TOSCA Architecture Documentation Index

Architecture Diagrams

Last Updated: 2025-11-04 **Version:** 1.0 **Purpose:** Comprehensive guide to TOSCA architecture documentation

---

Quick Navigation

Document Type	Primary Files	Purpose
System Overview	`01_system_overview.md`	Architecture introduction, technology stack
C4 Diagrams	`diagrams/c4-*.puml`	Visual architecture (Context, Container, Component)
ADRs	`ADR-*.md`	Architecture decision records
Security	`SECURITY_THREAT_MODEL.md`	STRIDE threat analysis
Quality	`QUALITY_ATTRIBUTES.md`	Performance, reliability, safety metrics
Data	`02_database_schema.md`, `diagrams/database-schema-erd.puml`	Database design
Safety	`03_safety_system.md`, `SAFETY_SHUTDOWN_POLICY.md`	Safety architecture

Architecture Documentation Structure

1. Core Architecture Documents

System Overview and Fundamentals: - `01_system_overview.md` - Complete system architecture, technology stack, hardware components - `02_database_schema.md` - SQLite schema, entity relationships, data models - `03_safety_system.md` - Safety philosophy, interlocks, state machine - `04_treatment_protocols.md` - Protocol data model, execution engine - `05_image_processing.md` - Computer vision algorithms, ring detection, focus measurement - `06_protocol_builder.md` - Action-based protocol engine (current design) - `07_safety_watchdog.md` - Hardware watchdog implementation - `08_security_architecture.md` - Security design (Phase 6 roadmap) - `09_test_architecture.md` - Testing infrastructure, mock patterns - `10_concurrency_model.md` - Thread safety, async patterns - `11_event_logging.md` - Two-tier logging strategy (JSONL + SQLite) - `12_recording_manager.md` - Video recording, session data management - `13_calibration_procedures.md` - Device calibration workflows

2. Visual Architecture (C4 Model Diagrams)

Location: `diagrams/`

System Context (Level 1): - `c4-01-system-context.puml` - TOSCA and external systems (camera, laser, actuator, GPIO)

Container Architecture (Level 2): - `c4-02-container.puml` - PyQt6 UI, Application Core, HAL, Database, File System

Component Architecture (Level 3): - `c4-03-component-core.puml` - Application Core: Safety Manager, Session Manager, Protocol Engine - `c4-04-component-hal.puml` - Hardware Abstraction Layer: Camera, Laser, TEC, Actuator, GPIO controllers

Sequence Diagrams: - `sequence-treatment-workflow.puml` - Normal treatment execution flow

Data Architecture: - `database-schema-erd.puml` - Entity-Relationship Diagram (ERD) - `data-architecture.puml` - Two-tier logging strategy visualization

Generating Diagrams:

```
cd docs/architecture/diagrams
./generate_diagrams.sh  # Linux/macOS
generate_diagrams.bat   # Windows
```

3. Architecture Decision Records (ADRs)

Template: - `ADR-TEMPLATE.md` - Standard format for documenting architectural decisions

Existing ADRs: - `ADR-001-protocol-consolidation.md` - Action-based vs step-based protocol model - `ADR-002-dependency-injection-pattern.md` - Hardware controller dependency injection - `ADR-003-pyqt6-gui-framework.md` - PyQt6 vs PySide6 vs Tkinter selection - `ADR-004-sqlite-database.md` - SQLite vs PostgreSQL vs MySQL vs MongoDB - `ADR-005-arduino-gpio-migration.md` - Arduino Uno vs FT232H vs Raspberry Pi Pico - `ADR-006-selective-shutdown-policy.md` - Selective shutdown vs total shutdown on safety fault

Creating New ADRs: 1. Copy `ADR-TEMPLATE.md` to `ADR-XXX-title.md` 2. Fill in all sections (Context, Options, Decision, Consequences) 3. Add cross-references to related ADRs 4. Commit to version control

4. Security and Compliance

Security Threat Model: - `SECURITY_THREAT_MODEL.md` - STRIDE analysis, threat scenarios, mitigation roadmap - Spoofing: User authentication missing (Phase 6 required) - Tampering: Database encryption missing (SQLCipher in Phase 6) - Repudiation: Non-repudiation not implemented (electronic signatures in Phase 6) - Information Disclosure: Database unencrypted (CRITICAL for clinical use) - Denial of Service: Disk space exhaustion risk (monitoring required) - Elevation of Privilege: No RBAC (role-based access control in Phase 6)

Compliance Requirements: - FDA 21 CFR Part 11: Electronic records/signatures (MISSING: Phase 6 required) - HIPAA: Database encryption, access control (MISSING: Phase 6 required) - IEC 62304: Software lifecycle (IMPLEMENTED)

Safety Policy: - `SAFETY_SHUTDOWN_POLICY.md` - Selective shutdown rationale, systems preserved vs disabled

5. Quality Attributes

Quality Attributes Document: - `QUALITY_ATTRIBUTES.md` - Performance, reliability, safety, maintainability, portability, usability

