Engineering agents on Katonic: the list

35 agents across nine areas, for the teams that build and maintain a company's own software

An engineering department's work runs through its source control, its issue tracker, its build and release system, its error and alerting tools, and its own standards, designs and runbooks. Every agent reaches them in one of four ways, all through the same tool gateway and the same approval rules:

	HOW EACH AGENT CONNECTS
01	**A file the user uploads** in the conversation: a diff, a test report, a build log, a design document, an export from the issue tracker.
02	**A knowledge source.** The coding standard, the review checklist, design documents, architecture decision records, runbooks and past incident reviews are indexed in the Knowledge Engine, with each reader's permissions kept.
03	**An imported API.** The company gives Katonic the description of an API it already runs, and the operations it chooses become tools. This is the route to systems of record: source control, the issue tracker, the build system, the package registry.
04	**An MCP server from the catalogue,** deployed inside the company's own cluster. This is the route to repositories, issues, on-call alerts, metrics and logs, and chat channels.

System names are left out on purpose. The company supplies the address and the sign-in. Nothing here is a tested integration with a named product.

THE NINE AREAS

01	Backlog and issues	2
02	Specifications and design	2
03	Code review	3
04	Tests	3
05	Documentation	4
06	Dependencies	4
07	Build and release	5
08	Incidents and reliability	5
09	Running the department	6

AREA 01

Backlog and issues

AGENT	WHAT IT DOES	READS FROM	WHERE IT STOPS FOR A PERSON
Issue Triage Agent	Reads a new issue, says what kind it is under the team's own labels, lists what is missing to reproduce it, and proposes the component and the owner the team's rules give	The issue through an MCP server or an imported API; the triage rules as a knowledge source	An engineer or the product owner confirms the labels and the owner. The agent assigns nothing and closes nothing
Duplicate Issue Check Agent	For a new issue, lists earlier issues that describe the same fault, with the passage in each that matches	The issue tracker through an MCP server or an imported API	An engineer decides whether it is a duplicate
Bug Report Check Agent	Checks a bug report against the team's template and lists what is missing: version, steps, expected and actual result, logs	The report; the template as a knowledge source	The reporter completes the report. The agent rejects nothing
Backlog Summary Agent	Summarises the backlog for a planning meeting: open items by component and age, items with no owner, and items that depend on each other	The issue tracker through an imported API or a reporting database	The product owner and the team decide the order

AREA 02

Specifications and design

AGENT	WHAT IT DOES	READS FROM	WHERE IT STOPS FOR A PERSON
Specification Draft Agent	Turns notes and a discussion into a draft specification in the team's template, and lists the open questions	The notes as an uploaded file; the template and related specifications as a knowledge source	The author and the product owner agree the specification
Design Review Preparation Agent	Reads a design document against the team's design checklist and earlier decisions, and lists what the document leaves out or departs from	The document as an uploaded file; the checklist and decision records as a knowledge source	The reviewers judge the design
Decision Record Draft Agent	Drafts an architecture decision record from a design discussion: the context, the options considered and the decision the team reached	The discussion as an uploaded file or a channel export; the record template	The team lead corrects and signs the record. The agent adds no decision the team did not take
Requirement Trace Check Agent	Lists the requirements in a specification that no test and no change refers to	The specification; the test list and the changes through an imported API	The engineering lead decides what is missing

AREA 03

Code review

AGENT	WHAT IT DOES	READS FROM	WHERE IT STOPS FOR A PERSON
Change Summary Agent	Summarises a proposed change for its reviewer: what it changes, in which files, and what the description does not mention	The diff as an uploaded file or through an MCP server	The reviewer reads the change itself and decides
Code Review Preparation Agent	Reads a proposed change against the team's own review checklist and coding standard and lists each point for the reviewer, with the line	The diff; the checklist and the standard as a knowledge source	An engineer reviews and approves or asks for changes. The agent approves nothing and merges nothing
Change Impact Note Agent	For a proposed change, lists what else it touches: other services that call the changed code, database migrations, configuration and feature flags	The diff and the repository through an MCP server; design documents as a knowledge source	The author and the reviewer decide what to test
New Dependency Check Agent	For a change that adds a package, checks that each new package exists in the registry the company uses, and reports its age, its publisher, its licence and known advisories	The diff; the registry and an advisory source through imported APIs	An engineer decides whether the package may be added

