



TraceBoard Suite User Manual

Version 1.0.10 | August 2026

Requirements Management · Compliance Documents · Test Execution

Generated on August 11, 2026

TraceBoard Suite — Professional Traceability Platform

Table of Contents

1 Introduction

- 1.1 What is the TraceBoard Suite?
- 1.2 Who Is It For?

2 Getting Started

- 2.1 System Requirements
- 2.2 Installation
- 2.3 Licensing
- 2.4 First Login

3 TraceBoard – Core Traceability

- 3.1 Projects and Sprints
- 3.2 Dashboard Overview
- 3.3 Customizable Dashboards
- 3.4 Items – Requirements, Tasks, Bugs, Tests
- 3.5 Relations and Traceability
- 3.6 Trace Views
- 3.7 Impact Analysis
- 3.8 Baselines
- 3.9 Document Diffing (Baseline Comparison of Generated Documents)
- 3.10 Import / Export
- 3.11 LLM Chat
- 3.12 Generate Test Cases
- 3.13 Cross-Module Consistency Check
- 3.14 Audit Narrative

4 Recommended Models for Audit Narratives

- 4.1 Important Notes
- 4.2 Import Assistant
- 4.3 Import Tasks from File

5 TraceDocs – Document Generation

- 5.1 Overview
- 5.2 Document Management
- 5.3 Generating Documents
- 5.4 Document Templates
- 5.5 Markdown Editor
- 5.6 Export Word
- 5.7 Bi-Directional Document-Item Linking
- 5.8 Corporate Branding

6 TraceTest – Test Execution and Coverage

- 6.1 Overview
- 6.2 Test Cases
- 6.3 Test Runs
- 6.4 Coverage Report
- 6.5 Evidence Upload
- 6.6 Coverage Trends
- 6.7 Test Report Export

7 Git Integration

- 7.1 Key Capabilities
- 7.2 Quick Start
- 7.3 Where Commits Appear

8 Unified SPA Shell & Product Switcher

- 8.1 The Unified SPA Shell
- 8.2 The Product Switcher
- 8.3 Real-Time Notifications (WebSocket)

9 Administration

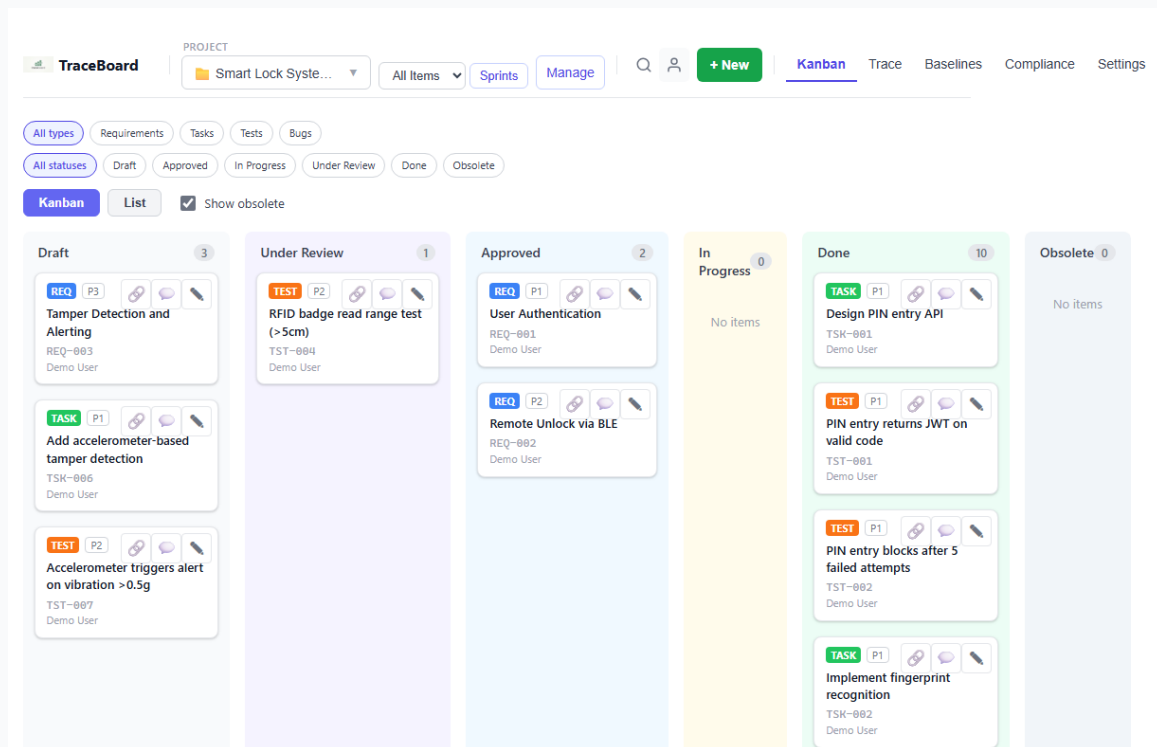
- 9.1 User Management
- 9.2 SSO (OIDC)
- 9.3 LLM Configuration
- 9.4 Backup and Restore
- 9.5 Onboarding Tour

10 Troubleshooting

- 10.1 License Not Found or Expired
- 10.2 TraceDocs or TraceTest Cannot Connect to TraceBoard
- 10.3 LLM (AI) Connection Issues
- 10.4 PDF Generation Fails
- 10.5 Cannot Switch Products

11 Glossary

12 Support



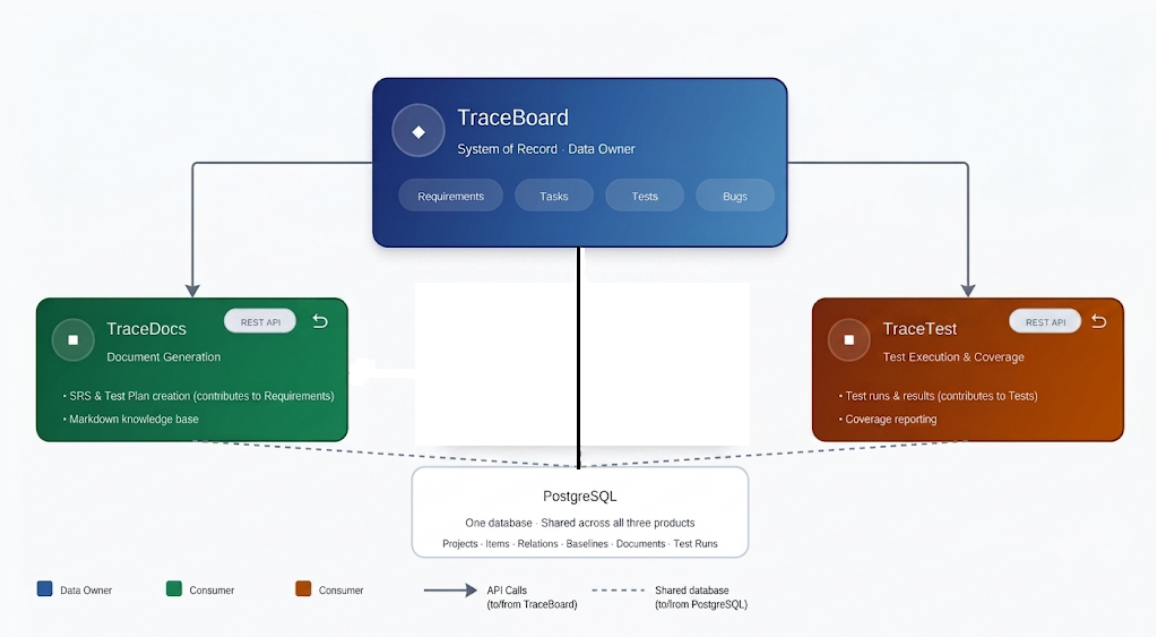
1 Introduction

1.1 What is the TraceBoard Suite?

The TraceBoard Suite is an integrated platform for requirements management, compliance document generation, and test execution tracking. It consists of three products that work together from a single source of truth:

Product	Purpose
TraceBoard	Requirements, baselines, traceability, and test planning. The core product — owns all domain data.
TraceDocs	Compliance document generation (SRS, Test Plan, Design Description, and User Manual) directly from TraceBoard baselines.
TraceTest	Test execution tracking, evidence management, and coverage reporting.

All three products share a common interface, a single PostgreSQL database, and the same authentication system. TraceBoard is the sole owner of projects, baselines, requirements, test cases, and traceability links. TraceDocs and TraceTest consume that data but never modify it — they store only their own operational data (document generations, test results, evidence).



1.2 Who Is It For?

- **Engineering teams** who need structured requirements management and traceability.
- **Compliance officers** who need audit-ready documentation (SRS, Test Plan, Design Description, and User Manual) with verified traceability.
- **Quality assurance teams** who need to track test execution and measure coverage against requirements.
- **Project managers** who need to oversee deliverables and review compliance status.

2 Getting Started

2.1 System Requirements

- **Docker** and **Docker Compose** (latest stable versions recommended).
- Recommended hardware: 4 GB RAM, 2 CPU cores, 10 GB free disk space.
- Ollama — the default LLM backend for all AI features (chat, import assistant, audit narratives, consistency checks). The suite ships pre-configured to use a local Ollama instance so no project data leaves your infrastructure (see Section 8.3).

2.2 Installation

1. Download the `docker-compose.yml` file from the project repository.
2. Obtain a license file (`traceboard.license`) — see Section 2.3.
3. Place the license file in the same directory as `docker-compose.yml`.
4. Open a terminal in that directory and run:

```
bash
```

```
docker compose up -d
```

```
\
```

Docker pulls the required images and starts all services: PostgreSQL, Redis, the TraceBoard, TraceDocs, and TraceTest backends, the background job worker (RQ), and nginx.

1. Once the services are healthy, open the suite in your browser at `http://localhost` (nginx serves a single frontend on port 80):

| Product | URL |

|---|---|

| TraceBoard (landing page) | `http://localhost/board` |

| TraceDocs | `http://localhost/docs` |

| TraceTest | `http://localhost/test` |

All three products share one entry point. The navigation bar at the top lets you switch between them without leaving the page (see §7).

```
root@demo:/opt/traceboard-suite# docker compose up -d
[+] Running 6/6
✓Network traceboard-suite_suite-net      Created      0.1s
✓Container traceboard-suite-db-1         Healthy      7.2s
✓Container traceboard-suite-traceboard-backend-1 Healthy      12.0s
✓Container traceboard-suite-tracetest-backend-1 Started      12.4s
✓Container traceboard-suite-tracedocs-backend-1 Started      12.3s
✓Container traceboard-suite-nginx-1      Started      12.6s
root@demo:/opt/traceboard-suite#
```

2.3 Licensing

The TraceBoard Suite uses an RSA-signed JSON license file to control access to product features.

How to obtain a license:

- Contact your sales representative for a trial or full license.
- You will receive a file named `traceboard.license`.

How to install the license:

- Place `traceboard.license` in the same directory as your `docker-compose.yml`.
- The file is automatically picked up by all services when they start.

License file format:

The license is a JSON document containing:

- `subject`: the licensed organization or individual.
- `email`: contact email.
- `issued_at` and `expires_at`: validity period.
- `products`: an array listing which products are entitled (e.g., `["TraceBoard", "TraceDocs", "TraceTest"]`).
- `signature`: a cryptographic RSA signature that prevents tampering.

License tiers and AI features:

Your license tier determines which AI features are available. The table below shows the intended mapping shipped with each tier. The AI capability level is an *independent* environment-level override — it is not derived from the license tier in code — and can be tuned per deployment via the `AI_CAPABILITY_LEVEL` environment variable (see Section 9.3).

License Tier	AI Capability Level	AI Features
Trial	basic	Limited AI queries, no audit narratives
Starter	medium	Consistency checks, basic chat
Professional	full	Full audit narratives, gap analysis
Team	full	Professional-level features for larger teams
Business	full+	Custom models, advanced reasoning

What happens if the license is missing or expired?

- If no license file is found, the services will start in a restricted mode (e.g., only the demo project is accessible).
- If the license is expired, a warning banner is displayed and access to licensed features is limited.
- Contact your sales representative to obtain or renew a license.

