Logic Analyzer	Digital signal debugging (EtherCAT, SPI, I2C)
Multimeter	Voltage, current, resistance measurement
Power Analyzer	Power consumption measurement
3D Printer	Prototype gripper parts, fixtures
CMM (Coordinate Measuring Machine)	Precision position measurement ($\pm 0.01\text{mm}$)

1.15 14. Technology Stack Summary Table

Layer	Core Technologies
Hardware	UR5e / ABB robot, RealSense D435i, ATI F/T sensor, Jetson Xavier, Intel NUC
Firmware	RT-Linux (5.15-rt), Jetson Linux (L4T), FreeRTOS, EtherCAT Master (IgH)
Perception	OpenCV 4.8, PCL 1.13, librealsense2 2.54, Open3D 0.17
Middleware	ROS2 Humble, MoveIt2 2.5, ros2_control 2.27, TF2 0.25
AI/ML	PyTorch 2.0, TensorRT 8.5, YOLOv8, PVNet, GraspNet, MLflow
Application	FastAPI, gRPC, PostgreSQL, Redis, InfluxDB, BehaviorTree.CPP
UI/Visualization	RViz2, Foxglove, Grafana, Prometheus, React, Next.js
DevOps	Docker, GitHub Actions, colcon, CMake, pytest, gtest
Security	OAuth2, Keycloak, OpenSSL, Fail2ban, Snort
Observability	ELK Stack (Elasticsearch, Logstash, Kibana), Jaeger, OpenTelemetry, Prometheus

1.16 15. Technology Selection Rationale

Category	Selected	Alternatives	Reason for Selection
Robot	ROS2 Humble	ROS1, YARP, OROCOS	Industry standard, active development, real-time support
Middleware	MoveIt2	OMPL	Integrated with ROS2, mature, good
Motion		standalone, Pilz	community support
Planning	OpenCV	VTK, SimpleCV	Comprehensive, optimized, large
Vision			community
Library	PyTorch	TensorFlow, JAX	Research-friendly, dynamic graphs, good
Deep			ONNX/TensorRT export
Learning			

Category	Selected	Alternatives	Reason for Selection
Object Detection	YOLOv8	Faster R-CNN, SSD	Best speed/accuracy trade-off for real-time
Database (OLTP)	PostgreSQL	MySQL, MariaDB	Feature-rich, extensible, excellent JSON support
Time-Series DB	InfluxDB	TimescaleDB, Prometheus	Purpose-built for time-series, easy integration
Message Queue	Redis	RabbitMQ, Kafka	Low latency, in-memory, pub/sub support
API Framework	FastAPI	Flask, Django	Fast, async, auto-generated API docs
Container	Docker	Podman, LXC	Industry standard, extensive ecosystem
CI/CD	GitHub Actions	GitLab CI, Jenkins	Integrated with GitHub, easy to configure
Monitoring	Grafana + Prometheus	Datadog, New Relic	Open-source, flexible, large community
Logging	ELK Stack	Splunk, Graylog	Open-source, powerful search, scalable

1.17 16. Dependency Graph (Simplified)

```

Application Layer (FastAPI, React)
  ↓
ROS2 Middleware (MoveIt2, ros2_control)
  ↓
Perception (OpenCV, PCL) ↔ AI/ML (PyTorch, TensorRT)
  ↓
Drivers (librealsense2, EtherCAT Master)
  ↓
Hardware (Robot, Sensors, Actuators)

```

Cross-Cutting: - **Monitoring:** Prometheus, Grafana (all layers) - **Logging:** ELK Stack (all layers) - **Security:** OAuth2, TLS (Application, Middleware) - **DevOps:** Docker, GitHub Actions (build, deploy)

