LEGACY SYSTEMS FOUNDATION

EPOCH+ 2038 Safe™ Standard

Time Resilience Movement

EPOCH+ 2038 Safe™ Standard

Issued by: Legacy Systems Foundation
Document Identifier: LSFE-2038-STD-1.0
Version: 1.6
Publication Date: August 2025
Status: Published Standard

Citation

This document should be cited as:
Legacy Systems Foundation. EPOCH+ 2038 Safe™ Standard. Version 1.0. LSFE-2038-STD-1.0. August 2025.

Revision and Maintenance

- **Supersession:** This standard will be superseded by future versions published by the Legacy Systems Foundation.
- **Transition Period:** Organisations certified under previous versions shall transition to new versions within **18 months** of publication.
- **Feedback:** Comments and proposed revisions may be submitted to the Legacy Systems Foundation Standards Committee.

Acknowledgements

The Legacy Systems Foundation acknowledges contributions from industry partners, researchers, and practitioners in developing this framework.

Foreword

Computing is built on time. Every transaction processed by a stock exchange, every train departure, every medical scan stored in a hospital record depends on the consistent, accurate, and predictable measurement of time. Yet history has shown us that timekeeping in computers is fragile.

In 1999, the world confronted the Y2K problem. It was a deceptively simple error: dates stored with two digits for the year would reset to "00," interpreted as 1900. Left unchecked, this could have caused mass failures. Through global coordination, billions of lines of code were inspected, patched, and tested. The feared catastrophe never materialised: not because the risk was overstated, but because the problem was taken seriously.

Other crises have been less successfully anticipated. In 2019, the GPS epoch rollover caused aircraft navigation systems to report incorrect positions and disrupted GPS-enabled infrastructure. In 2012, a poorly handled leap second crashed major websites, including Reddit and LinkedIn, as Linux kernels and Java applications entered unstable states. In each case, seemingly small issues in time representation cascaded into significant failures.

The Year 2038 Problem is the next systemic challenge. At 03:14:07 UTC on 19 January 2038, signed 32-bit integer time values will overflow, reverting to negative numbers. Instead of 2038, systems will interpret the date as 1901. The failure mode is unpredictable: some systems will stop entirely, others will silently corrupt data, while still others may miscalculate schedules, interest rates, or maintenance intervals.

The issue is particularly insidious because:

- It affects embedded devices with long lifespans (controllers in aircraft, medical devices, SCADA systems).
- It exists in software libraries still widely deployed (*time_t* in C, legacy database schemas).
  It can manifest today, wherever forward dates (beyond 2038) are processed.

Unlike Y2K, there is no global awareness campaign. No governments are mobilising resources. No mass remediation projects are underway. The problem advances quietly.

The Legacy Systems Foundation developed the EPOCH+ 2038 Safe™ Standard to provide the missing structure. It is not a patchwork of best practices but a coherent, certifiable framework. It gives organisations a roadmap to identify risks, migrate safely, validate resilience, and document compliance.

This document is modelled on the NIST Cybersecurity Framework: it is outcome-driven, sector-neutral, and adaptable. It is intended for executives, developers, auditors, and

regulators alike. Its adoption will mark not only preparedness for 2038, but a broader cultural shift toward time resilience: recognising that timekeeping is a critical dependency of modern infrastructure.

Contents

1. Scope and Applicability

1.1 General Scope

The EPOCH+ Standard applies to any system, component, or process that **stores, computes, transmits, or depends on time values**. This includes, but is not limited to:

- **Software applications and services** that rely on or generate timestamps.
- **Databases and schemas** containing time-dependent fields.
- **Operating systems, APIs, and middleware** providing time services.
- **Embedded and industrial devices** expected to remain in service into the 2040s or beyond.
- **Organisational governance processes**, including procurement, vendor management, and risk oversight, where decisions may affect time resilience.
- Systems and organisations within scope **shall** demonstrate the ability to process and validate time values beyond 19 January 2038 without error.