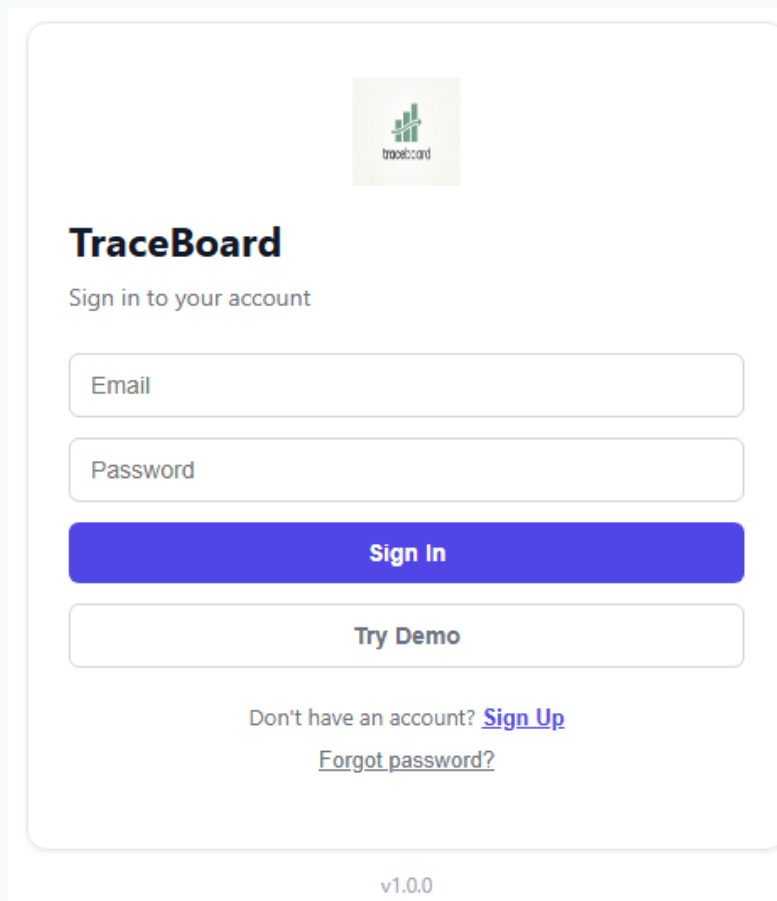
2.4 First Login

1. Open the suite at `http://localhost/board` (TraceBoard).
2. Log in with the default credentials:

> ■■ **Security Warning:** The default credentials are `admin@traceboard.local` / `admin`. Change this password immediately after first login.

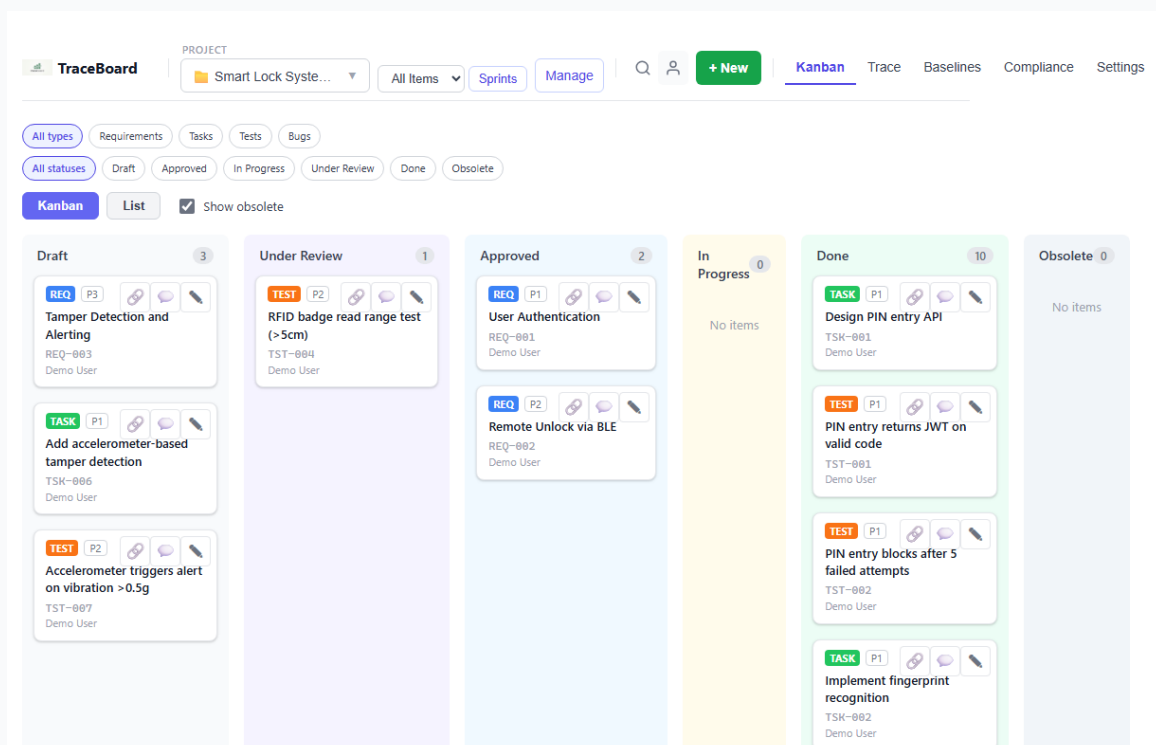
1. **Change the password immediately** from the user settings page.
2. (Optional) A demo login may be available for evaluation — check with your administrator.

All three products share the same authentication session. Logging into one product logs you into all three.



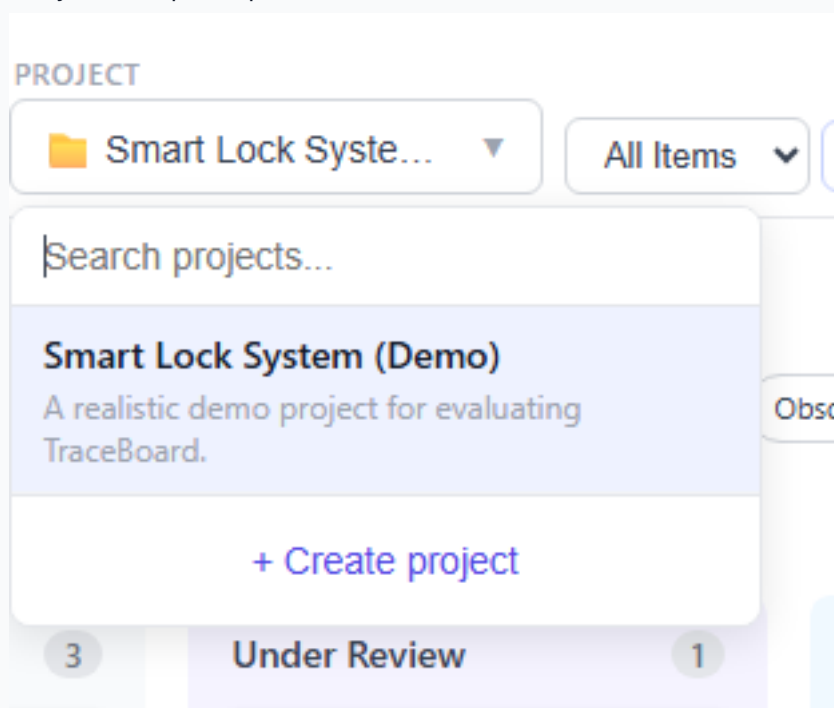
3 TraceBoard – Core Traceability

TraceBoard is the core product of the suite. It owns all domain data: projects, requirements, test cases, traceability links, and baselines. Everything in TraceDocs and TraceTest originates from TraceBoard.



3.1 Projects and Sprints

- **Projects** organize your work into separate units. Each project has its own requirements, test cases, baselines, and settings.
- Create a project from the project selector dropdown in the top navigation bar.
- Within a project, you can define **Sprints** to group work into time-boxed iterations.
- Switch between projects using the project selector — all views update to show the selected project's data.
- **Deleting a project** removes the project and its traceability data from TraceBoard. Documents already generated in TraceDocs and test runs recorded in TraceTest are **not** deleted — they are kept to preserve your audit history and prevent accidental data loss. Administrators can remove orphaned records manually if cleanup is required.



3.2 Dashboard Overview

The **Dashboard** tab opens the customizable dashboards system (see §3.1.2) and lands on your project's **default dashboard** — a real-time summary of your project's health and compliance status. The default layout starts with the following widgets:

- **Requirement Status** — Requirements grouped by workflow status.
- **Test Coverage** — Requirements that are covered, partially covered, or not covered.
- **Test Pass Rate** — Pass rate over the last 30 days.
- **Recent Activity** — Latest item, link, and comment events.
- **Traceability Completeness** — A donut chart of the percentage of requirements that have both a linked test and a linked task, with Complete vs Incomplete counts.

Every widget is a live view over the current project data. You can rearrange, resize, add, and remove widgets, save your own layouts, and export the dashboard as a PDF for management reviews. All saved dashboards are listed in the sidebar on the left of the view.

3.3 Customizable Dashboards

The **Dashboard** tab opens the customizable dashboards system — it lands on your project's default dashboard, and the sidebar lists every saved dashboard. From here you can build, save, and export project dashboards from a library of widgets:

- **Dashboard list** – A sidebar shows all saved dashboards for the selected project. The first time you open the view, a default dashboard with a starter layout is created automatically.
- **Widget toggles** – While editing, click **Manage Widgets** to open a panel with a toggle for every widget type (Requirement Status, Test Coverage, Test Pass Rate, Recent Activity, Baseline Snapshots, Traceability Completeness). Toggle a widget **ON** to add it to the grid — each type can appear only once. Toggle it **OFF** (or click the **X** on the widget header) to remove it. A removed widget keeps its position and size, so toggling it back **ON** restores it exactly where it was. The **Traceability Completeness** widget shows a donut chart of requirements that have both a linked test and a linked task.
- **Drag & drop grid** – Drag widgets by their header to reposition them, and drag the bottom-right corner to resize. Changes are only saved when you click **Save**; **Cancel** discards them.
- **Save / Load** – Saved layouts are stored per project on the server, so they are available from any device and to other project members.
- **Export PDF** – Click **Export PDF** to download the current dashboard as a landscape A4 PDF for management reviews.
- **Permissions** – Viewers can read dashboards and export PDFs; members and admins can create, edit, and delete them.

3.4 Items – Requirements, Tasks, Bugs, Tests

Items are the building blocks of traceability in TraceBoard. Each item belongs to a project and has a type:

Type	Purpose
Requirement	A functional or non-functional requirement (e.g., "The system shall allow users to reset their password").
Task	A unit of development work.
Bug	A defect report.
Test	A test case definition (used by TraceTest for execution).

Item properties:

- **Status:** Draft, In Progress, Review, Approved, Closed, etc.
- **Priority:** Low, Medium, High, Critical.
- **Owner:** The user responsible for the item.
- **Title and Description:** Free-text with Markdown support.
- **External ID:** A reference ID from an external system (e.g., JIRA, Azure DevOps).

Creating items:

1. Select a project from the project selector.
2. Click **New Item** (or the **+** button).
3. Choose the item type, fill in the details, and click **Create**.

Editing and deleting:

- Click an item to open its detail view. Edit fields inline or in a form.
- Delete an item via the actions menu (three dots). Deletion removes the item and all its relations. This action cannot be undone.

The screenshot shows the detail view for a requirement titled "Remote Unlock via BLE". The interface includes a header with the requirement ID "3668e25d-f45e-48e8-9248-5823101a5d51" and a close button. Below the header is a navigation bar with tabs for "Edit", "Comments", "History", "Attachments", and "Links". The main content area displays the requirement details: "Type: requirement", "Version: 1", and "Short ID: REQ-002". A "View Trace" button is visible. The "Title" field contains "Remote Unlock via BLE". The "Description" field contains the text: "The system must allow remote unlocking via Bluetooth Low Energy from an authorised mobile app." Below the description is a "Preview" button. At the bottom, there are several fields: "Status" (set to "Approved"), "Priority (1-5)" (set to "2"), "Owner" (set to "Demo User"), "Sprint" (set to "None (Backlog)"), and "Estimated hours" (set to "e.g., 8").

3.5 Relations and Traceability

Relations link items together to create a traceability graph. Each relation has a type:

Relation Type	Meaning
implements	A task implements a requirement.
verifies	A test verifies a requirement.
parent	Parent-child hierarchy between items (e.g., a requirement decomposed into sub-requirements).
relates-to	A generic bidirectional link.

Creating relations:

1. Open an item's detail view.
2. In the **Relations** section, click **Add Relation**.
3. Select the target item and the relation type.
4. Click **Create**.

The trace tree:

- The trace tree shows an item and all its recursively linked items.

- Expand/collapse branches to explore the trace graph.
- The tree automatically detects and flags **circular references** (e.g., $A \rightarrow B \rightarrow C \rightarrow A$). Circular references are highlighted in red with a warning icon.

Tip: Use *verifies* relations to build the compliance matrix between requirements and tests. This matrix is used by TraceTest for coverage reporting.

