# Statement of Work: Short-Term Security Improvements for Open VSX

## Objective

The primary goal of this engagement is to improve the security posture of the Eclipse Foundation–hosted Open VSX service. We have observed malicious or fake extensions, impersonation attempts, and accidental leaks of secrets. The contractor will design and implement pre-publish security checks inspired by best practices from other marketplaces (e.g. Microsoft VS Code Marketplace protections) and tailored to the specifics of Open VSX.

## Background

Eclipse Open VSX is an open source registry for Visual Studio Code extensions and compatible editors. The Eclipse Foundation operates the public instance at open-vsx.org, which is widely used by developers, IDE vendors, and tool providers as a trusted alternative to the Microsoft Marketplace.

The hosted service requires continuous improvements in security, reliability, performance, user experience, and integration with Eclipse Foundation infrastructure. Short-term improvements are aimed at addressing security risks and reducing operational friction.

## Scope (ordered by priority)

1. **Pre-publish security checks**: implement automated controls to detect malicious code, impersonation attempts, and accidental inclusion of secrets.
    a. **Workflow rules:**
        i. Publication must occur only if all checks pass.
        ii. If a check fails, the extension is quarantined, and an alert is triggered for manual review (alerting system and review process to be defined).
        iii. The system should be designed to be extensible, so new checks can be added over time.
    b. **Checks to implement:**
        i. **Malware detection**: Detection rules will be provided at a later stage and will involve executing external tools on the submitted extensions. These tools and procedures will remain undisclosed to prevent attackers from reverse-engineering the verification process.
        ii. **Name squatting**: Leverage well-documented distance algorithms to detect name squatting attempts, both at the namespace and extension name levels.
        iii. **Block list**: Support the incremental development of a smart blocklist based on previously flagged and removed malicious extensions (e.g., file SHAs, etc.)

    iv. **Secret scanning**: Detect well-known patterns of API keys and credentials (e.g., GitHub PATs, Open VSX tokens, etc.).
    v. **Scanning of binaries:** Detect and flag unexpected or suspicious binaries within submitted extensions to reduce the risk of hidden malware.
2. **Download flood control:** introduce a mechanism to prevent users from artificially inflating extension popularity by generating large numbers of fake or automated downloads. The system should ensure download metrics remain a reliable signal of genuine user adoption, while not preventing legitimate downloads.
3. **Documentation & transition**: provide clear documentation and recommendations to ensure maintainability.
    a. Developer documentation for adding pre-publish checks.
    b. Admin runbook for reviewing quarantined extensions.
    c. Knowledge transfer session with Foundation staff.
    d. Handoff checklist for delivered code, docs, and configs.
    e. Recommendations list for long-term improvements.

## Deliverables

- Quarantine workflow with basic admin alerting/notification.
- Pre-publish security check system integrated into the publishing workflow.
- Download flood control mechanism ensuring download metrics cannot be artificially inflated.
- Updated automation and CI/CD safeguards.
- Documentation (developer and admin guides, runbooks).
- Knowledge transfer and recommendations for future improvements.

All Deliverables must be contributed to the Eclipse Open VSX project and related resources under the project's license. If the contractor is not currently a Committer on the project, then working closely with an existing Committer to ensure the Deliverables are accepted is considered part of the work effort.

All work must be done in conformance with the Eclipse Development Process. All issues are to be tracked either in public issues, or where directed by Eclipse, to be tracked using confidential issues and resources.

## Skills Required

- Strong proficiency in **Java (JDK 17+)** and **Spring Boot 3.x** for backend services (core of Open VSX server).
- Solid experience with **TypeScript** and **React** for the web UI and publisher flows
- **PostgreSQL**, **Elasticsearch** for persistence and search, and (nice to have) **Redis** (used for rate-limiting/caching).
- Practical knowledge of **Docker** and **Kubernetes** for deployments and scaling.
- Experience with **CI/CD pipelines** (GitHub Actions, Jenkins).
- Strong background in **security best practices**: authentication, publishing workflows, dependency management, vulnerability scanning.

- Understanding of **monitoring/observability** tools and operational resilience
- Comfort working in **open source** contribution workflows under the Eclipse Foundation process.
- Experience implementing metrics-based fraud detection or abuse prevention mechanisms (e.g., flood control, throttling, anomaly detection).

## Possible Technical Approach

- Extend the publish pipeline to include asynchronous verification before completion.
- Use external scanning tools for malware and secrets, with results stored in a quarantine/verification table.
- Implement name-squatting checks using distance algorithms.
- Maintain a blocklist table for flagged extensions.
- Emit alerts/metrics to Prometheus, exposed via Grafana dashboards.
- Introduce a lightweight, configurable approach to flood control (e.g., thresholding, deduplication, or similar methods). Contractors are encouraged to propose alternative solutions.

These are possible approaches. Contractors are encouraged to propose improvements or alternatives that meet the requirements while ensuring scalability, security, and maintainability.

## Roles & Responsibilities

- **Development Team (Contractor)**: Implement security checks, quarantine workflow, download flood control, and admin tools; write tests and documentation.
- **DevOps/Infrastructure (EF)**: Provide infrastructure for scanning tools, monitoring, and CI/CD updates.
- **Quality Assurance (EF)**: Validate correctness of checks, quarantine behavior, flood control mechanism, and alerting.
- **Open VSX Admins (EF)**: Define blocklist entries, review quarantined extensions, manage alerts.
- **Project Manager**: Coordinate reviews, track progress, manage stakeholder communication.