1.2 Sector Applicability

The Year 2038 Problem affects multiple sectors with differing levels of criticality. Sector profiles are non-exhaustive but illustrate where resilience is essential.

Financial Services

Banks, insurers, and trading platforms depend on accurate time for compliance, settlement, and contractual enforcement. Risks include:

- Loan systems: a 30-year mortgage issued in 2009 will remain active beyond 2038.
- Derivatives and long-dated contracts: may already trigger rollover errors.
- ATMs and transaction engines: timestamp errors could halt services or corrupt records.

Compliance Expectation: Financial institutions **shall** demonstrate that systems supporting contracts, settlements, and regulatory reporting are resilient beyond 2038.

Healthcare

Healthcare providers and device manufacturers rely on long-lived systems:

- Medical devices (e.g., pacemakers, infusion pumps) often use embedded kernels with 32-bit time.
- Electronic Health Records must preserve patient data across lifetimes.
- Scheduling systems may reject appointments beyond 2038.

Compliance Expectation: Healthcare organisations **shall** ensure that devices, records, and scheduling systems remain time-resilient to protect patient safety.

Industrial and Utilities

Utilities and industrial control systems rely on SCADA controllers and PLCs with lifespans of 20–40 years. Failures could:

- Shut down substations.
- Interrupt oil and gas pipelines.
- Corrupt monitoring of water treatment facilities.

Compliance Expectation: Operators of critical infrastructure **shall** implement mitigation strategies for non-upgradeable devices and validate time resilience in control systems.

Transport and Aviation

Transport and aviation depend on long-range scheduling and embedded systems:

- Flight planning and maintenance systems project decades ahead.
- Aircraft subsystems may be vulnerable to rollover.
- GPS and air traffic control must operate without date corruption.

Compliance Expectation: Aviation and transport operators **shall** validate that planning, scheduling, and safety-critical systems are resilient beyond 2038.

Government and Defence

Governments and defence organisations maintain long-lived registries and critical systems:

- National registries (birth, death, taxation) require centuries of continuity.
- Defence logistics depend on embedded systems in vehicles and weapons.
- Failure could disrupt essential services or compromise security.

Compliance Expectation: Government and defence bodies **shall** ensure continuity of registries, logistics, and critical systems beyond 2038.

Cloud and SaaS Providers

Cloud platforms and managed services multiply the impact of unsafe time handling:

- Databases, message queues, and IoT services may propagate errors across tenants.
- Outages in cloud time services could simultaneously affect thousands of organisations.

Compliance Expectation: Cloud and SaaS providers **shall** attest to compliance and provide guarantees that tenant workloads are not exposed to unsafe time services.

2. Normative References

The following standards, frameworks, and publications are considered **normative references** for the EPOCH+ Standard. They provide the technical, procedural, and governance foundation upon which this framework is built.

Although the EPOCH+ Standard is designed to stand independently, implementers **should** consult these references to ensure alignment with existing compliance, assurance, and risk management practices.

ISO/IEC 8601: Representation of Dates and Times

ISO/IEC 8601 defines internationally agreed notations for dates and times. It establishes unambiguous formats such as **YYYY-MM-DD** for dates and **YYYY-MM-DDThh:mm:ssZ** for date-time strings.

Adoption of ISO 8601 eliminates confusion caused by localised formats (e.g., US *MM/DD/YY* vs EU *DD/MM/YY*) and is essential for interoperability across systems and jurisdictions.
EPOCH+ Requirement: ISO 8601 **shall** be the canonical exchange format for transmitting time values between systems.

IEEE Std 1003.1 (POSIX): Portable Operating System Interface

POSIX 1003.1 defines the C programming API for many operating systems, including the **time_t** data type. Traditionally implemented as a signed 32-bit integer, time_t is the root cause of the Year 2038 Problem.

POSIX is therefore both a **risk origin** and a **remediation reference**.
EPOCH+ Requirement: Implementers **shall** migrate away from 32-bit time_t to safe alternatives, such as 64-bit implementations or language-native high-precision APIs.