3.6 Trace Views

TraceBoard provides multiple ways to visualize traceability:

Flow View:

- An interactive visual graph showing items as nodes and relations as directed edges.
- Pan, zoom, and drag nodes to explore the graph.
- Click a node to open the item detail.
- Different relation types are shown with different colored arrows.

Project-wide trace matrix

[Flow](#) [Grid](#)

☐ Show gaps only (6 items with gaps)

REQUIREMENTS	TASKS	TESTS
REQUIREMENT REQ-001 GAP User Authentication Approved	TASK TSK-001 Design PIN entry API Done	TEST TST-001 PIN entry returns JWT on valid code Done
REQUIREMENT REQ-002 Remote Unlock via BLE Approved	TASK TSK-002 Implement fingerprint recognition Done	TEST TST-002 PIN entry blocks after 5 failed attempts Done
REQUIREMENT REQ-003 GAP Tamper Detection and Alerting Draft	TASK TSK-003 GAP Add RFID badge support Done	TEST TST-003 Fingerprint sensor returns match within 1 second Done
	TASK TSK-004 Design BLE communication protocol Done	TEST TST-004 GAP RFID badge read range test (> 5cm) Under Review
	TASK TSK-005 Implement end-to-end encryption for BLE Done	TEST TST-005 BLE command response time < 200ms Done

Grid View (Compliance Matrix):

- A table showing requirements (rows) against tests (columns).
- Each cell shows whether a *verifies* relation exists.
- Use this view to quickly see which requirements lack test coverage.

Project-wide trace matrix

Flow Grid

3 requirements · 6 tasks · 7 tests · 2 gaps detected

☐ Show gaps only

	REQUIREMENT ID	REQUIREMENT TITLE	LINKED TASK ID	LINKED TASK TITLE	LINKED TEST ID	LINKED TEST TITLE	TEST STATUS
 	REQ-001	User Authentication	TSK-001	Design PIN entry API	TST-001	PIN entry returns JWT on valid code	Passed
 	REQ-001	User Authentication	TSK-001	Design PIN entry API	TST-002	PIN entry blocks after 5 failed attempts	Passed
 	REQ-001	User Authentication	TSK-002	Implement fingerprint recognition	TST-003	Fingerprint sensor returns match within 1 second	Passed
 	REQ-001	User Authentication	TSK-003	Add RFID badge support	TST-004	RFID badge read range test (>5cm)	Review
 	REQ-002	Remote Unlock via BLE	TSK-004	Design BLE communication protocol	TST-005	BLE command response time <200ms	Passed
 	REQ-002	Remote Unlock via BLE	TSK-005	Implement end-to-end encryption for BLE	TST-006	Encrypted payload cannot be replayed	Passed
 	REQ-003	Tamper Detection and Alerting	TSK-006	Add accelerometer-based tamper detection	TST-007	Accelerometer triggers alert on vibration >0.5g	Draft

Highlighted rows = missing coverage (gap)

Gap Detection:

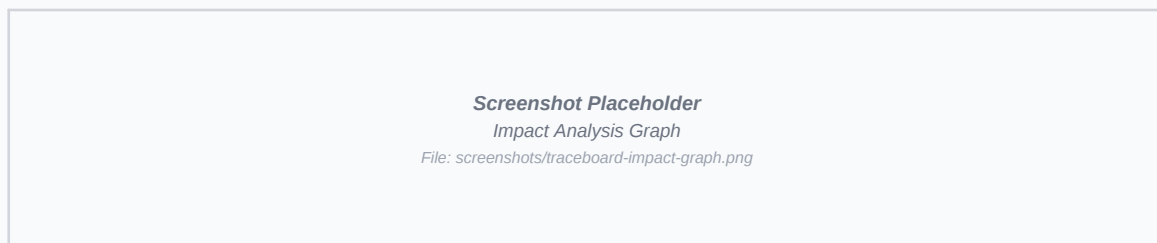
- Toggle the **"Show Gaps only"** filter to display only requirements that have no linked tests.
- This highlights coverage gaps in your trace model.

3.7 Impact Analysis

The **Impact Analysis** view shows how a change to one item ripples through the rest of your project. It renders a directed dependency graph for the selected item — ideal for change-impact assessments in safety-critical workflows (e.g., ISO 26262).

How to open it:

1. Open any item's detail drawer (from Tree, Flow, or Grid view).
2. Click  **Impact Analysis** in the drawer header.



What you see:

- **Downstream** (default) — every item *and document* that depends on the selected item; i.e., everything that is impacted when the item changes.
- **Upstream** — what the selected item itself depends on.
- Nodes are colour-coded by type (requirements blue, tasks green, tests amber, bugs red, documents grey) and show the item's status as a small dot. The root item is marked with a target icon (■).

- **Depth** — a slider controls how many levels of dependencies are traversed (1–50 hops).
- Pan, zoom, and drag the graph; click any node to jump to that item's detail drawer.

The root item is always included, and cycles in the trace model are handled safely, so even complex dependency networks render without infinite loops.

3.8 Baselines

A baseline is a point-in-time snapshot of all items and relations in a project. Baselines are used by TraceDocs to generate documents and by TraceTest to create test runs. Once created, a baseline is immutable — it captures the exact state of your project at that moment.

Creating a baseline:

1. Navigate to the **Baselines** tab in TraceBoard.
2. Click **New Baseline**.
3. Provide a name and an optional tag (e.g., v1.0, Sprint-3).
4. Click **Create**. All items and relations in the current project are frozen in the snapshot.

Comparing baselines:

1. In the Baselines view, select two baselines using the checkboxes.
2. Click **Compare**.
3. The comparison view shows:

Baselines Create Baseline

Baseline created successfully ✕

Name	Created	ID
Test Readiness Review	04/07/2026, 19:36:00	19e18308...
Initial Design Review	04/07/2026, 19:24:17	7fab7835...

Compare Baselines

Baseline A: Initial Design Review ▼ Baseline B: Test Readiness Review ▼ Compare

☐ Show system fields

Modified Items (2)

3f221966

Field	Old Value	New Value
status	under_review	done

61a2f8f8

Field	Old Value	New Value
status	draft	approved

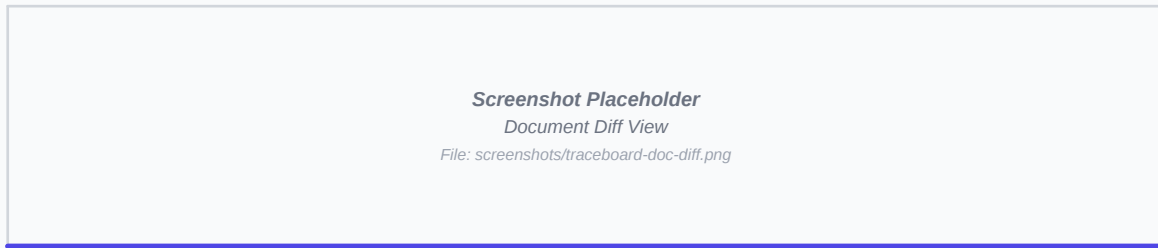
3.9 Document Diffing (Baseline Comparison of Generated Documents)

When documents have been generated for two baselines (via TraceDocs), you can compare the generated revisions side by side to see exactly what changed between baseline versions.

How to diff generated documents:

1. In the **Baselines** view, select two baselines with the checkboxes.

2. Switch to the **Documents** tab.
3. Click **Load Documents Diff**.

**What you see:**

- One entry per document type — **SRS**, **Test Plan**, **Design Description**, and **User Manual** — showing the document version stored in each baseline.
- **Added lines** (highlighted green) — content present in the newer baseline only.
- **Removed lines** (highlighted red) — content that was present in the older baseline but removed or changed.
- **Stats** — counts of added, removed, and unchanged lines for the selected document type.
- **Export PDF** — download the current diff for review records.

Notes:

- Only *generated* documents are compared (drafts and soft-deleted documents are ignored); for each type the latest generated revision of each baseline is used.
- If a document type was not generated for one of the two baselines, it is reported as *unavailable* for that type instead of failing the whole comparison.

3.10 Import / Export

Import from Excel:

1. Go to the project's import page.
2. Upload an `.xlsx` file with your items.
3. The Excel file must include columns for type, title, description, status, priority, and owner.
4. The **Relations** sheet (optional) supports two column formats:
 1. After selecting the file, a preview shows the relations using human-readable IDs (short IDs or `source_id` values) so you can verify them before importing.
2. Review the preview and click **Confirm Import**.

Export to Excel:

1. Go to the project's export page.
2. Click **Export Excel**.
3. The downloaded file includes an `items` sheet (all items with their UUIDs) and a `relations` sheet (all relations with source/target UUIDs).
4. This file can be re-imported (e.g., for bulk editing or backup).

3.11 LLM Chat

TraceBoard includes an "Ask AI" assistant that can answer questions about your trace data using natural language. This feature requires an LLM endpoint to be configured — either a local Ollama instance or a cloud provider (see Section 8.3).

Enabling the feature:

- Once the LLM is configured with `LLM_API_URL` and `LLM_MODEL` environment variables, an **"Ask AI"** button appears in TraceBoard.

Using the Ask AI feature:

1. Click the **"Ask AI"** button in the TraceBoard toolbar.
2. Type your question in natural language (e.g., "Which requirements have no linked tests?").
3. The AI responds with a detailed answer. **Citations** appear as clickable references — click a citation to navigate directly to the referenced item.
4. You can ask follow-up questions to drill deeper into the data.

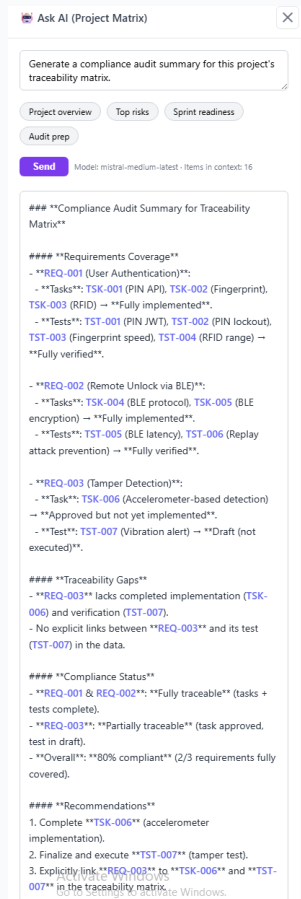
Improved user experience:

- **Polished, human-readable responses** — AI answers are now rendered in a clean, readable format instead of raw JSON.
- **Structured coverage analysis** — For coverage questions, the AI shows a structured analysis with a list of missing test scenarios.
- **Plain-English trace explanations** — For trace explanations, the AI provides a clear, plain-English description of the trace chain.
- **Context-aware answers** — The AI is now context-aware: if you ask about "this item" from the Trace View, the system automatically passes the selected item's context to the AI.

Example questions you can ask:

- "Which requirements have no linked tests?" (gap analysis)
- "Summarise the trace coverage for project X."
- "What is the highest priority requirement that has not been verified?"
- "Identify any risks based on the current trace data."

The AI assistant has read-only access to your project's items, relations, and baselines. It does not modify any data.



3.11.1 Fast vs Detailed mode (thinking control)

On providers that support reasoning/thinking mode (e.g. DeepSeek v4), TraceBoard lets you choose between **Fast** and **Detailed** AI responses per use case:

Feature	Default	User toggle	Effect
LLM Chat	Detailed (thinking ON)	"Show reasoning" checkbox in the chat panel	Detailed answers show step-by-step reasoning; Fast answers are quicker and cheaper
Import Document Assistant	Detailed (thinking ON)	"Detailed analysis" checkbox	Deeper document analysis; turn off for faster extraction
Test Case Generation	Detailed (thinking ON)	"Detailed mode (AI reasoning)" checkbox	More thorough test design; turn off for faster generation
Audit Narrative	Fast (thinking OFF)	no toggle	Deterministic, cheap, immune to the empty-response failure mode
Consistency Check	Fast (thinking OFF)	no toggle	Deterministic, cheap, immune to the empty-response failure mode

- When the user toggle is **off**, the backend sends `thinking: {"type": "disabled"}` to the provider, which typically returns faster, cheaper, and more deterministic output.
- The **global** `LLM_DISABLE_THINKING` environment variable only applies to features without a user toggle (and as a fallback for features that have one). See Section 9.3 for the environment variable reference.