1.18 17. Version Pinning & Compatibility

1.18.1 17.1 Critical Version Constraints

Dependency	Version Constraint	Reason
ROS2	= Humble (LTS)	Long-term support, stable until 2027
Ubuntu	= 22.04 LTS	Required for ROS2 Humble
CUDA	= 11.4	Compatible with TensorRT 8.5 and PyTorch 2.0
TensorRT	= 8.5.x	Optimized for Jetson Xavier

Dependency	Version Constraint	Reason
Python	= 3.10	Default for Ubuntu 22.04, ROS2 Humble support
OpenCV	>= 4.5, < 5.0	API stability, avoid breaking changes
MoveIt2	>= 2.5, < 3.0	Humble-compatible

1.18.2 17.2 Package Managers

Ecosystem	Package Manager	Lock File
ROS2	rosdep, apt	package.xml, rosdep.yaml
Python	pip, conda	requirements.txt, environment.yml
C++	apt, vcpkg	CMakeLists.txt
Node.js (UI)	npm, yarn	package-lock.json
Docker	Docker Compose	docker-compose.yml

1.19 18. Deployment Architecture

1.19.1 18.1 Single-Machine Deployment (Development/Small-Scale)

Intel NUC (Ubuntu 22.04 RT)

ROS2 Humble Nodes (MoveIt, control)

PostgreSQL, Redis, InfluxDB

USB 3.0

NVIDIA Jetson Xavier NX

Vision Pipeline (YOLOv8, Pose Est.)

EtherCAT

Servo Drives (EtherCAT slaves)

1.19.2 18.2 Distributed Deployment (Production/Cloud-Connected)

Cloud (AWS / Azure)

- MLflow (model registry)
- Grafana (dashboards)
- Elasticsearch (log aggregation)

HTTPS

Edge Gateway (Intel NUC)

- FastAPI (REST API)
- Data uplink to cloud

ROS2 DDS

Robot Controller (RT Linux)

- MoveIt2, ros2_control, TF2

Vision Processor (Jetson Xavier)

- YOLOv8, Pose Estimation

1.20 19. Total Cost of Ownership (TCO) Estimate

Category	Item	Cost (USD)	Quantity	Total
Hardware	UR5e Robot Arm	\$35,000	1	\$35,000
	Robotiq 2F-85 Gripper	\$5,000	1	\$5,000
	RealSense D435i	\$350	1	\$350
	ATI Mini45 F/T Sensor	\$2,500	1	\$2,500
	NVIDIA Jetson Xavier NX	\$500	1	\$500
	Intel NUC	\$800	1	\$800
	Power Supply, Electrical	\$500	1	\$500
	Hardware Subtotal			\$44,650
Software (Licenses)	All open-source (ROS2, PyTorch, etc.)	\$0	-	\$0
	Windows/proprietary tools (if any)	\$1,000	1	\$1,000
	Software Subtotal			\$1,000

Category	Item	Cost (USD)	Quantity	Total
Development	Engineering (6 months, 2 FTEs)	\$100,000	1	\$100,000
	Development Subtotal			\$100,000
Operations (Annual)	Maintenance, electricity	\$2,000/year	-	\$2,000/year
Total (Initial)				\$145,650

1.21 20. Conclusion

This technical stack represents a **comprehensive, production-ready** architecture for a vision-based pick-and-place system, featuring: - **Open-source** core (ROS2, OpenCV, PyTorch) → minimal licensing costs - **Real-time** performance (RT-Linux, EtherCAT, 1kHz control) - **Scalability** (Docker, Kubernetes-ready, cloud integration) - **Observability** (Prometheus, Grafana, ELK, Jaeger) - **Security** (OAuth2, TLS, intrusion detection) - **Flexibility** (modular architecture, easy to swap components)

Next Steps: 1. Procure hardware based on specifications 2. Set up development environment (Docker containers) 3. Implement CI/CD pipeline (GitHub Actions) 4. Begin software development following architecture docs

Document Status: Complete **Last Updated:** 2025-10-18 **Author:** Technical Architecture Team **Review Status:** Pending Review