ISO/IEC 27001: Information Security Management Systems (ISMS)

ISO/IEC 27001 defines governance structures for information security, including risk assessment and continuous improvement. While it does not address the Year 2038 Problem directly, its control structure is highly relevant.

For example, Annex A.12.1 (Operational Procedures and Responsibilities) aligns with the testing and validation outcomes defined in EPOCH+.

EPOCH+ Expectation: Organisations **should** integrate time resilience into their existing ISMS controls and certification activities.

NIST Cybersecurity Framework (CSF)

The NIST CSF provides a flexible, outcome-driven approach to managing risk, structured into five functions: **Identify, Protect, Detect, Respond, Recover**.

The EPOCH+ Standard mirrors this outcome-driven style, enabling seamless integration into existing NIST-aligned programmes without duplication of effort.

ISO/IEC 25010: Systems and Software Quality Models

This standard defines key software quality characteristics, including **reliability, maintainability, and portability**.

Time resilience maps closely to these attributes:

- *Reliability*: Correct operation under defined conditions for a defined period.
- *Maintainability*: Ease of addressing defects and implementing updates.

EPOCH+ Expectation: Organisations **shall** treat time resilience as a measurable, auditable quality attribute.

OWASP Secure Coding Practices

The Open Worldwide Application Security Project (OWASP) provides widely adopted guidance for secure coding.

Although not focused on time resilience, its principles: avoiding unsafe functions, validating input, and applying rigorous testing: apply directly.
EPOCH+ Expectation: Developer coding standards **should** incorporate OWASP principles alongside EPOCH+ requirements.

Additional References

The following publications are not mandatory but are recommended to support sector-specific implementations:

- **RFC 3339**: Internet profile of ISO 8601.
- **CERT Secure Coding Standards**: Language-specific guidelines for safe time handling.

- **IEC 62443**: Security for industrial control systems (relevant to SCADA and embedded environments).
- **FDA Guidance on Medical Device Software**: Regulatory expectations for resilience in healthcare contexts.

Summary

These references collectively form the **technical and governance backdrop** against which EPOCH+ is positioned. They ensure that compliance with the EPOCH+ Standard integrates seamlessly into wider programmes such as ISO 27001 certification, NIST CSF adoption, and sector-specific regulation.

3. Terms and Definitions

For the purposes of this standard, the following terms and definitions apply. Unlike a glossary, this section is *normative*: terms are defined precisely to remove ambiguity during audits and implementation.

Epoch
A fixed reference point for time calculations. In Unix-like systems, the epoch is defined as 1 January 1970 at 00:00:00 UTC. All subsequent time values are represented as the number of seconds since this epoch.

Year 2038 Problem
A systemic computing issue caused by the overflow of signed 32-bit integers used to represent time since the Unix epoch. At 03:14:07 UTC on 19 January 2038, these integers will exceed their maximum value and roll over into negative numbers, corresponding to 13 December 1901.

Rollover
The event of exceeding the maximum representable value of a counter, causing it to reset or wrap around. In the context of 2038, rollover refers to the integer overflow of 32-bit time values.

time_t
A POSIX-defined data type traditionally implemented as a signed 32-bit integer representing seconds since the epoch. The reliance on time_t is the primary source of the Year 2038 Problem.

Unsafe Function
Any programming construct or API call that is not resilient to 2038 rollover. Examples include gmtime, ctime, localtime, and functions that rely on 32-bit time_t.

Safe Function
Any programming construct or API call that is resilient beyond 2038. Examples include Java's java.time API, C++'s chrono::system_clock, Python's datetime64, and 64-bit implementations of time_t.

Windowing
A short-term remediation strategy that assumes a specific date window when interpreting truncated or limited date formats. While used during Y2K, windowing is discouraged for 2038 because it defers rather than resolves the underlying risk.