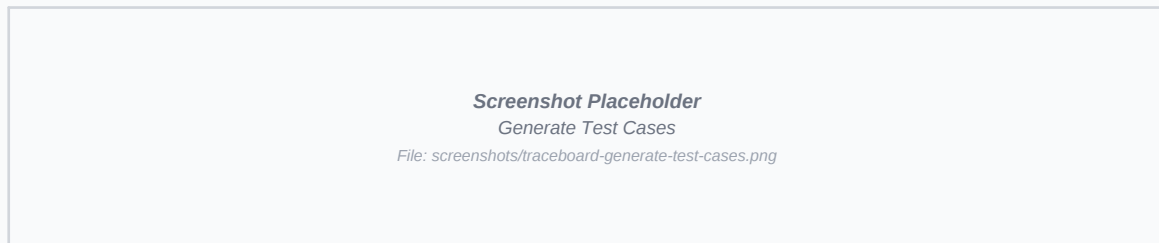
3.12 Generate Test Cases

TraceBoard can automatically generate test cases from your requirements and tasks using AI. This feature uses the configured LLM to analyse your items and propose relevant test cases.

Generating test cases:

1. Navigate to a project and open the **Generate Test Cases** dialog (found in the project actions menu).
2. Select which requirements or tasks you want to generate test cases from.
3. Click **Generate**. The AI analyses the selected items and proposes test cases.
4. Review the generated test cases in the preview table. Each proposed test case includes a title, description, and suggested priority.
5. **Edit** any test case before importing by clicking on it in the preview.
6. **Deselect** any test cases you do not want to import.
7. Click **Import** to create the selected test cases in your project. New test cases are automatically linked to their source items via `verifies` relations.
8. Refresh the item list to see the newly created test cases.

Tip: You can re-run the generation with different selections or LLM settings. Generated test cases are only created after you explicitly confirm the import.



3.13 Cross-Module Consistency Check

The consistency check detects contradictions and inconsistencies across your requirements, design documents, and test cases. This is especially valuable for compliance with standards like ISO 26262, IEC 62304, and DO-178C.

Running a consistency check:

1. Navigate to the **Consistency** tab in TraceBoard (or access it from the AI tools menu).
2. Select the scope of the check — you can run it against a specific baseline, a project, or across multiple modules.
3. Click **Run Check**. The check is enqueued as a **background job** and the page returns immediately — you can keep working in TraceBoard while it runs.
4. While it runs, the page shows a progress spinner and polls the job status automatically every couple of seconds. You can even navigate away: the pending task is remembered (per project), so when you come back the page resumes checking automatically. A **Refresh status** button lets you re-check manually at any time.

5. When the check finishes you are notified in two ways:

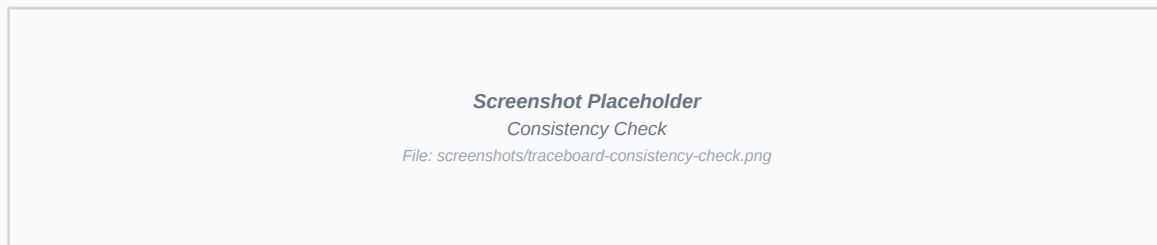
1. Each finding includes:

1. Review the findings and address them by updating items, adding missing relations, or creating new test cases.

Validated findings: Every finding is grounded in your actual project data. Citations are cross-checked against existing relations (e.g., IMPLEMENTS, VERIFIES) and are only included if they are present — the same deterministic validation layer used by the Audit Narrative (Section 3.10). This ensures the results are both accurate and auditable.

Generation Info: Below the summary bar the report shows a **Generation Info** block with the **start time**, **finish time** (both displayed in your local timezone, with the timezone abbreviation such as EEST or GMT+3), and the **duration** of the run. The duration is a useful estimate for how long future checks on projects of similar size will take. This block is only shown for reports that carry the metadata — older saved reports do not include it.

Tip: Run consistency checks before creating baselines and generating compliance documents to catch issues early.



Saving and re-opening consistency reports:

When a consistency check finishes, a **Save Report** button appears next to the export buttons. Click it to store the full report for later.

1. Optionally enter a name for the report (e.g. "Pre-release consistency check").
1. Click **Save Report**. The report is stored for the current project.
2. To open a saved report later, navigate to **Reports & Audit → Consistency**
1. Click a report name to re-render the original report view — including all
1. Use the **Delete** button to remove a report you no longer need.

Reports are scoped to the project: you only ever see the reports saved for the project you are currently working in. If an item referenced in a saved report is later deleted, the link fails gracefully with a "item not found" message.

Tip: Save consistency reports before making major changes to your project so you can compare findings across time.

3.14 Audit Narrative

The audit narrative feature generates a compliance-ready narrative from your project data, covering ISO 26262, IEC 62304, and DO-178C. The AI is never the source of truth for facts or figures — it writes narrative prose around data that has already been calculated and verified.

How it works:

1. **Deterministic data collection and calculation.** Before any AI involvement, TraceBoard pulls requirements, tasks, tests, relations, and documents from your project (or a frozen baseline) and calculates coverage percentages, a readiness score, and standard-specific facts — for example, whether hazard analysis artifacts (HARA, or SSA/FHA for DO-178C) are present. These figures are computed directly from your data, not estimated by the AI.
2. **Narrative generation.** The AI writes the Executive Summary, Requirements Coverage, Test Execution, Gaps, Risk Assessment, and Recommendations sections using the verified data and calculations from step 1 as its source of truth, following standard-specific terminology (e.g., DO-178C narratives use "SSA/FHA," never "HARA").
3. **Automatic validation.** Every generated narrative is checked against your actual project data before you see it: every claimed link between items (e.g., "TST-005 verifies REQ-001") is verified against real relations in your project; claimed safety artifacts are checked against what actually exists; contradictions (e.g., "100% coverage" alongside "gaps identified") are flagged; and vague or unverifiable claims are marked. If validation fails, the system automatically retries with stricter instructions.
4. **Review.** If any issues remain after validation, the narrative includes a warning banner and a Retry Generation button. **A narrative with an active warning should not be treated as audit-ready** — review and resolve the flagged items before export.

Every section is labeled to show whether its content is **Verified** (from your data) or **AI Analysis**, so you always know which parts are computed fact and which are AI-generated narrative.

Generating a narrative (runs in the background):

1. Click **Generate Narrative**. Generation is enqueued as a **background job** and the page returns immediately — the UI never blocks and you can keep working in TraceBoard.
2. While it runs, the page shows a spinner with the message *"Narrative generation is running in the background. This may take several minutes — this page will update automatically when it is ready."* A **Refresh status** button lets you check manually at any time.
3. You can navigate away (even to another product) and come back later: the pending task is remembered per project, and the page resumes polling when you return.
4. When the narrative is ready you are notified in two ways:
 1. The latest narrative is **saved automatically**: even if you never return to the page, the generated report is stored in the database and reloaded the next time you open the Audit Narrative tab (a *"Report loaded from storage"* note is shown). Use **Regenerate** to produce a new narrative (a confirmation prompt warns you the existing report will be replaced) or **Clear report** to remove the stored report.
 2. To publish the narrative as a formal document, click **Save to TraceDocs**. The narrative is pushed to TraceDocs as a generated document for the selected baseline, where you can export it as PDF or Word (see Section 4).

Generation Info: Above the narrative content, the header shows a **Generation Info** block with the **start time**, **finish time** (both displayed in your local timezone), and the **duration** of the run — useful for estimating how long future narratives on projects of similar size will take. This block is only shown for reports that carry the metadata (it is not persisted with stored audit narratives, so reports loaded from storage will not show it).

4 Recommended Models for Audit Narratives

The quality and factual accuracy of the generated audit narrative depend heavily on the underlying large language model. The default deployment uses a local **Ollama** instance (air-gapped, no data leaves your

infrastructure) — the recommended starting point for sovereignty- and compliance-sensitive customers. Based on extensive testing with ISO 26262 compliance data, we recommend the following tiers:

Tier	Model Size / Type	Recommended Use Case	Expected Accuracy
Development	≤ 8B parameters (e.g., Qwen 7B, Llama 3.1 8B)	UI testing, workflow validation, internal prototyping	May hallucinate metrics and contradict itself. Not suitable for export.
Draft	~26B–40B (e.g., Gemma 26B, Mistral 34B)	Internal drafts, early gap analysis	Generally coherent but may misinterpret complex traceability logic (e.g., direct vs. indirect links). Requires human review.
Production	≥ 70B dense or ≥ 100B MoE (e.g., DeepSeek v4 pro, Nemotron 120B, GPT 4o, Claude 3.5 Sonnet)	Customer-facing reports, audit-ready exports	High factual accuracy, correct interpretation of traceability graphs, and clause-specific regulatory references.

4.1 Important Notes

- Smaller models will produce a report, but they may contain factual errors or logical contradictions. Always review the output before sharing with external stakeholders.
- For automated or compliance-critical workflows, we strongly recommend using a Tier 3 (Production) model.
- If you choose a smaller model for speed or local deployment, treat the output as a **DRAFT** and verify the following manually:

4.2 Import Assistant

The Import Assistant extracts structured requirements from Word (.docx) and PDF documents using AI. Instead of manually retyping requirements from specification documents, you can upload them and let the AI do the extraction.

Using the Import Assistant:

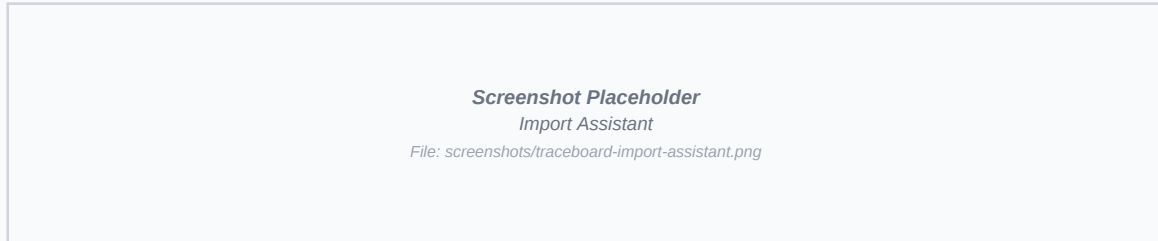
1. Navigate to the **Import Assistant** from the project import page or the AI tools menu.
2. Upload a Word (.docx) or PDF document containing requirements (e.g., a customer specification, regulatory document, or legacy SRS).
3. Click **Analyse**. The AI processes the document and extracts candidate requirements.
4. Review the extracted items in the preview table. Each row shows the proposed title, description, type, and priority.
5. **Edit** any item by clicking on it — adjust the title, description, type, or priority as needed.
6. **Deselect** items you do not want to import.
7. Click **Import** to create the selected items in your project.
8. After import, you can add relations, assign owners, and include the items in baselines as usual.

Validated extraction: The Import Assistant uses the same propose■validate■approve pipeline as the Audit Narrative (Section 3.10). Extracted requirements are cross■checked against your existing project data, and citations are only included when they are grounded in the source document — so the proposed items are both accurate and auditable.

Supported formats:

- **Word (.docx)** — Standard Word documents.
- **PDF** — Text-based PDF documents (scanned documents without OCR may not extract well).

Tip: The Import Assistant works best with structured documents where requirements are clearly labelled (e.g., "REQ-001", "The system shall...").



4.3 Import Tasks from File

You can import a list of tasks from an Excel, CSV, DOCX, MD, or PDF file and automatically create them in TraceBoard, linking them to existing requirements via `IMPLEMENTS` relations. This is useful when your task backlog lives in a spreadsheet or external tool.

Supported formats: `.xlsx`, `.csv`, `.docx`, `.md`, `.txt`, `.pdf`.

How it works:

1. Upload a file containing tasks via the **"Import Tasks"** option in Settings → Data Management.
2. The AI parses the file and extracts the tasks (titles, descriptions, priorities, estimates, and optional requirement references).
3. The system validates each task:

Preview and resolution:

- The preview table shows all extracted tasks, with color-coded flags for unresolved references and duplicates.
- Users can resolve conflicts inline:
- Bulk actions (e.g., **"Skip all duplicates"**) are available for large imports.

Inference of grouped rows:

- If the Excel file uses a grouped format (only the first task under a requirement has the reference ID, and subsequent tasks leave it blank), the system automatically inherits the last seen requirement reference for those blank rows.

Completion:

- After confirming the preview, the tasks are created and linked to their requirements.

