VIOE - Vulnerability Intelligence & Orchestration Engine

Maintenance & Upgrade Strategy

Document Version: 1.0 **Classification:** Internal - Operations **Last Updated:** January 2026

Table of Contents

1. Introduction

1.1 Purpose

This document defines VIOE's approach to ongoing maintenance, updates, and long-term system health. Following these strategies ensures system reliability, security, and optimal performance.

1.2 Maintenance Philosophy

Principle	Description
Proactive	Address issues before they impact users
Scheduled	Regular, predictable maintenance windows
Documented	All changes tracked and recorded
Tested	Changes validated before production
Reversible	Ability to rollback any change

1.3 Stakeholder Responsibilities

Role	Responsibility
Operations Team	Execute maintenance procedures
Product Owner	Approve upgrade schedules
Development Team	Provide updates and fixes
Security Team	Review security updates
Users	Report issues promptly

---

2. Update Cadence

2.1 Release Schedule

Release Type	Frequency	Contents
Major Version	Annual	New features, major changes
Minor Version	Quarterly	Enhancements, improvements
Patch Release	Monthly	Bug fixes, minor improvements
Security Update	As needed	Security patches

2.2 Update Calendar

Period	Activity
Week 1 of Month	Patch release deployment
Week 2 of Month	Performance review
Week 3 of Month	Security scan and updates
Week 4 of Month	Planning for next cycle

2.3 Maintenance Windows

Regular Maintenance:

Window	Time	Duration
Weekly	Sunday 2:00 AM - 4:00 AM	2 hours
Monthly	First Sunday 12:00 AM - 6:00 AM	6 hours
Quarterly	Planned weekend	8 hours

Communication:

- 7 days notice for planned maintenance
- 24 hours notice for urgent maintenance
- Immediate notice for emergency maintenance

2.4 Update Notification Template

```
SCHEDULED MAINTENANCE NOTIFICATION

Date: [Date]
Time: [Start Time] - [End Time] [Timezone]
Duration: [X hours]

WHAT'S HAPPENING:
[Description of maintenance activities]

IMPACT:
[Expected impact to users]

ACTION REQUIRED:
[Any user actions needed]
```

```
QUESTIONS:
Contact [support email/channel]
```

---

3. Backup Strategy

3.1 Backup Types

Backup Type	Frequency	Retention	Purpose
Continuous	Real-time	7 days	Point-in-time recovery
Daily Snapshot	Daily 2 AM	30 days	Daily recovery point
Weekly Full	Sunday 1 AM	90 days	Weekly recovery point
Monthly Archive	1st of month	1 year	Long-term retention
Annual Archive	January 1	7 years	Compliance/legal

3.2 Backup Components

Component	Method	Location
Database	Automated snapshot	Cloud backup service
File Storage	Cross-region replication	Secondary region
Configuration	Version control	Git repository
Secrets	Encrypted backup	Secure vault
Logs	Log aggregation	Log storage service

3.3 Backup Verification

Verification	Frequency	Procedure
Backup completion	Daily	Automated check
Backup integrity	Weekly	Checksum verification
Restore test	Monthly	Restore to test environment
Full recovery drill	Quarterly	Complete recovery simulation

3.4 Backup Monitoring

Alerts:

Condition	Alert Level
Backup failed	Critical
Backup delayed > 1 hour	High
Backup size anomaly	Medium
Retention policy violation	High

3.5 Recovery Procedures

Database Point-in-Time Recovery:

Step	Action	Time
1	Identify recovery timestamp	5 min
2	Stop application writes	2 min
3	Initiate recovery	1 min
4	Monitor recovery progress	Variable
5	Verify data integrity	10 min
6	Resume application	2 min

Full System Recovery:

Step	Action	Time
1	Assess damage and recovery needs	15 min
2	Provision replacement infrastructure	30 min
3	Restore database	30-60 min
4	Restore file storage	20 min
5	Deploy application	15 min
6	Verify functionality	20 min
7	Update DNS/routing	5 min

---

4. Data Migrations

4.1 Migration Types

Type	Description	Risk Level
Schema Change	Database structure modification	Medium-High
Data Transform	Modify existing data	Medium
Data Backfill	Populate new fields	Low-Medium
Data Cleanup	Remove obsolete data	Low

4.2 Migration Process

Planning Phase:

Step	Activity	Output
1	Define migration requirements	Specification document
2	Assess impact and risks	Risk assessment
3	Develop migration script	Tested script
4	Create rollback script	Rollback procedure
5	Define validation criteria	Test plan