Future-Proofing
Designing systems so they remain resilient not only through 2038 but far into the future, ideally until at least 2100. Futureproofing requires structural solutions rather than temporary workarounds.

Outcome
A measurable result that demonstrates compliance with the EPOCH+ framework. For example: “System APIs validate input time values against dates beyond 2038.”

Profile
A tailored application of the framework for a specific industry, sector, or organisation. For example, a “Healthcare Profile” may emphasise embedded device testing, while a “Financial Profile” may prioritise database schema audits.

Certification
Formal recognition by the Legacy Systems Foundation that an individual, system, or organisation has achieved compliance with the EPOCH+ 2038 Safe™ Standard.

Time Resilience Officer (TRO)
The designated role within an organisation responsible for overseeing compliance with this standard. Equivalent to roles such as Chief Information Security Officer (CISO) in security standards.

Digital Continuity
The assurance that digital information remains usable, accessible, and trustworthy over time. Digital continuity requires that timekeeping functions remain accurate across system lifecycles.

Technical Debt
Design compromises made in the short term that create fragility in the long term. Continuing to use unsafe time constructs constitutes technical debt.

Resilience Engineering
A discipline focused on designing systems that can anticipate, absorb, and recover from unexpected disruptions. Time resilience is a specific application of resilience engineering.

Software Bill of Materials (SBOM)
An inventory of software components, libraries, and dependencies used in a system. SBOMs are essential for detecting unsafe libraries vulnerable to the 2038 problem.

Test Suite
A structured collection of test cases designed to validate specific outcomes. In this standard, a Year 2038 test suite refers to automated tests simulating rollover and post-2038 dates.

Regression Testing
Testing performed after system modifications to ensure that new changes have not reintroduced vulnerabilities. Regression testing for 2038 resilience should include rollover scenarios.

Certification Badge
A digital or physical symbol indicating successful certification against this standard. Badges may be issued to individuals (e.g., “2038 Safe Developer”) or organisations.

4. Principles

The EPOCH+ Standard is grounded in a set of **guiding principles**. These principles are **mandatory foundations**: they shape how organisations interpret, implement, and sustain their compliance posture. Each requirement in this framework is derived from, and must be consistent with, these principles.

4.1 Accuracy First

Timekeeping **shall** be correct before any other consideration. Performance, convenience, or backward compatibility must never override accuracy.

Accuracy applies to both:

- **Representation**: bit-level correctness in storage and processing.
- **Semantics**: correct handling of leap seconds, daylight savings, time zones, and forward projections.

Example: A financial institution may face legal liability if interest calculations are based on incorrect dates. “Close enough” is not acceptable when safety, contractual obligations, or financial stability are at stake.

4.2 Future-Proof by Design

Temporary workarounds create technical debt and guarantee future crises. All remediation actions **shall** target resilience until at least the year 2100.

- “Windowing” or schema patches that defer risk are prohibited.
- Systems **shall** adopt scalable representations such as 64-bit integers, ISO 8601 formats, or language-native date-time classes.

Example: Migrating a database field from 32-bit to 64-bit integers may involve short-term overhead, but it eliminates decades of risk compared to a stopgap solution.

4.3 Assumption Minimisation

Time handling **shall** minimise assumptions about calendars, time zones, and platform behaviour. Unsafe assumptions include:

- That all days have 24 hours.
- That daylight savings or leap seconds are irrelevant.
- That time_t is interpreted identically across platforms.

Where assumptions are unavoidable, they **shall** be explicitly documented.

Example: A cloud provider documenting its exclusive use of UTC for all internal storage avoids ambiguity across customers and regions.

4.4 Defence in Depth

Resilience **shall** be layered. If one component fails (e.g., a third-party library mishandles rollover), surrounding components must detect, contain, and prevent error propagation.

This principle applies across software, governance, testing, and incident response.

Example: An API gateway rejecting malformed or pre-1970 timestamps before they reach downstream systems provides containment against external errors.