5 TraceDocs – Document Generation

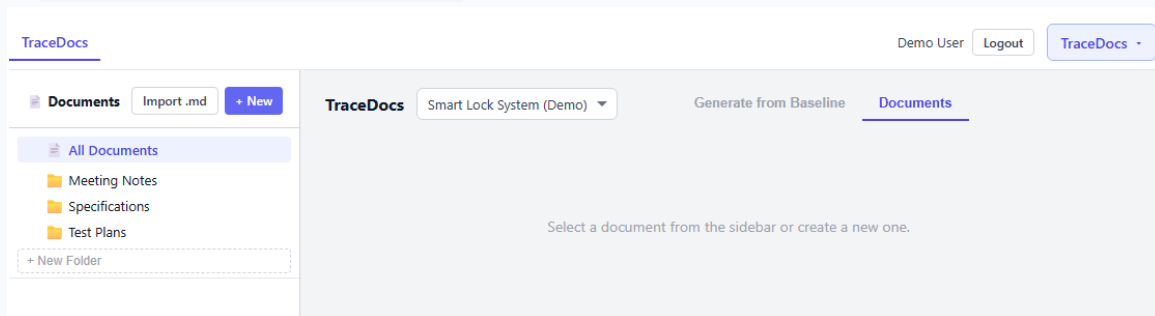
5.1 Overview

TraceDocs generates compliance-grade documents — Software Requirements Specifications (SRS), Test Plans, Design Descriptions, and User Manuals — directly from TraceBoard baselines. Instead of manually copying and formatting data, you select a baseline, choose a template, and TraceDocs produces a professional PDF or Word document. Every generated document embeds its source baseline ID for full traceability.

5.2 Document Management

Creating a document generation:

1. Navigate to TraceDocs (use the Product Switcher or go to `http://localhost/docs`).
2. Click **New Document**.
3. Configure your document:
 1. Click **Generate**. A progress indicator shows the generation status.
 2. When complete, review the document in the **HTML Preview**.
3. Click **Export** and choose **PDF** or **Word** format to download. The file is downloaded with a descriptive filename: `ProjectName_DocType_BaselineTag_Date.pdf` (e.g., `MyProject_SRS_v1.0_2026-06-29.pdf`).

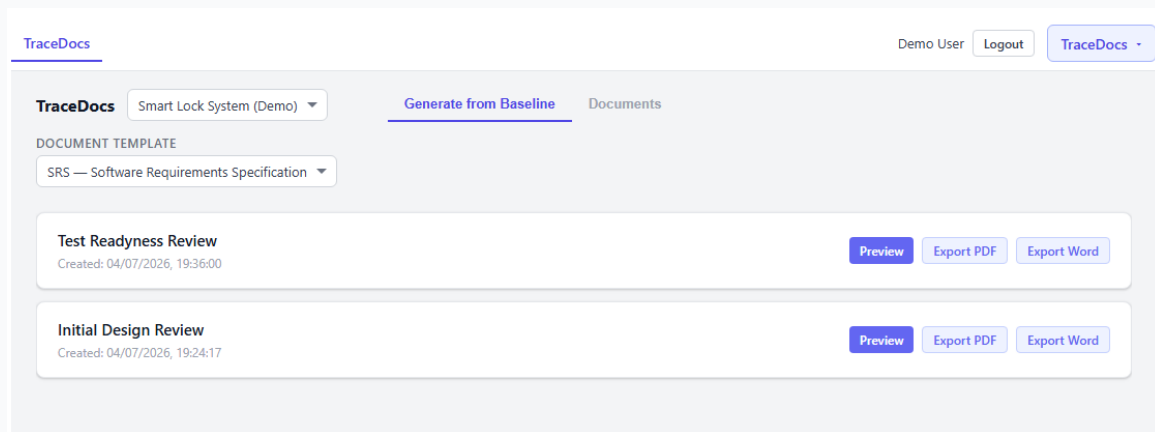


Organising documents into folders:

- Documents can be organised into a nested folder hierarchy in the Knowledge Base.
- Click **+ New Folder** to create a folder. Use the folder's actions menu to rename or delete it.
- Create a sub-folder by selecting an existing folder and adding a new folder inside it.
- Move a document between folders by dragging it onto the target folder in the folder tree, or by choosing a folder from the **Folder** dropdown in the document editor.
- Deleting a folder moves its documents and sub-folders up to the parent folder (or the root level) — it does not delete documents.

Generation history:

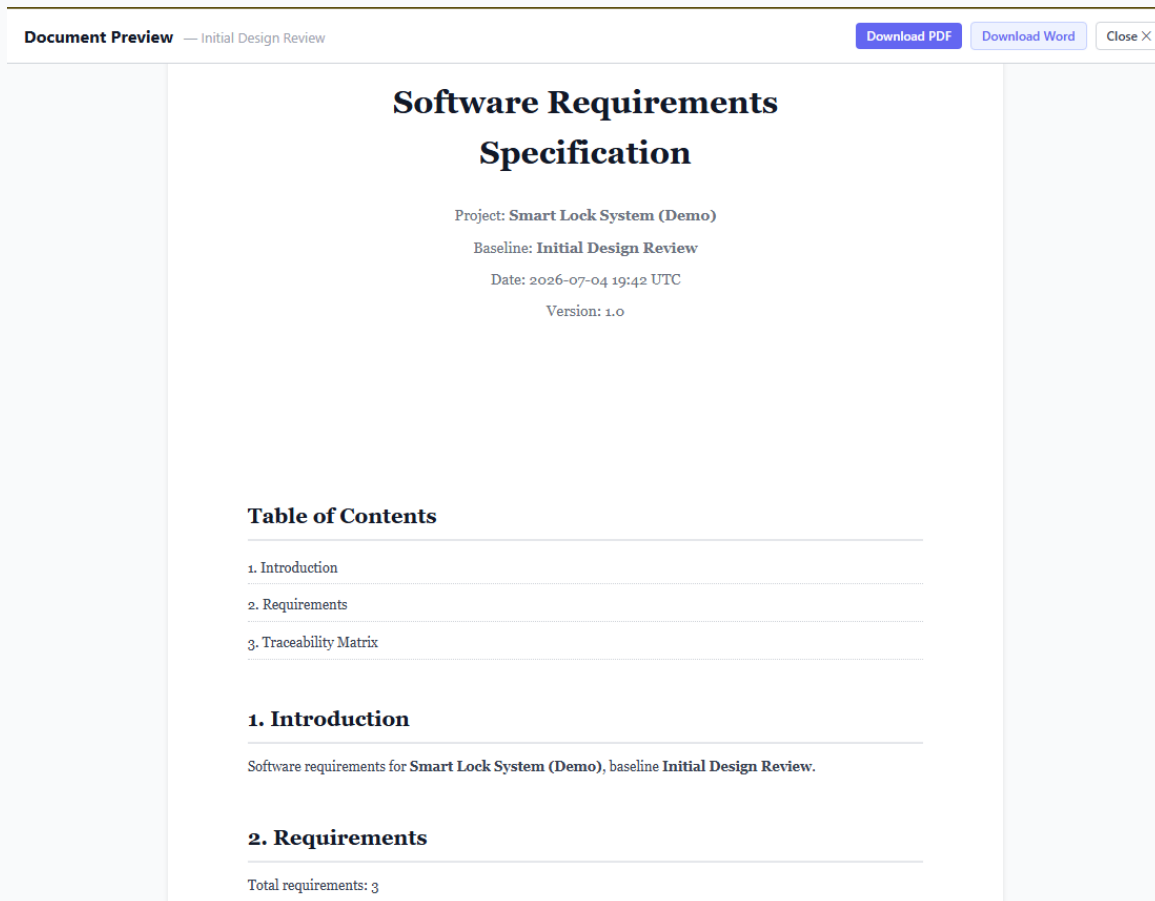
1. Click the **History** tab in TraceDocs.
2. View a list of all past document generations, including date, template type, baseline used, and status.
3. Click **Download** to re-download a previously generated file.
4. Click **Re-generate** to pre-fill a new document form with the same settings.



5.3 Generating Documents

To generate a document:

1. **Select a baseline** — this determines which requirements and test cases are included.
2. **Choose a template** — SRS, Test Plan, Design Description, or User Manual (see Section 4.4).
3. **Click "Generate Document"** — the backend fetches the baseline data from TraceBoard, merges it with the template, and produces a styled document.
4. **Preview** — review the HTML preview in your browser.



1. **Export** — download as PDF or Word (.docx). The filename is descriptive: `ProjectName_DocType_Baseline_Date.pdf`.

The PDF includes a cover page, auto-generated table of contents, styled headings, page numbers, and a generation timestamp. The Word export mirrors the PDF styling with a Word-native TOC field.

[Preview](#)[Export PDF](#)[Export Word](#)

5.4 Document Templates

TraceDocs comes with four built-in templates:

Template	Description
SRS	Software Requirements Specification — structured with sections for introduction, functional requirements, non-functional requirements, interfaces, and appendices.
Test Plan	Test planning document — includes test strategy, scope, test cases table, schedule, and resource sections.
Design Description	Software Design Description — documents the system architecture and design derived from the baseline.
User Manual	End-user manual — describes how to operate and administer the product.

Templates currently planned for future releases:

- **Custom templates** — user-defined template creation via the UI.

5.5 Markdown Editor

TraceDocs includes a built-in Markdown editor for writing knowledge base articles, design notes, and supplementary documentation.

Using the editor:

1. Navigate to the **Knowledge Base** section in TraceDocs.
2. Click **New Article** to create a new document.
3. Write your content in Markdown in the left pane. The right pane shows a **live preview** of the rendered output.
4. Organise articles into **folders** — create a folder structure that matches your project's documentation needs.
5. Click **Save** to persist your article.

Tip: Use the Markdown editor to document design decisions, meeting notes, or supplementary material that complements your generated SRS, Test Plan, Design Description, and User Manual documents.

5.6 Export Word

Generated documents can be exported as Word (.docx) files for further editing in Microsoft Word or compatible editors.

Exporting to Word:

1. After generating a document, click the **Export** button in the preview toolbar.

2. Select **Word (.docx)** from the dropdown menu.
3. The file downloads with a descriptive filename: `ProjectName_DocType_BaselineTag_Date.docx` (e.g., `MyProject_SRS_v1.0_2026-06-29.docx`).
4. Open the file in Microsoft Word or your preferred editor. The document includes:

Tip: The Word export is ideal for teams that need to add final polish, signatures, or custom formatting before submission.

5.7 Bi-Directional Document-Item Linking

TraceDocs now supports linking specific paragraphs or text selections directly to TraceBoard items (requirements, tasks, tests). This provides complete traceability from the document source to the requirement.

How to link:

1. In TraceDocs, open a document (e.g., an SRS or Test Plan).
2. Select a text paragraph or sentence.
3. Right-click and choose **"Link to TraceBoard Item"**.
4. A modal opens where you can search for and select a TraceBoard item (requirement, task, or test).
5. After confirmation, the link is created.

How links are displayed:

- **In TraceDocs** — the linked text is highlighted with a small document icon. Click the icon to open a popover showing the linked item's details and an **"Open in TraceBoard"** button.
- **In TraceBoard** — each item that has a document link displays a document icon next to its title. Clicking that icon opens a popover showing the linked text snippet and an **"Open in TraceDocs"** button that navigates directly to the linked section.

Benefits:

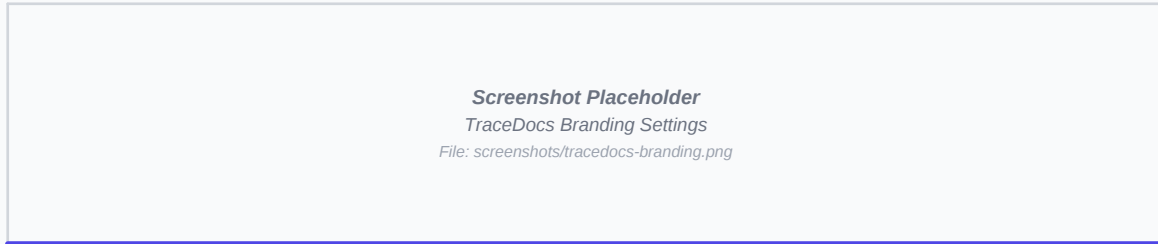
- Auditors can instantly see which requirement originated from which document section, saving hours of manual cross-referencing.

5.8 Corporate Branding

TraceDocs supports customizable corporate branding for exported documents. Administrators can upload a company logo and set a footer text (for example, *"Confidential – ACME Corp"*) that is automatically applied to every generated PDF and Word document — SRS, Test Plan, Design Description, and User Manual.