## Assumptions & Dependencies

- Scanning tools and malware rules will be provided or selected collaboratively.
- The existing Open VSX publishing pipeline is available for extension.
- Download event logging is available and can be extended to support flood control.

## Reporting & Communication

- Weekly status reports including progress updates (burn-down charts or equivalent).
- Regular updates to issues on GitHub

- Stakeholder demos at key milestones to showcase dashboards and rate-limit functionality.

## Acceptance Criteria

- All three checks (malware, name squatting, secret scanning) are implemented and configurable.
- Extensions are published only after passing all checks.
- Failing checks result in quarantine with clear admin notification.
- Blocklist is active and applied to new submissions.
- Download flood control is in place and verified through tests to prevent artificial popularity inflation while allowing legitimate downloads.
- Documentation and handoff materials delivered.

## Proposal Evaluation Criteria

In evaluating proposals, the Eclipse Foundation will consider:

- Price and timeliness,
- Plan for proposed development and deliverables,
- Skillset and experience of proposed developers, with preference given to committers in the relevant area,
- Bidder's relationship with EF, with preference given to either Contributing Members with committers, or self-employed committers with relevant expertise,
- Any additional relevant elements in the bid, including delivery date, whether fixed price vs. time and materials basis, etc.

## Proposal Submission

Proposals must be submitted by email to software-dev@eclipse-foundation.org no later than 18 September 2025 at 18:00 CEST (UTC+2).

Submissions should include:

- A detailed plan addressing the scope, deliverables, and a timeline.
- Profiles of the proposed developers and their relevant experience.
- Pricing information.
- Any additional information the bidder believes is relevant for the proposal.

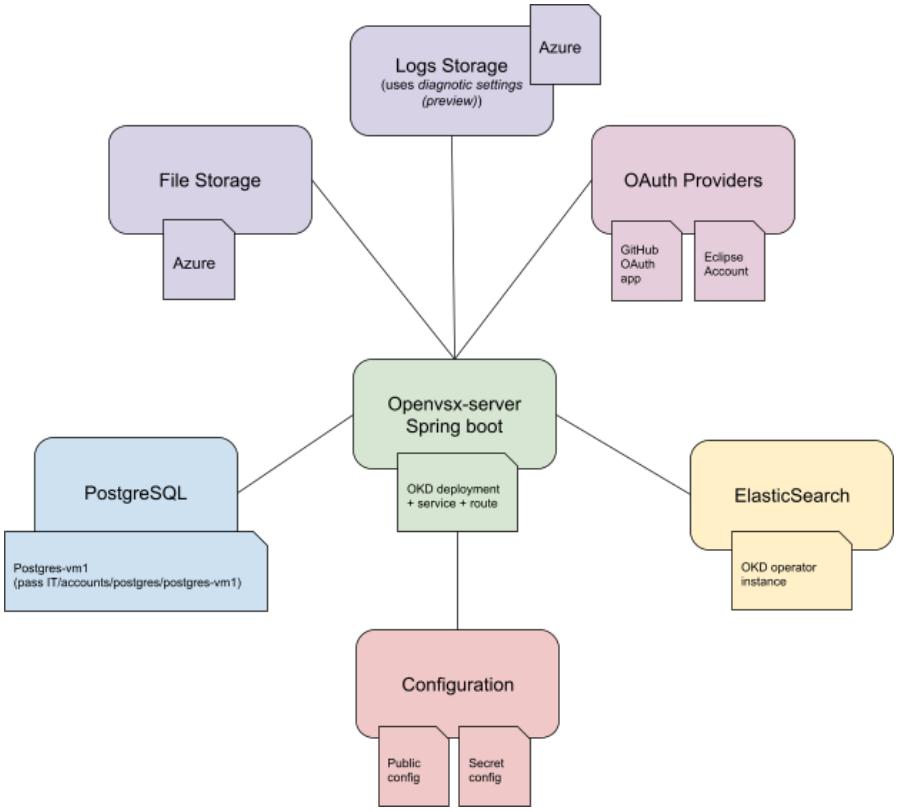
Late submissions may not be considered.

## Duration

Short-term engagement (**[X weeks/months]**, to be defined), focused exclusively on pre-publish security checks, with the goal of delivering an immediate improvement to Open VSX security posture.

# Architecture & projects

- Eclipse Open VSX project: https://github.com/eclipse/openvsx
- The source of open-vsx.org (the public instance of Eclipse Open VSX) is at: https://github.com/EclipseFdn/open-vsx.org
- Microsoft Code Marketplace – Extension runtime security practices
    - https://code.visualstudio.com/docs/configure/extensions/extension-runtime-security
    - https://code.visualstudio.com/docs/configure/extensions/extension-runtime-security#_marketplace-protections
- Reference documentation on the instance of Open VSX hosted by the Eclipse Foundation is available at: https://github.com/EclipseFdn/open-vsx.org/wiki/Deployment-Details




# References

1. GitHub Issue:
    a. Short-Term Security & Integrity Improvements