AREA 04

Tests

AGENT	WHAT IT DOES	READS FROM	WHERE IT STOPS FOR A PERSON
Test Case Draft Agent	Drafts test cases for a specification or a change, as a file for the engineer, and marks the cases it could not derive from the text	The specification or the diff as an uploaded file	The engineer reviews, edits and adds the tests through the normal change process
Test Failure Summary Agent	Reads a failed test run and groups the failures by likely cause, with the log lines behind each group and the changes that went in since the last passing run	The test report and the build log as files or through an imported API	The engineer on duty decides what to fix or revert
Unstable Test Report Agent	Lists tests that pass and fail on the same code, with how often and since when	Test run history through an imported API or a reporting database	The team decides which tests to fix or remove
Test Gap List Agent	Lists the files and functions a change touches that no test exercises, from the coverage report	The coverage report as a file; the diff	The reviewer decides whether more tests are needed

AREA 05

Documentation

AGENT	WHAT IT DOES	READS FROM	WHERE IT STOPS FOR A PERSON
Knowledge Agent	Answers from the company's own document stores with the passage cited, and keeps each reader's permissions	Document stores as a knowledge source	It informs. The reader checks the source
Code Documentation Draft Agent	Drafts the documentation of a module from its code and its tests, and marks every statement it could not confirm in the code	The repository through an MCP server, read operations only	The module's owner corrects and adds it through the normal change process
Interface Change Note Agent	Compares two versions of an interface description and lists what was added, removed and changed, and which changes break existing callers	The two versions as uploaded files	The interface owner decides the version number and the notice to callers
Runbook Draft Agent	Drafts or updates a runbook from past incident records and the engineer's notes, and marks each step it could not confirm	Past incidents, notes and the current runbook	The service owner tests and approves the runbook

AREA 06

Dependencies

AGENT	WHAT IT DOES	READS FROM	WHERE IT STOPS FOR A PERSON
Dependency Update List Agent	Lists the dependencies of a service that have a newer version, with the entries in each changelog that matter and the advisories that apply	The dependency file; registries and an advisory source through imported APIs	An engineer decides what to update and makes the change
Licence Check Agent	Lists the licence of each dependency against the company's licence policy and marks the ones the policy does not allow or does not cover	The dependency file; the policy as a knowledge source; the registry through an imported API	The engineering lead or the legal team decides
Advisory Impact Agent	For a new security advisory, lists the company's services that use the affected package and version	The advisory as a file or through an imported API; the dependency files through an MCP server	The service owners schedule the fix

AREA 07

Build and release

AGENT	WHAT IT DOES	READS FROM	WHERE IT STOPS FOR A PERSON
Build Failure Summary Agent	Reads a failed build and says where it failed, with the log lines and the change that went in before it	The build log as a file or through an imported API	The author of the change fixes or reverts it
Release Note Draft Agent	Drafts the release notes from the changes and the issues closed since the last release, grouped as the team's template asks	The changes and issues through an imported API; the template	The release manager edits and publishes
Production Change Record Draft Agent	Drafts the change record for a release to production: what changes, the tests run, the rollback plan and the approvals on file, and lists what the change policy asks for that is missing	The release contents, test results and approvals; the change policy as a knowledge source	The change manager approves the change. An engineer deploys
Release Readiness Check Agent	Checks a release against the team's own release checklist and lists each item with no evidence on file	The checklist as a knowledge source; test results, reviews and approvals	The release manager decides to release or hold