Where to configure it:

1. In TraceDocs, open the **Settings** tab.
2. Scroll to the **Project Branding** card.
3. **Upload a logo** (PNG or JPEG, up to 5 MB). A live preview is shown immediately.
4. **Enter the footer text** — for example `Confidential – ACME Corp`.
5. Click **Save Project Branding**.

**How branding appears in documents:**

- **Logo** — placed in the header of every page (except the title page, which stays clean).
- **Footer text** — centered at the bottom of every page.

Per-project vs global defaults:

- Branding is configured **per project**. Different projects can have different logos and footers.
- If a project has no branding of its own, it automatically **inherits the global default** configured by an administrator in the **Global Default Branding** card.
- To stop using project-specific branding, click **Reset to Global Defaults** — the project immediately falls back to the global logo/footer.

Logo recommendations:

- Use a **transparent PNG** for the best look on white paper.
- Logos are scaled to fit a 1.6 inch header area; very wide images are capped at that width while keeping their aspect ratio.
- If the logo file is replaced, existing generated documents keep their old branding until they are regenerated.

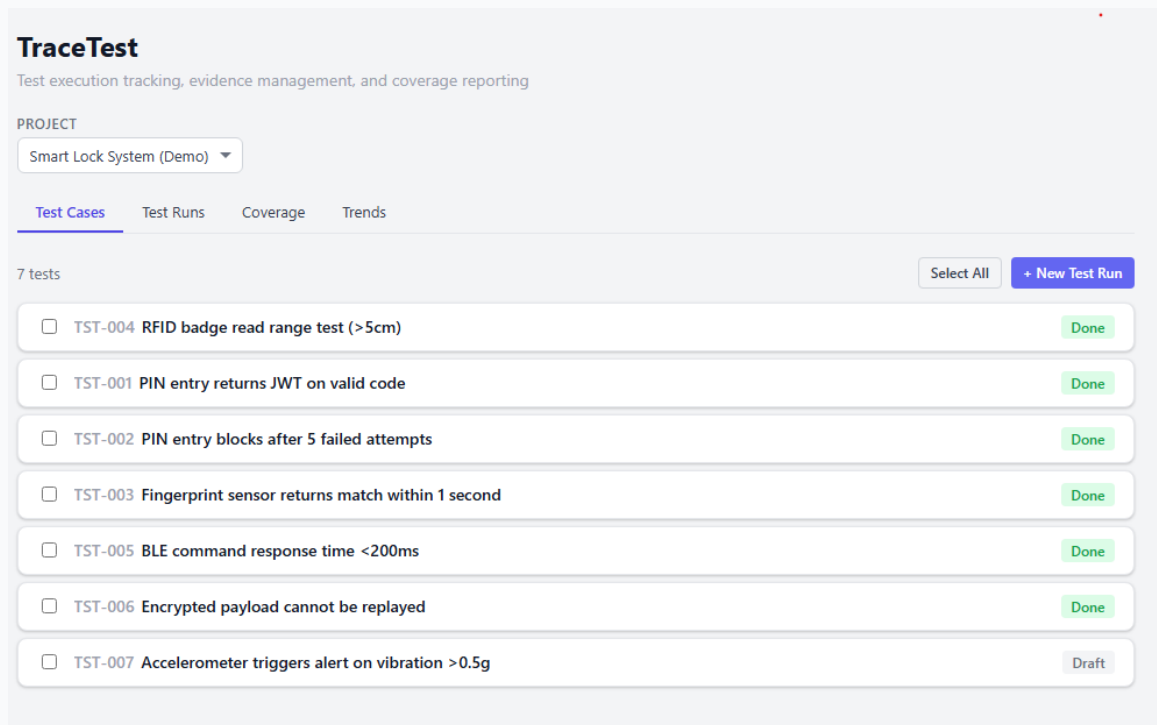
Tips:

- Only **project admins and owners** can change branding; other users see the current settings in read-only mode.
- Auto-generated documents (created when a baseline is approved in TraceBoard) use the same branding automatically.

6 TraceTest – Test Execution and Coverage

6.1 Overview

TraceTest tracks the execution of test cases defined in TraceBoard baselines. Testers record results (Pass/Fail/Blocked/Not Run), attach evidence, and generate coverage reports that show which requirements are verified by passing tests.



6.2 Test Cases

All items of type test from TraceBoard automatically appear as test cases in TraceTest. Test cases are grouped by project and are read-only — they are managed in TraceBoard.

To view test cases:

1. Navigate to TraceTest (use the Product Switcher or go to <http://localhost/test>).
2. Select a project from the project selector.
3. The test cases for that project are listed, grouped by the baseline they belong to.

6.3 Test Runs

A **Test Run** is a named execution session where test cases from a baseline are executed and results are recorded.

Creating a test run:

1. In TraceTest, click **New Test Run**.
2. Provide a name for the test run.
3. Select a baseline from the dropdown (populated from TraceBoard projects).
4. Optionally, select specific test cases to include — or include all test cases from the baseline.
5. Click **Create**. TraceTest seeds the run with the selected test cases, all initially set to "Not Run".

×

New Test Run

Run Name

e.g. Sprint 1 Regression

All 7 tests will be included

Cancel

Create Run

Recording results:

1. Open a test run from the runs list.
2. For each test case, click the status chip to set the result:
 1. Changes save automatically — no save button needed.
 2. Optionally, add a **note** to each result explaining the outcome or linking to a bug report.

TraceTest

Test execution tracking, evidence management, and coverage reporting

PROJECT

Smart Lock System (Demo) ▾

Test Cases **Test Runs** Coverage Trends

1 test run

[+ New Test Run](#)**Sprint 3 Regression**

7 tests · 0 pass · 0 fail · 0 blocked · 04/07/2026

Close

Delete

TST-004 RFID badge read range test (>5cm)

Pass

Fail

Blocked

 Upload evidence

TST-001 PIN entry returns JWT on valid code

Pass

Fail

Blocked

 Upload evidence

TST-002 PIN entry blocks after 5 failed attempts

Pass

Fail

Blocked

 Upload evidence

TST-003 Fingerprint sensor returns match within 1 second

Pass

Fail

Blocked

Bulk updates:

1. Use checkboxes to select multiple test cases.
2. Click **Bulk Update**.
3. Choose the new status to apply to all selected test cases.

4. Confirm the action. All selected test cases are updated simultaneously.

Filtering and search:

- Click status chips (Pass, Fail, Blocked, Not Run) to filter the list.
- Use the free-text search to find test cases by title or external ID.
- Filter by functional area via the dropdown.
- All filter selections are preserved in the URL for sharing and bookmarking.

6.4 Coverage Report

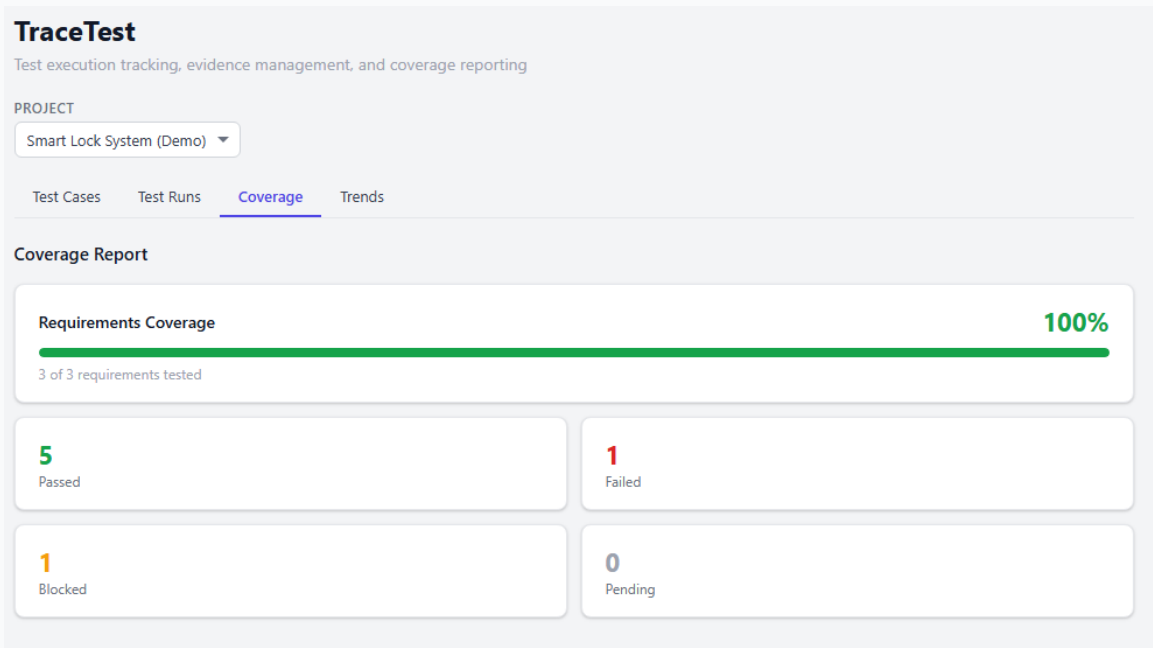
Coverage is calculated automatically based on `verifies` relations from TraceBoard and test results from TraceTest.

How coverage is calculated:

- A requirement is **Covered** if at least one linked test case has a passing result.
- A requirement is **Partially Covered** if some linked tests passed but others failed or are not run.
- A requirement is **Not Covered** if no linked tests have a passing result.
- **Coverage percentage** = (Covered requirements / Total requirements) × 100.

Viewing the coverage report:

1. In a test run, click the **Coverage** tab.
2. View the summary cards showing Covered, Partially Covered, Not Covered counts and the coverage percentage.
3. The requirement-level table shows each requirement with its coverage status and linked tests.
4. Click a requirement to drill down and see per-test results.



6.5 Evidence Upload

Attach files (screenshots, logs, test data) to individual test results as evidence of execution.

Uploading evidence:

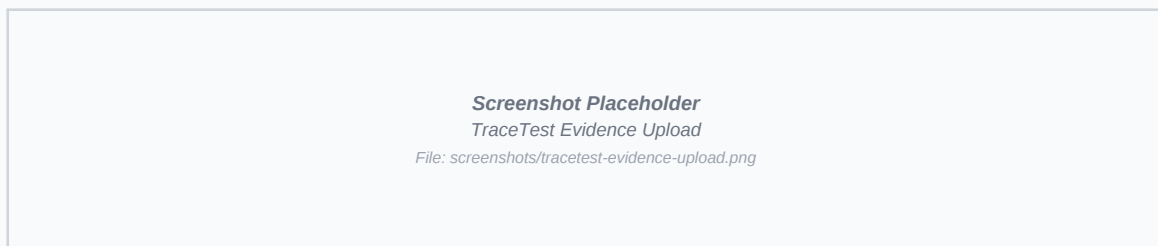
1. Open a test run and locate the test result you want to attach evidence to.

2. Click the **Attach Evidence** button (paperclip icon) next to the result.
3. Select one or more files from your computer. Supported file types: PNG, JPG, GIF, TXT, LOG, CSV, JSON, PDF, ZIP.
4. Maximum file size: 25 MB per file.
5. Uploaded files appear in the result row with a thumbnail or file icon. Click a file to download or preview it.
6. Evidence attachments are stored alongside test results and included in exported test report PDFs.

Managing evidence:

- Click an uploaded file to download it.
- Click the delete icon to remove an attachment (requires confirmation).
- All evidence for a test run is visible in the run detail view.

Tip: Use evidence uploads to capture screenshots of test execution, log files from test runs, or signed-off test result sheets. This creates a complete audit trail for compliance reviews.



6.6 Coverage Trends

The Coverage Trends view shows how your test coverage has evolved over time across multiple test runs.

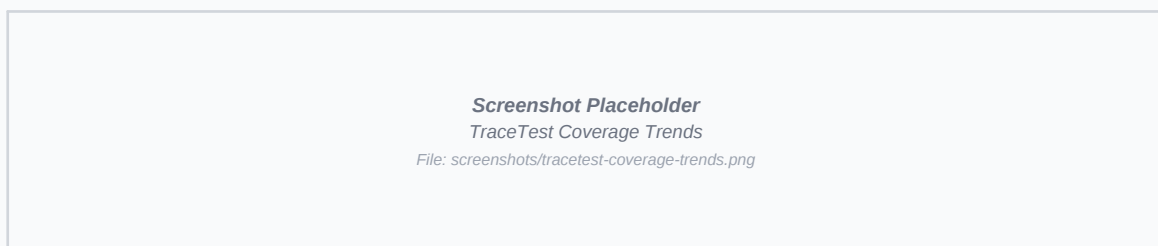
Viewing coverage trends:

1. In TraceTest, navigate to the **Coverage Trends** tab.
2. Select a project and optionally filter by time range.
3. The trends chart displays:
 1. Hover over any data point to see the exact values for that test run.
 2. Use the chart to identify coverage regressions or improvements across sprints and releases.