4.5 Auditability and Transparency

Time resilience is only credible when it can be demonstrated. Organisations **shall** maintain auditable records of their compliance journey, including:

- Test results.
- Architectural diagrams.
- SBOMs documenting removal of unsafe dependencies.

Transparency fosters trust with regulators, customers, and partners.

4.6 Shared Responsibility

Time resilience is a collective responsibility across the organisation:

- Developers **shall** adopt safe coding practices.
- Procurement **shall** verify vendor compliance.
- Auditors **shall** validate evidence.
- Executives **shall** allocate resources and champion resilience.
- Operations **shall** maintain ongoing vigilance.

No single role or department is sufficient; resilience requires shared accountability.

4.7 Continuous Improvement

The Year 2038 Problem is one milestone in a broader challenge of **digital continuity**. Organisations **shall**:

- Monitor for emerging risks (e.g., Year 2106 rollover for unsigned 32-bit counters).
- Integrate lessons from testing and incidents into improvement cycles.
- Ensure time resilience evolves as systems, threats, and standards change.

5. Framework Categories

The EPOCH+ Standard is organised into **eight categories**. Each category contains **outcomes**: measurable results that organisations **shall** achieve.

The categories are sector-neutral but may be adapted into **profiles** (e.g., finance, healthcare, cloud) as described in Annex B.

5.1 Data Representation

Outcome 5.1.1: Use of Safe Data Types

All systems **shall** store and process time values using representations capable of handling years beyond 2038.

- Acceptable formats include 64-bit integers, ISO 8601 strings, or language-native date-time classes (e.g., *java.time*).
- 32-bit signed integers are prohibited.
  Rationale: Migration to safe types eliminates rollover risk and ensures long-term consistency.

Outcome 5.1.2: Canonical Exchange Format

Inter-system communication **shall** use ISO 8601 / RFC 3339 formats.
Rationale: Prevents failures caused by inconsistent local formats (e.g., *MM/DD/YY* vs *DD/MM/YY*).

Outcome 5.1.3: Explicit Time Zones

All systems **shall** explicitly state time zones or store in UTC. Ambiguity caused by daylight savings or offsets is prohibited.

Outcome 5.1.4: Documentation of Epoch Choices

If non-standard epochs are used (e.g., GPS 1980, custom business epochs), they **shall** be documented with clear conversion rules.

5.2 Code & Function Safety

Outcome 5.2.1: Elimination of Unsafe APIs

Functions dependent on 32-bit time_t (e.g., *ctime, localtime*) **shall** be replaced. Unsafe functions **shall** be prohibited by coding standards.

Outcome 5.2.2: Adoption of Safe Libraries

Teams **shall** migrate to modern date-time APIs, such as Java's *java.time*, .NET *DateTimeOffset*, or Python *datetime64*.

Outcome 5.2.3: Language-Specific Standards

Each programming language in use **shall** have documented 2038-safe coding standards.
Example: C++ teams adopting *chrono*.

Outcome 5.2.4: Secure Input Validation

Applications **shall** validate input dates to detect rollover corruption or malformed values.

5.3 System Architecture

Outcome 5.3.1: Centralised Time Services

Architectures **should** rely on centralised, hardened time services (e.g., NTP, GPS, cloud APIs) with resilience features.

Outcome 5.3.2: Redundancy

Critical time services **shall** be redundant, using at least two independent trusted sources.

Outcome 5.3.3: Forward Date Simulation

Architectures **shall** support simulation of post-2038 dates for validation and testing without requiring real-time passage.

Outcome 5.3.4: Embedded Device Strategy

For devices that cannot be upgraded (e.g., industrial controllers), organisations **shall** plan mitigation strategies such as encapsulation, replacement, or external validation.

5.4 Testing & Validation

Outcome 5.4.1: Automated Rollover Tests

CI/CD pipelines **shall** include automated tests simulating the 2038 rollover.

