

# Permitted-snippet decision pack

Engineering security and tech leads

[Read the complete guide on aona.ai](#)

Developers should share employer-owned code only when the organisation permits that disclosure to the selected AI service and account. Start with the smallest example that still explains the problem. Remove unnecessary business logic, identifiers and configuration, and check what the client adds automatically. A provider's no-training policy does not replace the employer's permission.

Compare three synthetic code classes and capture an employer-owned disclosure decision.

Fictional code and illustrative classifications. No real employer approval or protection test is implied.

## Your review

Reviewer: \_\_\_\_\_ Date: \_\_\_\_\_

Organisation or team: \_\_\_\_\_

☐ **Record the employer's decision**

Name the code owner, permitted purpose, service and account.

☐ **Check that the smaller example still works**

Preserve the relevant error or behaviour without unrelated business context.

☐ **Review automatically added material**

Confirm selected files, repository context and tool results for the exact client action.

Checked items record your review, not a compliance approval or an installed-product test result.

## Notes and next actions

**Decision and scope:** \_\_\_\_\_

**Evidence to collect:** \_\_\_\_\_

**Owner and next review date:** \_\_\_\_\_

## Review matrix

| Code class                 | Needed for the question? | Next step                       |
|----------------------------|--------------------------|---------------------------------|
| Public-style utility       | Small relevant function  | Use an approved service/account |
| Internal business rule     | Possibly reducible       | Ask the code owner              |
| Restricted-labelled marker | Only for an access test  | Keep it synthetic and isolated  |

# Permitted-snippet training pack

Every example is fictional. The labels do not grant permission to share real employer or client code. No file contacts an API or reads your environment.

Compare the examples, then complete the “Employer-owned snippet decision” section. Do not replace the examples with restricted code. To discuss a real problem, first obtain the owner decision and prepare a minimal reproduction separately.

## Employer-owned snippet decision

Programming question: \_\_\_\_\_  
Code owner and permitted disclosure authority: \_\_\_\_\_  
Service, account and action: \_\_\_\_\_  
Minimum example needed: \_\_\_\_\_  
Unnecessary business context removed: \_\_\_\_\_  
Automatic context still to review: \_\_\_\_\_  
Decision: NOT YET REVIEWED  
Reviewer/date: \_\_\_\_\_

Illustrative example: an empty-list sum question can use public-style-example.py. The fictional contract-tier rule is not required to explain `sum([])`, so leave it out. This is an example of minimisation, not an employer approval.

## Practice files

The complete download pack includes these original working files. Keep their original formats when running or inspecting the examples.

- [Public style example \(public-style-example.py\)](#)
- [Fictional internal example \(fictional-internal-example.py\)](#)
- [Restricted marker \(restricted-marker.txt\)](#)

SYNTHETIC\_D02\_RESTRICTED\_CODE\_MARKER  
No actual proprietary algorithm or credential is present.

## Sources and scope

Sources checked 2026-09-21. Review the current requirements and your actual agreement before using this workbook for a real decision. Aona does not decide intellectual-property ownership or authorize disclosure, and no universal repository or coding-client coverage is implied.

[Cursor: Privacy and Data Governance](#). Explains transfer of prompts and code context to model providers.

[GitHub: Content exclusion for Copilot](#). Documents indirect semantic context and exclusion limitations.