Interpreting the trends:

- An **upward trend** indicates increasing test coverage.
- A **downward trend** may indicate new requirements added without corresponding tests.
- **Flat lines** suggest no new testing activity.

Tip: Review coverage trends before release milestones to ensure coverage is trending in the right direction.



6.7 Test Report Export

Export test run results and coverage data as a PDF report for distribution, archiving, or audit submission.

Exporting a test report:

1. Open a test run in TraceTest.
2. Click the **Export Report** button in the run toolbar.
3. The system generates a PDF report containing:
 1. The PDF downloads automatically. The filename includes the test run name and date:
`TestRunName_YYYY-MM-DD.pdf`.

Tip: Export reports after each test run and archive them as part of your compliance documentation. The reports embed source baseline information for full traceability.

7 Git Integration

TraceBoard Suite includes a built-in Git integration that links code commits to requirements, tasks, test cases, and documents. It supports **GitHub**, **GitLab**, and **Gitea** (including self-hosted instances) and works in air-gapped environments.

7.1 Key Capabilities

- **Automatic commit linking** — Webhooks push commit events from your repository to TraceBoard in real time.
- **Manual sync** — Fetch and process commits on demand when webhooks are not feasible.
- **Branch filtering** — Restrict processing to specific branches (e.g., `main`, `develop`).
- **Private repository support** — Access tokens are encrypted at rest.
- **Commit visibility across products** — Linked commits appear in TraceBoard item drawers, TraceDocs document views, and TraceTest test case/run details.
- **CI/CD API** — Create test runs automatically from GitHub Actions, GitLab CI, or any CI pipeline.

7.2 Quick Start

1. Navigate to your project's **Settings** → **Git Integration**.
2. Enter the **Repository URL** and select the **Git Provider**.
3. (Optional) Configure **Branch Filtering** and enter an **Access Token** for private repos.
4. Click **Save**, then **Test Connection**.
5. Set up a **webhook** in your Git provider using the URL and secret shown in the settings.

For detailed instructions, including webhook configuration, manual sync, CI/CD integration, and troubleshooting, see the Git Integration & CI/CD Guide.

7.3 Where Commits Appear

Product	Location	Details
TraceBoard	Item Cards, Trace View / Item Drawer (Commits tab), Project Commit History (sidebar)	Badge shows linked commit count; full list with SHA, message, author, date

Product	Location	Details
TraceDocs	Document detail page (Commits tab)	Commits linked to items in the document's baseline or referenced in Markdown
TraceTest	Test Case Detail (Commits tab), Test Run Detail (Commits tab)	Commits linked to the test case or included test cases in a run

8 Unified SPA Shell & Product Switcher

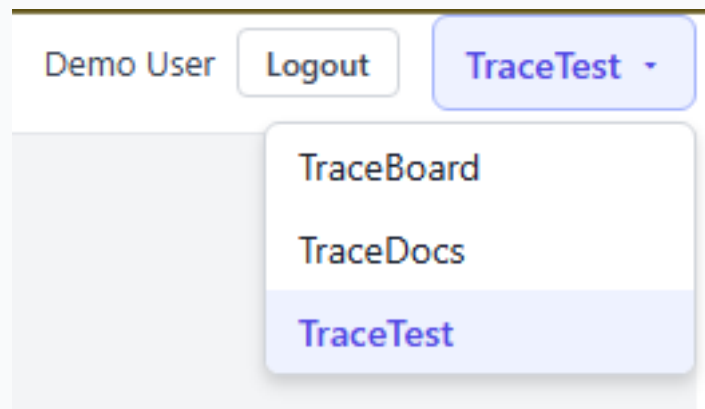
8.1 The Unified SPA Shell

The TraceBoard Suite ships as a **single web application** (the *suite shell*). All three products — TraceBoard, TraceDocs, and TraceTest — are loaded into the same page, so you never have to navigate to a different URL or log in again when moving between them.

- A persistent **navigation bar** is visible at the top of every screen. It shows the product switcher, the active project selector, a real-time notification bell, and the user menu.
- Each product's interface is **lazy-loaded** on first visit, so the shell stays fast even though it contains all three products.
- Deep links such as `http://your-server/docs` open the correct product automatically.

8.2 The Product Switcher

The Product Switcher lets you move between TraceBoard, TraceDocs, and TraceTest without leaving the page or logging in again.



How to switch products:

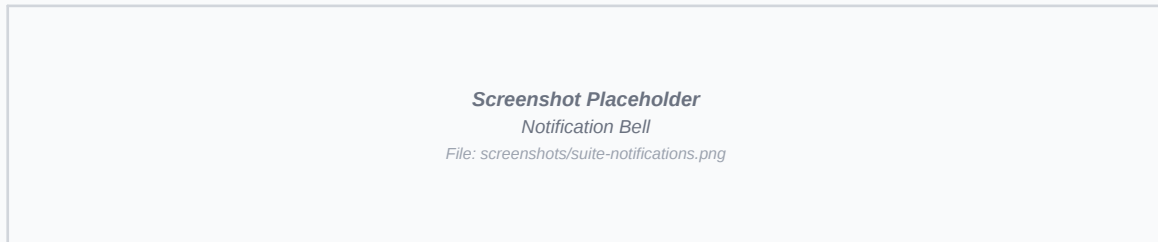
1. Click the **TraceBoard / TraceDocs / TraceTest** tabs in the navigation bar.
2. The active product is highlighted.
3. The switch is instant — there is no full page reload, and your session and active project are preserved.

Tips:

- Deep links (e.g., a link to a specific test run) open the correct product automatically.
- You can bookmark any product URL (`/board`, `/docs`, `/test`) for direct access.

8.3 Real-Time Notifications (WebSocket)

The suite keeps you informed about important events in real time through a **notification bell** in the navigation bar. Notifications are pushed to your browser over a WebSocket connection (authenticated with your session), so there is no need to refresh the page.



What triggers a notification:

- **Baseline approved** ■ — when a baseline is approved in TraceBoard, you are notified immediately (documents for that baseline are regenerated automatically in the background).
- **Test run completed** ■ — when a test run's results are finalized in TraceTest.
- **Audit narrative ready** ■ — when a background audit narrative finishes generating (Section 3.10). The toast deep-links back to the report.
- **Consistency check ready** ■ — when a background consistency check finishes (Section 3.9). The toast deep-links back to the report.

Using the bell:

- A badge shows the number of **unread** notifications.
- Click the bell to open a dropdown listing the most recent notifications (up to 20, kept across page reloads in your browser).
- Opening the dropdown marks everything as read; **Clear** removes the notification history.
- If the connection drops (for example, during a network blip), the shell reconnects automatically and continues delivering events.

9 Administration

9.1 User Management

User management is available to users with the **Admin** role.

Adding a user:

1. Go to **Settings** → **Users** in TraceBoard.
2. Click **Add User**.
3. Enter the email address, name, and select a role.

Removing a user:

1. In the Users list, find the user and click the delete action.
2. Confirm the deletion. The user will no longer be able to log in.

Roles:

Role	Permissions
Admin	Full access: manage users, configure SSO, manage projects, create baselines, view everything.
Member	Standard access: create/edit items, create baselines, generate documents, execute tests.
Viewer	Read-only access: view items, baselines, documents, and test results. Cannot modify data.

Roles apply across all three products. An Admin in TraceBoard is also an Admin in TraceDocs and TraceTest.

Settings

Data ManagementAI ImportMembersComplianceLicensePreferences

Project Members (2)

Name	Email	Role	Actions
Super Admin	admin@traceboard.local	Admin	Reset Remove
Demo User	demo@traceboard.local	Member	Reset Remove

Add Member

User

Role

Member

Add

9.2 SSO (OIDC)

The TraceBoard Suite supports Single Sign-On via OpenID Connect (OIDC). This allows users to log in with their existing identity provider.

Supported providers:

- Keycloak
- Azure AD
- Any OIDC-compliant provider (Auth0, Okta, Google, etc.)

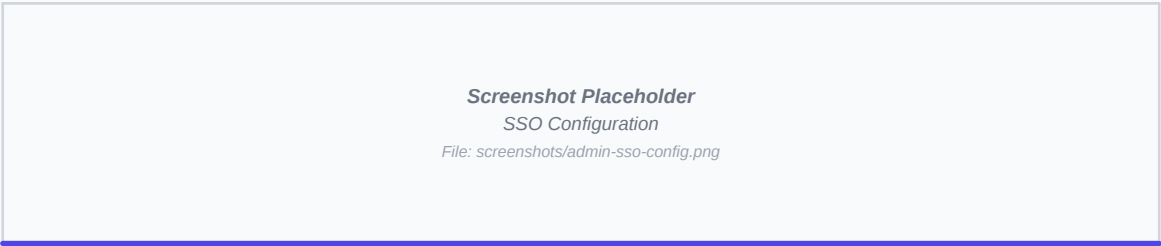
Configuration:

1. In TraceBoard, go to **Settings** → **SSO**.

2. Provide the following OIDC configuration details:

1. Click **Enable OIDC**.

2. Once configured, the login page shows an additional "Log in with SSO" button.



SSO applies across all three products. Users authenticated via SSO have their roles mapped from the identity provider's claims.

9.3 LLM Configuration

The AI features in TraceBoard — "Ask AI" chat (Section 3.7), Import Assistant (Section 3.11), Test Case Generation (Section 3.8), Audit Narrative (Section 3.10), and Consistency Check (Section 3.9) — rely on an LLM endpoint. **The default deployment uses a local Ollama instance**, so no data leaves your infrastructure. Cloud providers (DeepSeek, Mistral, OpenAI, Anthropic, Google Gemini) can be enabled by editing the `.env` file, but they are strictly opt-in.

Using the default local Ollama setup:

No configuration is required — the suite ships pre-configured to talk to a local Ollama instance at `http://localhost:11434` with model `llama3`. If Ollama is not yet installed:

1. Install Ollama from ollama.com.
2. Pull the default model:

```
bash
```

```
ollama pull llama3
```

```
`
```

1. Verify Ollama is running at `http://localhost:11434`.

Using a cloud LLM provider (opt-in):

Cloud providers like DeepSeek, Mistral, and OpenAI can be enabled by editing the `.env` file. Set the provider URL, API key, and model, then restart the backend. For example, for Mistral AI:

```
yaml
```

```
environment:
  LLM_API_URL: https://api.mistral.ai/v1
  LLM_API_KEY: your-api-key
  LLM_MODEL: minstral-14b-2512
  LLM_MAX_TOKENS: 32768
```

Provider behaviour is configuration-driven: request parameters are adapted automatically to the model family detected from the model name (DeepSeek `thinking mode`, OpenAI `reasoning_effort`, Mistral `safe_prompt/random_seed`, Claude `max_tokens`, etc.). Adding a new provider only requires updating the model-family configuration — no code changes.

Key environment variables:

Variable	Description
<code>OLLAMA_URL</code>	Base URL of the local Ollama server (default <code>http://localhost:11434</code>)
<code>LLM_API_URL</code>	Base URL of an OpenAI-compatible <code>/chat/completions</code> API. Leave empty to use the local Ollama setup
<code>LLM_API_KEY</code>	API key for cloud providers (leave empty for local Ollama)

Variable	Description
LLM_MODEL	Model name (default llama3 for Ollama)
LLM_MAX_TOKENS	Output-token budget for cloud calls — required for reasoning models that share the budget between chain-of-thought and the final answer (see Section 9.3)
LLM_DISABLE_THINKING	Global fallback for thinking/reasoning mode (1 = off, the shipped default). Per-feature toggles and defaults take precedence (see Section 3.7)
RQ_DEFAULT_TIMEOUT	Timeout (seconds) for background audit-narrative / consistency-check jobs (default 1800 = 30 minutes)

After configuration, restart the TraceBoard backend and the "Ask AI" button appears in the interface. See Section 9.3 for the full troubleshooting guide.

9.4 Backup and Restore

One-click backup from the UI (super-admin only):

1. Log in with a super-admin account.
2. Navigate to Settings, then Database Backup.
3. Click Backup Database.
4. The suite generates a full PostgreSQL dump of the entire database and downloads it automatically as `traceboard-suite-backup-YYYYMMDD-HHMMSS.sql`.
5. A progress indicator is shown while the dump is being generated. On success a confirmation message appears; on failure an error message is displayed.

The backup file is a plain SQL dump created with `pg_dump` containing all three product schemas. Store the file securely — it contains all project data for the suite.