Execution Phase:

Step	Action	Verification
1	Backup database	Backup confirmed
2	Execute in staging	Staging passes tests
3	Validate staging data	Data integrity verified
4	Execute in production	Production successful
5	Validate production	All tests pass
6	Document completion	Migration record updated

4.3 Migration Best Practices

Practice	Description
Backward compatible	New code works with old schema first
Incremental	Break large migrations into steps
Tested	Run in non-production first
Timed	Execute during low-traffic periods
Monitored	Watch for errors during execution

4.4 Migration Rollback Criteria

Trigger	Action
Data corruption detected	Immediate rollback
Application errors > 1%	Investigate, consider rollback
Performance degradation > 50%	Investigate, consider rollback
User-impacting bug	Assess and decide

---

5. Versioning Approach

5.1 Semantic Versioning

VIOE follows semantic versioning: **MAJOR.MINOR.PATCH**

Component	When Incremented
MAJOR	Breaking changes, major new features
MINOR	New features, backward compatible
PATCH	Bug fixes, security patches

Examples:

- 1.0.0 → 2.0.0: Major redesign, breaking API changes
- 1.0.0 → 1.1.0: New dashboard feature
- 1.0.0 → 1.0.1: Bug fix

5.2 Version History Tracking

Information Tracked	Purpose
Version number	Identification
Release date	Timeline
Changes included	Change log
Known issues	User awareness
Upgrade path	Migration guide

5.3 Compatibility Matrix

Component	Compatibility Window
API version	2 major versions
Database schema	Current + 1 previous
Integration protocols	Current + 1 previous
Browser support	Latest 2 versions

5.4 Deprecation Policy

Phase	Timeline	Action
Announced	T	Feature marked deprecated
Warning	T + 6 months	Usage warnings displayed
Disabled	T + 12 months	Feature disabled
Removed	T + 18 months	Code removed

Deprecation Communication:

- Release notes announcement
- In-app warnings
- Direct notification to known users
- Documentation updates

---

6. Maintenance Procedures

6.1 Routine Maintenance Tasks

Daily Tasks:

Task	Time	Owner
Check system health	9:00 AM	Operations
Review overnight alerts	9:00 AM	Operations
Verify backup completion	9:00 AM	Operations
Check integration status	9:00 AM	Operations

Weekly Tasks:

Task	Day	Owner
Review performance metrics	Monday	Operations
Check disk usage	Tuesday	Operations
Review security alerts	Wednesday	Security
Validate backups	Thursday	Operations
Update documentation	Friday	Operations

Monthly Tasks:

Task	Week	Owner
Apply patch updates	Week 1	Operations
Review access logs	Week 2	Security
Performance optimization	Week 3	Operations
Capacity planning review	Week 4	Operations

6.2 Database Maintenance

Task	Frequency	Purpose
Index optimization	Weekly	Query performance
Statistics update	Daily	Query optimization
Vacuum/cleanup	Weekly	Reclaim space
Connection pool check	Daily	Resource management

6.3 Log Maintenance

Task	Frequency	Purpose
Log rotation	Daily	Manage storage
Log archival	Weekly	Long-term storage
Log analysis	Weekly	Trend identification
Old log cleanup	Monthly	Storage optimization

6.4 Security Maintenance

Task	Frequency	Purpose
Vulnerability scan	Weekly	Identify vulnerabilities
Dependency audit	Weekly	Check for CVEs
Certificate check	Monthly	Prevent expiry
Access review	Monthly	Validate permissions
Penetration test	Annually	Deep security assessment

---

7. Upgrade Planning

7.1 Upgrade Assessment

Before Any Upgrade:

Assessment	Questions
Compatibility	Will existing data/integrations work?
Impact	What will change for users?
Risk	What could go wrong?
Rollback	Can we revert if needed?
Testing	How will we validate?