Key Metrics (v0.9.12-alpha): - Performance: 30 FPS camera, <10ms safety response, 100Hz photodiode sampling - Reliability: 0 crashes in 250+ hours, 100% fault tolerance test pass rate - Safety: 100% safety interlock test pass rate, comprehensive hardware interlocks - Maintainability: Grade A architecture, 85% test coverage, modular design - Security: **Grade D** (encryption/authentication missing - Phase 6 required)

6. Supporting Documentation

Hardware: - `hardware_controller_base_usage.md` - HardwareControllerBase pattern guide - `../hardware/HARDWARE_CONFIG_SUMMARY.md` - Device specifications, COM port configuration

Implementation Status: - `00_IMPLEMENTATION_STATUS.md` - Component completion status (consider consolidating)

Integration: - `11_asyncio_pyqt6_integration.md` - Asyncio + PyQt6 event loop integration

Documentation Workflow

Creating New Architecture Documentation

1. **Identify Documentation Need:**
   - New architectural component?
   - Significant design decision?
   - Security/quality concern?
2. **Choose Documentation Type:**
   - **Core Architecture:** Add to numbered sequence (14_*.md)
   - **Decision:** Create ADR using `ADR-TEMPLATE.md`
   - **Diagram:** Create `.puml` file in `diagrams/`
   - **Security:** Update `SECURITY_THREAT_MODEL.md`
   - **Quality:** Update `QUALITY_ATTRIBUTES.md`
3. **Document Content:**
   - Use consistent markdown formatting
   - Include code examples where applicable
   - Add cross-references to related documents
   - Include metadata (date, version, status)
4. **Generate Diagrams:**

```
cd diagrams/
./generate_diagrams.sh
```

5. **Review and Commit:**

```
git add docs/architecture/*.md
git add docs/architecture/diagrams/*.puml
git add docs/architecture/diagrams/output/*.png
git commit -m "docs: add architecture documentation for [topic]"
```

Updating Existing Documentation

1. **Identify Outdated Content:**
   - Review "Last Updated" dates
   - Check against current implementation (v0.9.12-alpha)
   - Verify code examples still accurate
2. **Update Content:**
   - Modify markdown file
   - Update diagrams if architecture changed
   - Update version/date metadata
3. **Regenerate Diagrams:**

```
cd diagrams/
./generate_diagrams.sh
```

4. **Commit Changes:**

```
git add docs/architecture/
git commit -m "docs: update [document] with current implementation"
```

Documentation Review Cycle

Quarterly Review (Every 3 Months): 1. Review all architecture documents for accuracy 2. Update diagrams to reflect current state 3. Add ADRs for recent architectural decisions 4. Update security threat model with new risks 5. Validate quality attributes against current metrics

Phase Transitions: 1. Comprehensive review before phase completion 2. Update all documentation to match implementation 3. Archive temporal documents (move to `archive/`) 4. Create phase completion summary

---

Documentation Quality Standards

1. Completeness

- [ ] All architectural components documented
- [ ] All significant decisions captured in ADRs
- [ ] All diagrams current and accurate
- [ ] Security risks identified and documented
- [ ] Quality metrics validated and current

2. Consistency

- [ ] Consistent terminology across documents
- [ ] Cross-references accurate and up-to-date
- [ ] Diagram naming consistent with code
- [ ] Metadata (date, version) current

3. Clarity

- [ ] Technical concepts explained clearly
- [ ] Code examples included where helpful
- [ ] Diagrams labeled and annotated
- [ ] Assumptions and constraints documented

4. Maintainability

- [ ] Diagrams stored as code (`.puml` files)
- [ ] Version controlled (all changes committed)
- [ ] Generation scripts automated
- [ ] Documentation review cycle established

Tools and Resources

Diagram Generation

- **PlantUML:** https://plantuml.com/
- **C4 Model:** https://c4model.com/
- **C4-PlantUML:** https://github.com/plantuml-stdlib/C4-PlantUML

Documentation Standards

- **Architecture Decision Records:** https://adr.github.io/
- **STRIDE Threat Modeling:** https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling-tool-threats
- **ISO/IEC 25010 Quality Model:** https://iso25000.com/index.php/en/iso-25000-standards/iso-25010

Medical Device Compliance

- **FDA 21 CFR Part 11:** https://www.fda.gov/regulatory-information/search-fda-guidance-documents/part-11-electronic-records-electronic-signatures-scope-and-application
- **IEC 62304:** https://www.iso.org/standard/38421.html
- **HIPAA Security Rule:** https://www.hhs.gov/hipaa/for-professionals/security/index.html

---

Contact

Documentation Owner: System Architect **Last Review:** 2025-11-04 **Next Review:** 2026-02-04 (Quarterly)

For questions or clarifications about architecture documentation, contact the development team.

Version History:

Version	Date	Changes
1.0	2025-11-04	Initial comprehensive architecture documentation index created