Command-line backup (alternative):

```
bash
```

```
docker compose exec db pg_dump -U traceboard traceboard > backup.sql
```

This creates a full SQL dump of the entire suite database (all product schemas).

Restore from backup:

```
bash
```

```
docker compose exec -T db psql -U traceboard traceboard < backup.sql
```

Important notes:

- Stop the application services before restoring to avoid data conflicts.
- The backup includes all three product schemas (`traceboard.*`, `tracedocs.*`, `tracetest.*`).
- Test your backups regularly by restoring to a staging environment.

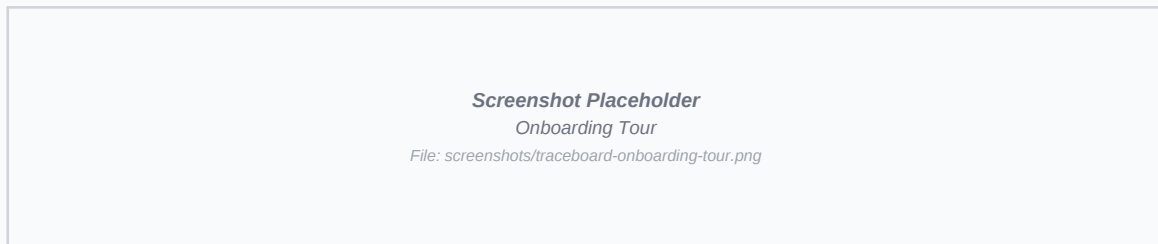
9.5 Onboarding Tour

TraceBoard Suite includes a guided onboarding tour that walks new users through the main features and interface.

Starting the tour:

1. The onboarding tour starts automatically when you log in for the first time.
2. To restart the tour at any time, go to **Settings** → **Help** and click **Start Onboarding Tour**.
3. The tour guides you step by step through:
 1. Each step highlights the relevant UI element and provides a short explanation. Click **Next** to advance, **Back** to revisit a step, or **Skip** to exit the tour.
 2. You can dismiss the tour at any time and restart it later from Settings.

Tip: The onboarding tour is a quick way to get team members productive. Point new users to this tour before they start working with the suite.



10 Troubleshooting

10.1 License Not Found or Expired

- Verify that `traceboard.license` is in the same directory as `docker-compose.yml`.
- Check the `expires_at` date in the license file — if it has passed, contact your sales representative.
- If the license file was recently updated, restart all services: `docker compose restart`.
- Check the logs for license-related errors: `docker compose logs traceboard-backend`.

10.2 TraceDocs or TraceTest Cannot Connect to TraceBoard

- Ensure TraceBoard is running and healthy: `docker compose ps` should show all services as up.
- Check that `TRACEBOARD_API_URL` is set correctly in the TraceDocs/TraceTest configuration.
- Check the logs: `docker compose logs tracedocs-backend` OR `docker compose logs tracetest-backend`.

10.3 LLM (AI) Connection Issues

If the "Ask AI" feature, Audit Narrative generation, or consistency checks are not responding, follow this structured guide.

Environment Variables (Required)

TraceBoard connects to an LLM via the OpenAI-compatible API. Ensure these variables are set in your `docker-compose.yml` (or `.env`):

Variable	Description	Example
LLM_API_URL	Base URL of the LLM API (must end with /v1 for OpenAI-compatible endpoints).	https://api.openai.com/v1, http://localhost:11434/v1 (Ollama), http://vllm-server:8000/v1
LLM_API_KEY	API key (or placeholder for local servers).	sk-... (OpenAI), ollama (Ollama), dummy (vLLM)
LLM_MODEL	Model name (must match what the provider expects).	gpt-4o, claude-3-5-sonnet, gemini-1.5-pro, deepseek-v4-flash, llama3.2
LLM_MAX_TOKENS	Maximum output tokens for cloud LLM calls. Required for reasoning models (e.g. DeepSeek deepseek-v4-flash/deepseek-v4-pro) whose chain-of-thought shares the output budget with the final answer – a too-small budget makes the API return 200 OK with an empty content.	32768
LLM_DISABLE_THINKING	Global default for thinking/reasoning mode on providers that support it (DeepSeek v4). Set 1 to disable thinking everywhere thinking is not explicitly chosen by the user. This is only a fallback: Audit Narrative and Consistency Check always run fast (thinking OFF) , while LLM Chat / Import Assistant / Test Case Generation expose a "Show reasoning" / "Detailed analysis" / "Detailed mode" toggle (default Detailed, i.e. thinking ON).	1 (shipped default)
AI_CAPABILITY_LEVEL	Optional override for feature availability (basic, medium, full, full+). Leave unset to auto-infer from the model name. See Section 2.3 for how it maps to license tiers.	auto

Step 1 – Verify the API Endpoint is Reachable

From within the TraceBoard container (or from the host if the container can reach it), test connectivity:

```
bash
```

```
# Test basic connectivity (no authentication)
curl -v http://localhost:11434/v1/models

# For cloud APIs, test with a simple request
curl https://api.openai.com/v1/models \
  -H "Authorization: Bearer sk-..."
```

If you get a connection timeout or refused, check:

- Network configuration (Docker networks, firewalls).
- For self-hosted LLMs (Ollama, vLLM), ensure the service is running and accessible from the TraceBoard container.
- For cloud APIs, verify your internet connection and any proxy settings.

Step 2 – Check API Key & Permissions

Cloud providers:

- **OpenAI:** `sk-...` – ensure the key is valid and has billing enabled.
- **Anthropic (Claude):** Use the OpenAI-compatible proxy (e.g., `https://api.anthropic.com/v1`). Requires an API key with appropriate model access.
- **Google Gemini:** Use the OpenAI-compatible endpoint (e.g., `https://generativelanguage.googleapis.com/v1beta/openai/`). Requires API key.
- **DeepSeek, Mistral, etc.:** Similar, ensure the key is active.

Self-hosted (Ollama, vLLM):

- **Ollama:** No API key required – use `ollama` or leave empty.
- **vLLM:** Usually no key required – use any string.

Test the key manually:

```
bash
```

```
curl https://api.openai.com/v1/chat/completions \
  -H "Authorization: Bearer sk-..." \
  -H "Content-Type: application/json" \
  -d '{"model": "gpt-4o", "messages": [{"role": "user", "content": "Hello"}]}'
```

If you get a 401/403, the key is invalid or lacks permissions.

Step 3 – Verify the Model Name

The `LLM_MODEL` must exactly match a model name supported by the provider.

- **Ollama:** `ollama list` to see installed models. Use the exact name (e.g., `llama3.2`).
- **vLLM:** The served model name is set at startup; check the server logs.
- **Cloud:** Use the exact model ID (e.g., `gpt-4o`, `claude-3-5-sonnet-20241022`, `gemini-1.5-pro`).

```
>
```

```
>
```

```
>
```

```
>
```

Step 4 – Check TraceBoard Logs

View the backend logs for LLM-related errors:

```
bash
```

```
docker compose logs traceboard-backend | grep -i "llm\|ai\|openai\|anthropic"
```

Common errors:

- `openai.APIError: 401 Unauthorized` – invalid API key.
- `openai.APIError: 404 Not Found` – wrong URL or model name.
- `openai.APITimeoutError` – network issues or server overload.
- `openai.RateLimitError` – quota exceeded (check billing).

Step 5 – Specific Scenarios

A) Ollama (local)

Ensure Ollama is running:

```
bash
```

```
curl http://localhost:11434/api/tags
```

Pull a model:

```
bash
```

```
ollama pull llama3.2
```

In `docker-compose.yml`, set:

```
yaml
```

```
LLM_API_URL=http://host.docker.internal:11434/v1
LLM_API_KEY=ollama
LLM_MODEL=llama3.2
```

B) vLLM (self-hosted, local network)

Ensure vLLM server is running:

```
bash
```

```
curl http://vllm-server:8000/v1/models
```

Set:

```
yaml
```

```
LLM_API_URL=http://vllm-server:8000/v1
LLM_API_KEY=dummy
LLM_MODEL=your-served-model
```

If vLLM is on a different host, use its IP/hostname.

C) Cloud Providers (OpenAI, Anthropic, Gemini)

OpenAI:

```
yaml
```

```
LLM_API_URL=https://api.openai.com/v1
LLM_API_KEY=sk-...
LLM_MODEL=gpt-4o
```

Anthropic (Claude) via OpenAI-compatible proxy (use the official proxy or a third-party adapter like LiteLLM):

yaml

```
LLM_API_URL=https://api.anthropic.com/v1
LLM_API_KEY=sk-ant-...
LLM_MODEL=claude-3-5-sonnet-20241022
```

Google Gemini (OpenAI-compatible):

yaml

```
LLM_API_URL=https://generativelanguage.googleapis.com/v1beta/openai
LLM_API_KEY=your-google-api-key
LLM_MODEL=gemini-1.5-pro
```

DeepSeek / Mistral / etc.: Use their respective OpenAI-compatible base URLs and model names.

Step 6 – Adjust AI Capability Level

If the LLM connection works but certain features (e.g., Audit Narrative) are not available, check `AI_CAPABILITY_LEVEL`.

- `basic` – only simple tasks.
- `medium` – most features.
- `full` – all features (default if unset).
- `full+` – experimental features.

Some models may not support all capabilities; test with `full`.

Step 7 – Restart the Backend

After any configuration change, restart:

bash

```
docker compose restart traceboard-backend
```

Still Not Working?

- Check the exact error message in the logs.
- Verify that your provider supports the `/chat/completions` endpoint (most do).
- If using a proxy or gateway (e.g., LiteLLM, Azure OpenAI), ensure the endpoint is correctly configured.
- For cloud providers, check billing and quota usage.

10.4 PDF Generation Fails

- Verify the selected baseline has requirements — an empty baseline produces an empty document (and may appear as a failure).
- Check TraceBoard API reachability from within the TraceDocs container:
- Large baselines with many requirements may take longer to generate. Wait and check the generation status in the History tab.
- Check the logs: `docker compose logs tracedocs-backend`.

10.5 Cannot Switch Products

- Ensure you are logged in. The Product Switcher requires an active session.
- If clicking a product tab does nothing, verify the product routes are reachable on the suite origin:
- Try navigating directly to the product URL in a new tab.
- If a product returns an error, check its backend is running: `docker compose ps`.

11 Glossary

Term	Definition
Baseline	A point-in-time snapshot of all items and relations in a project. Immutable once created. Used as the source for document generation and test runs.
Circular Reference	A trace cycle where items link back to themselves through a chain of relations (e.g., $A \rightarrow B \rightarrow C \rightarrow A$). Detected and flagged automatically.
Coverage	The percentage of requirements that have been verified by at least one passing test.
Dashboard	A user-customizable layout of live widgets (requirement status, test coverage, pass rate, recent activity, etc.), saved per project.
Document Diff	A side-by-side comparison of the generated documents of two baselines, highlighting added and removed lines.
Evidence	Files (screenshots, logs) attached to a test result as proof of execution.
Flow View	An interactive visual graph showing items as nodes and relations as edges.
Gap	A requirement that has no linked tests (no <code>verifies</code> relation).
Generation	The process of creating a document (PDF or Word) from a baseline and a template in TraceDocs.
Grid View	A compliance matrix showing requirements against tests in a table format.
Impact Analysis	A directed dependency graph showing everything an item affects (downstream) or depends on (upstream), used for change-impact assessment.
Item	A unit of trace data in TraceBoard. Can be a Requirement, Task, Bug, or Test.
Product Switcher	The tabs in the top navbar that let you switch between TraceBoard, TraceDocs, and TraceTest inside the single suite shell.

Term	Definition
Relation	A directed link between two items (e.g., <code>verifies</code> , <code>implements</code> , <code>parent</code> , <code>relates-to</code>).
Suite Shell	The single web application that hosts all three products and provides the shared navigation bar.
SRS	Software Requirements Specification — a compliance document listing all requirements.
Test Case	An item of type <code>test</code> in TraceBoard, linked to one or more requirements via <code>verifies</code> relations.
Test Run	A named execution session in TraceTest where test cases from a baseline are executed and results are recorded.
Trace	The network of items and relations that shows how requirements are implemented, verified, and decomposed.
Traceability	The mapping between requirements and test cases that shows which tests verify which requirements.
WebSocket Notification	A real-time alert pushed to the browser (via the shared Redis event bus) when an event such as a baseline approval or test-run completion occurs.

12 Support

- **Documentation:** See the Suite Architecture for technical details.
- **Product Specs:** TraceDocs Product Spec and TraceTest Product Spec for detailed feature descriptions.
- **Issue Tracker:** Report bugs or request features through the project's issue tracker.