7.2 Upgrade Checklist

Pre-Upgrade:

```
□ Review release notes
□ Check compatibility requirements
□ Verify backup is current
□ Test upgrade in staging
□ Prepare rollback plan
□ Schedule maintenance window
□ Notify stakeholders
□ Document current configuration
```

During Upgrade:

```
□ Put system in maintenance mode
□ Take final backup
□ Execute upgrade
□ Run smoke tests
□ Verify data integrity
□ Check integrations
□ Monitor error rates
```

Post-Upgrade:

```
□ Remove maintenance mode
□ Monitor system health
□ Verify all features working
□ Check performance metrics
□ Review user feedback
```

```
□ Document any issues
□ Update internal documentation
```

7.3 Major Version Upgrade Process

Phase	Duration	Activities
Planning	2-4 weeks	Assessment, scheduling, communication
Preparation	1-2 weeks	Staging test, team training
Execution	1 day	Upgrade execution
Stabilization	1-2 weeks	Monitoring, issue resolution
Closure	1 week	Documentation, lessons learned

7.4 User Communication

Pre-Upgrade (2 weeks):

- Announce upgrade date
- Share release notes
- Highlight key changes
- Provide training resources

Post-Upgrade:

- Confirm completion
- Share known issues
- Provide support contacts
- Gather feedback

---

8. Dependency Management

8.1 Dependency Categories

Category	Examples	Update Approach
Framework	React, Node.js	Scheduled, tested
Libraries	UI components, utilities	Regular updates
Infrastructure	Database, cache	Conservative, tested
Security	Auth, encryption	Priority updates

8.2 Dependency Update Schedule

Priority	Timing	Criteria
Critical Security	Immediate	CVE with known exploit
High Security	Within 7 days	CVE with high score
Regular	Monthly	Non-security updates
Major	Quarterly	Major version updates

8.3 Dependency Review Process

Step	Action	Owner
1	Scan for updates	Automated
2	Assess security impact	Security
3	Review breaking changes	Development
4	Test in development	Development
5	Deploy to staging	Operations
6	Validate functionality	QA
7	Deploy to production	Operations

8.4 Dependency Documentation

Track for Each Dependency:

Field	Purpose
Name	Identification
Version	Current version
License	Legal compliance
Purpose	Why we use it
Last updated	Maintenance tracking
Known issues	Risk awareness

---

9. Performance Maintenance

9.1 Performance Monitoring

Metric	Target	Alert Threshold
Page load time	< 2 seconds	> 3 seconds
API response time	< 500ms	> 1 second
Database query time	< 100ms	> 500ms
Error rate	< 0.1%	> 1%

9.2 Performance Optimization Schedule

Activity	Frequency	Owner
Performance review	Weekly	Operations
Slow query analysis	Weekly	DBA
Cache optimization	Monthly	Operations
Load testing	Quarterly	QA

9.3 Common Performance Issues

Issue	Indicator	Resolution
Slow queries	High query time	Optimize indexes
Memory pressure	High memory usage	Scale or optimize
Connection limits	Connection errors	Increase pool size
CPU saturation	High CPU usage	Scale horizontally

9.4 Capacity Planning

Review	Frequency	Focus
Current utilization	Monthly	Resource usage
Growth trends	Quarterly	Projection
Capacity forecast	Annually	Planning

---

10. Long-Term Health Monitoring

10.1 System Health Metrics

Metric	Good	Warning	Critical
Uptime	> 99.9%	< 99.9%	< 99%
Error rate	< 0.1%	< 1%	> 1%
Response time	< 500ms	< 1s	> 2s
User satisfaction	> 90%	< 90%	< 70%

10.2 Health Check Schedule

Check	Frequency	Method
Automated health	Continuous	Monitoring tools
Manual review	Weekly	Operations review
Deep assessment	Monthly	Comprehensive audit
External audit	Annually	Third-party assessment

10.3 Technical Debt Management

Activity	Frequency	Owner
Debt identification	Ongoing	Development
Debt assessment	Quarterly	Tech Lead
Debt prioritization	Quarterly	Product + Tech
Debt reduction	Ongoing	Development

Technical Debt Categories:

Category	Examples
Code quality	Complexity, duplication
Architecture	Outdated patterns
Dependencies	Old versions
Documentation	Incomplete/outdated
Testing	Low coverage

10.4 Continuous Improvement

Activity	Frequency	Output
Incident reviews	After each incident	Improvements
User feedback	Ongoing	Feature backlog
Performance analysis	Monthly	Optimization tasks
Security assessment	Quarterly	Security tasks
Process review	Quarterly	Process improvements

10.5 Annual Review

Area	Review Focus
Architecture	Is it still appropriate?
Technology stack	Are updates needed?
Performance	Meeting requirements?
Security	Gaps identified?
Compliance	Requirements met?
Capacity	Scaling needs?

Document Control

Version	Date	Author	Changes
1.0	January 2026	Documentation Team	Initial release

This guide ensures long-term system health and operational excellence.