Outcome 5.4.2: Regression Testing

Every system modification **shall** re-run rollover tests to prevent reintroduction of vulnerabilities.

Outcome 5.4.3: End-to-End Scenario Testing

Cross-system workflows (e.g., booking flights for 2040, issuing 30-year loans) **shall** be validated beyond 2038.

Outcome 5.4.4: Independent Validation

Compliance **should** be confirmed through third-party audits or accredited certification labs.

5.5 Documentation & Governance

Outcome 5.5.1: Time Resilience Policy

Organisations **shall** publish a policy defining their approach to 2038 resilience.

Outcome 5.5.2: Traceable Records

All remediation activities **shall** generate traceable records (e.g., logs, change tickets, audit evidence).

Outcome 5.5.3: Governance Mapping

Time resilience **shall** be integrated into existing risk frameworks (e.g., ISO 27001 Annex A, SOC 2).

5.6 Supply Chain & Vendor Assurance

Outcome 5.6.1: Vendor Declarations

Suppliers **shall** declare whether their products are compliant with 2038-safe practices.

Outcome 5.6.2: Contractual Obligations

Procurement contracts **shall** include clauses requiring vendors to maintain compliance through 2038 and beyond.

Outcome 5.6.3: SBOM Analysis

Organisations **shall** generate and monitor SBOMs to detect unsafe libraries and dependencies.

5.7 Incident Response & Continuity

Outcome 5.7.1: Playbooks

Incident response teams **shall** maintain playbooks specifically addressing time-related failures.

Outcome 5.7.2: Continuity of Operations

Critical systems **shall** be designed to degrade gracefully in the event of time errors (e.g., failover to manual modes).

Outcome 5.7.3: Lessons Learned Integration

Post-incident reviews **shall** feed directly into continuous improvement programmes.

5.8 Workforce Training & Awareness

Outcome 5.8.1: Developer Training

All developers **shall** complete training on 2038-safe coding practices.

Outcome 5.8.2: Executive Awareness

Executives **shall** be briefed on 2038 business risks, funding needs, and compliance obligations.

Outcome 5.8.3: Cross-Functional Drills

Organisations **should** conduct tabletop exercises simulating a 2038-related outage.

6. Certification and Audit

The EPOCH+ Standard is not only a set of guidelines but a **certifiable framework**. Certification provides independent assurance that organisations and individuals have implemented, tested, and documented their compliance with time resilience practices.

Certification is designed to be:

- **Trustworthy**: delivering assurance to regulators, partners, and customers.
- **Practical**: aligned with existing compliance programmes.
- **Sustainable**: requiring continuous improvement and surveillance.

6.1 Purpose of Certification

Certification serves three primary purposes:

1. **Trust and Assurance**: Stakeholders **shall** have confidence that certified systems and organisations are resilient to the Year 2038 Problem and beyond.
2. **Market Differentiation**: Certified organisations **may** demonstrate leadership and competitive advantage through adoption of a globally recognised benchmark.
3. **Continuous Improvement**: Certification **shall** establish a cycle of review, remediation, and re-certification, ensuring time resilience is an ongoing discipline, not a one-time project.

6.2 Levels of Certification

Certification is structured into tiers to reflect increasing maturity.

- **Foundation Level (Organisations)**
  Demonstrates that baseline policies, governance structures, and awareness exist. Comparable to ISO/IEC 27001 readiness assessments.
- **Intermediate Level (Systems)**
  Confirms that specific systems have been audited for safe data representation, coding practices, and architectural resilience. Includes validated test results.
- **Advanced Level (Enterprise-Wide)**
  Signifies full organisational integration of the framework. Covers vendor management, continuous improvement, and independent third-party validation.
- **Professional Certifications (Individuals)**
  - *2038 Safe Developer*: awarded to developers trained in safe coding practices.
  - *2038 Safe Auditor*: awarded to auditors capable of assessing compliance.

- *2038 Safe Architect*: awarded to system designers and architects embedding resilience into infrastructure.