AREA 08

Incidents and reliability

AGENT	WHAT IT DOES	READS FROM	WHERE IT STOPS FOR A PERSON
Incident Response Agent	Reads an alert, pulls the related metrics and logs, and drafts an incident summary with the runbook step that fits	The on-call alert, metrics and logs through MCP servers; runbooks as a knowledge source	An engineer carries out any runbook step
Post-Incident Review Draft Agent	Drafts the review after an incident: what happened, the timeline, what worked, what did not, and proposed actions, each tied to its evidence	The incident record, the timeline, the changes and past reviews	The review owner and the team agree the causes and the actions. The agent names no person as the cause
Error Report Grouping Agent	Groups the errors reported from production in a period by likely cause and by the release that introduced them	Error reports through an MCP server or an imported API	The service owner decides what to fix first
Service Level Report Agent	Prepares the service level report for a period: availability and response times against the targets, and the incidents behind each miss	Metrics through an MCP server; the targets as a knowledge source	The service owner reviews and issues the report

AREA 09

Running the department

AGENT	WHAT IT DOES	READS FROM	WHERE IT STOPS FOR A PERSON
Technical Debt Register Agent	Gathers the items the team has marked as debt in issues, code comments and reviews into one list, with where each was raised	The issue tracker and the repository through MCP servers	The engineering lead decides what to schedule
Engineer Onboarding Guide Agent	Prepares a reading and set-up guide for a new engineer on a team from its documents, its repositories and its runbooks	Documents as a knowledge source; the repository list	The team lead corrects and gives the guide
Delivery Report Agent	Prepares the delivery report for a period by team and service: changes released, lead time, failed changes and time to restore, with the records behind each figure	Source control, the build system and incident records through imported APIs or a reporting database	The head of engineering reviews and issues the report. The report counts work by team and scores no engineer
Change Control Evidence Agent	For an audit of change control, gathers for each sampled change the review, the approval, the test result and the deployment record	Source control, the build system and the change records	The control owner checks the evidence and hands it to the auditor

	WHAT EVERY AGENT HAS IN COMMON
01	**It runs on the company's own infrastructure or private cloud,** with a model the company serves. Source code, issues and logs stay there.
02	**Every tool call passes the company's policy.** A policy can allow a tool, record its use, hold it for a person's approval, or block it, for the whole company or for one team.
03	**A tool that changes something waits for approval** unless the company has decided otherwise for that tool. That is how no merge, deployment, database change or published package happens on an agent's word.
04	**Read and change are separate tools.** When the company imports an API, it chooses which operations become tools. An agent that only needs to read is given the read operations and nothing else.
05	**Each statement shows its source.** An agent that checks or summarises gives the file and the line, the issue or the log, and says when it found nothing. It does not fill a gap from memory.
06	**The agent acts as the person using it.** A knowledge source keeps each reader's permissions, so a colleague does not see through an agent a repository or a document they could not open themselves.
07	**Every call is recorded,** with who asked, which tool, and what was decided, and an agent, a tool or a whole server can be switched off at once.

WHERE TO START

Pick work where reading is slow, the decision stays with an engineer, and nothing changes on the agent's word.

THREE PLACES THAT FIT

Knowledge Agent on the engineering documents

Engineers ask how a service works and get the answer with the passage cited. No repository is changed, so it can run on day one.

Test Failure Summary

The agent groups the failures of a test run by likely cause. The engineer on duty decides.

Code Review Preparation

The agent lists the points of the team's own checklist for the reviewer. An engineer reviews and approves.

One team, one repository, invented data first. Then beside the manual process, measured against it.

Each one prepares the work. A named engineer decides.

About Katonic AI

Katonic AI builds the platform that organisations use to build, run and govern AI agents on their own infrastructure: in their own data centre, in a private or sovereign cloud, or with no connection to the internet at all. The platform is in production with governments, banks, telecom operators and utilities. Katonic is certified to ISO 27001.

KATONIC.AI