Requirement: Each certificate **shall** clearly state its scope (e.g., single system, enterprise-wide, or professional individual). Certificates **shall not** be represented as applying beyond their stated scope.

6.3 Certification Process

The certification lifecycle consists of five stages:

1. **Preparation**: Organisations **shall** identify scope, appoint a Time Resilience Officer (TRO), and map framework outcomes to their environment.
2. **Self-Assessment**: Internal audits, code scans, and test suites **shall** be performed. Evidence **shall** be collected against outcomes defined in Section 5.
3. **Independent Audit**: Accredited auditors **shall** review documentation, test results, and conduct stakeholder interviews. Spot testing **may** be performed.
4. **Decision and Award**: Certification **shall** only be awarded upon full evidence review. Certificates are valid for a maximum of **three years**.
5. **Surveillance and Renewal**: Surveillance audits **shall** be conducted annually. Renewal **shall** occur at or before the end of the three-year validity period.

Requirement: Certified organisations **shall transition** to new revisions of the EPOCH+ Standard within 18 months of publication.

6.4 Audit Methodologies

Audits against this standard adopt a multi-layered approach:

- **Document Review**: Examination of policies, coding standards, SBOMs, and governance records.
- **Technical Validation**: Execution of rollover simulations, CI/CD test results, and penetration tests for unsafe APIs.
- **Interviews**: Engagement with developers, executives, and operations teams.
- **Site Inspections**: Required for industries with embedded or safety-critical systems (e.g., industrial controls, healthcare devices).

Requirement: Auditors **shall** maintain independence and objectivity. The Legacy Systems Foundation **shall** operate an Auditor Accreditation Programme, renewed every three years, to ensure consistent application.

6.5 Evidence Requirements

The following evidence types are **mandatory** for certification at the system or enterprise level:

- **Policy Statements**: Board-approved policies on time resilience.
- **Technical Records**: Source code repositories, audit logs, and SBOMs.
- **Test Evidence**: Automated test reports proving rollover resilience.
- **Incident Records**: Evidence of tabletop exercises or incident response activity.
- **Training Records**: Proof of developer and workforce training on 2038 resilience.

6.6 Surveillance and Non-Conformance

Certification is not permanent.

- **Surveillance Audits**: Annual surveillance **shall** verify continued compliance.
- **Non-Conformance Classification**:
    - *Minor*: Corrective action **shall** be completed before the next surveillance audit.
    - *Major*: Corrective action **shall** be completed within **90 days**, or certification will be suspended or revoked.
- **Corrective Action Plans (CAPs)**: Organisations **shall** submit CAPs for all non-conformances.

6.7 Integration with Other Audits

To minimise duplication, this standard is designed for integration with existing compliance audits.

Evidence from related frameworks (e.g., ISO/IEC 27001, SOC 2 Type II, NIST CSF, PCI DSS, IEC 62443) **may** be reused where equivalent controls exist.

7. Annexes

The annexes provide **non-normative guidance**. They are useful but not mandatory, designed to support implementation, auditing, and workforce development.

Annex A: Mapping to Other Standards

This annex shows how the EPOCH+ Standard aligns with existing frameworks. Organisations may use this mapping to demonstrate integration and avoid duplicating effort.

ISO/IEC 27001

- **Annex A.12.1.2 (Change Management)** → Outcome 5.4.2 (Regression Testing).
    - **Demonstrated By:** Evidence that regression tests are re-run after each system modification.
- **Annex A.18.2 (Compliance)** → Outcome 5.5.3 (Governance Mapping).
    - **Demonstrated By:** Policies showing time resilience integrated into compliance reporting.

NIST Cybersecurity Framework (CSF)

- **Identify** → Outcome 5.1.4 (Documenting Epoch Choices).
    - **Demonstrated By:** System documentation clearly defining epoch usage and conversion rules.
- **Protect** → Outcome 5.2.1 (Elimination of Unsafe APIs).
    - **Demonstrated By:** Code scans proving unsafe APIs removed.
- **Detect** → Outcome 5.4.1 (Automated Rollover Tests).
    - **Demonstrated By:** CI/CD test reports.
- **Respond** → Outcome 5.7.1 (Incident Playbooks).
    - **Demonstrated By:** Published incident response runbooks.
- **Recover** → Outcome 5.7.2 (Continuity of Operations).
    - **Demonstrated By:** Business continuity plans with tested fallback modes.

ISO/IEC 25010

- **Reliability and Maintainability** → Section 4.2 (Future-Proof by Design).
    - **Demonstrated By:** Long-term resilience plans and system design documents.

Annex B: Sector Implementation Profiles

Sector profiles illustrate how EPOCH+ outcomes may be prioritised. Each includes a compliance expectation.

Finance

- **Priority:** Database schema migrations for long-term loans.
- **Risk:** Mispriced securities or failed settlements due to rollover errors.
- **Compliance Expectation:** Financial institutions **shall** prove that contract, loan, and trading systems accept and process post-2038 dates.

Healthcare

- **Priority:** Validation of embedded devices and alignment with FDA/EMA guidance.
- **Risk:** Patient safety if devices miscalculate schedules.
- **Compliance Expectation:** Healthcare providers **shall** validate that medical devices, health records, and scheduling systems are time-resilient beyond 2038.

Industrial Control Systems (ICS)

- **Priority:** Replacement strategies for non-upgradeable PLCs and SCADA controllers.
- **Risk:** Power grid instability or utility shutdowns.
- **Compliance Expectation:** Operators **shall** implement mitigation (replacement, encapsulation, or monitoring) for non-compliant embedded devices.

Cloud Services

- **Priority:** Standardisation of APIs to ISO 8601 / RFC 3339.
- **Risk:** Multi-tenant failures if cloud time APIs are inconsistent.
- **Compliance Expectation:** Providers **shall** attest to time resilience of all shared services and ensure tenant isolation from unsafe libraries.

Annex C: Sample Test Cases

Test cases are grouped into categories to support structured validation.

Unit Tests

- **Rollover Simulation:** Verify that the system accepts 2040-01-01T00:00:00Z without error.
- **Negative Timestamp Rejection:** Ensure invalid values (e.g., 1901-12-13) are rejected.

Integration Tests

- **Cross-System Workflow:** Validate a transaction that spans pre- and post-2038 dates.
- **Database Compatibility:** Confirm queries and indexes accept post-2038 dates.

Stress Tests

- **High Load Rollover Simulation:** Execute rollover scenarios under peak system load.
- **Batch Job Forward Dates:** Validate that long-running jobs scheduled beyond 2038 execute correctly.

Security Tests

- **Fuzz Testing:** Submit malformed date inputs (e.g., 9999-99-99) to verify robust handling.
- **Dependency Scanning:** Confirm no unsafe libraries remain in the SBOM.

Annex D: Training Curriculum

The following curriculum provides a structured pathway for developing professional competency. Each module includes a learning outcome.

Module 1: Understanding the Year 2038 Problem

- *Outcome:* Learners **shall** describe the technical cause and business impact of the Year 2038 rollover.

Module 2: Unsafe vs Safe APIs in Major Languages

- *Outcome:* Learners **shall** identify unsafe APIs and replace them with compliant alternatives.

Module 3: Database Migration Patterns

- *Outcome:* Learners **shall** apply safe schema migration strategies to replace 32-bit date fields.

Module 4: Rollover Testing in CI/CD

- *Outcome:* Learners **shall** integrate automated rollover tests into CI/CD pipelines.

Module 5: Governance and Audit Evidence

- *Outcome:* Learners **shall** prepare documentation suitable for audit under the EPOCH+ Standard.

Module 6: Sector-Specific Risks (Finance, Healthcare, ICS)

- *Outcome:* Learners **shall** assess sector-specific vulnerabilities and propose compliant mitigation strategies.