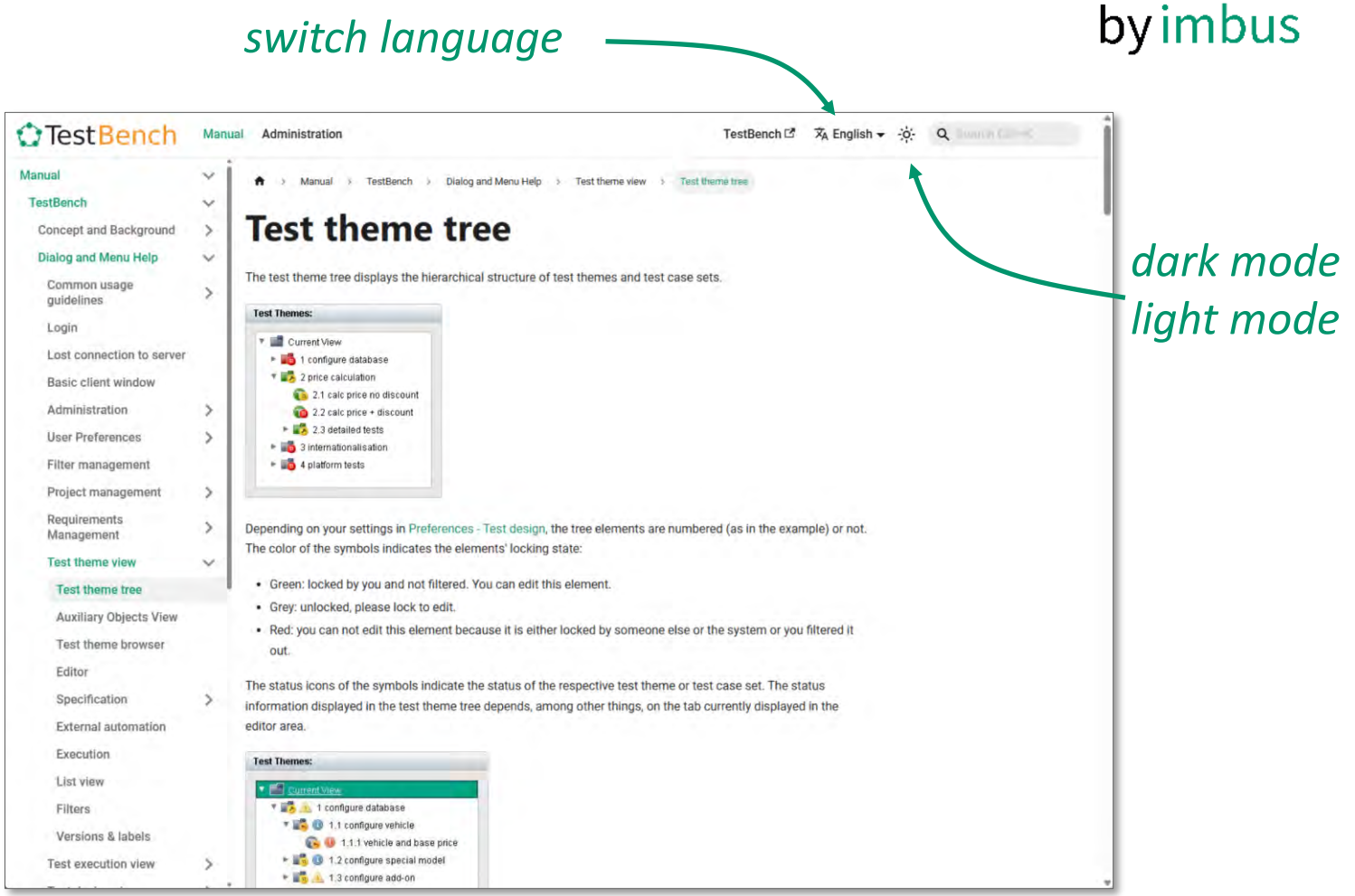
TestBench by imbus

New Features in Version 4.0

Patchlevel 4.0.39 BETA – 10.02.2026

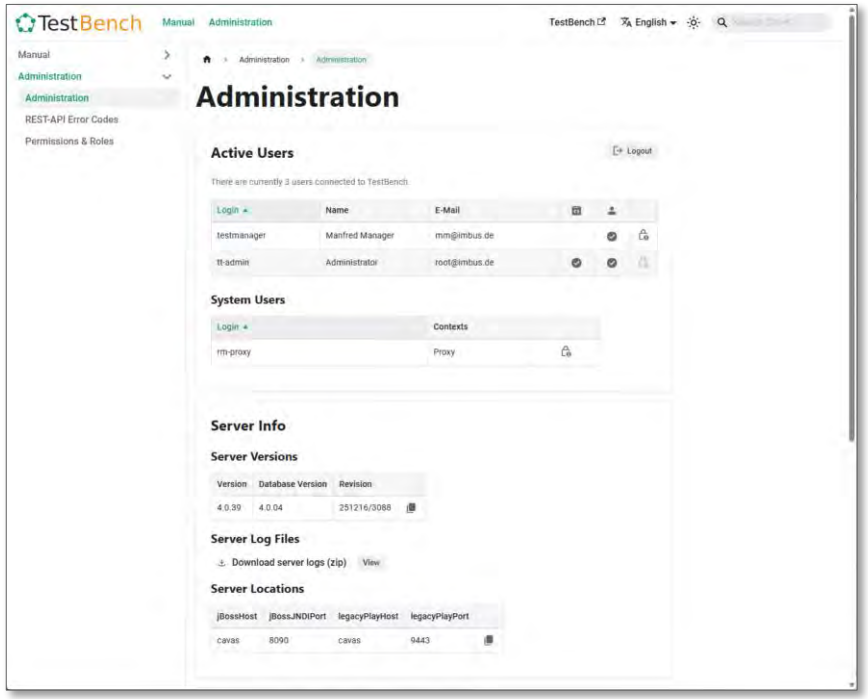
Online Help

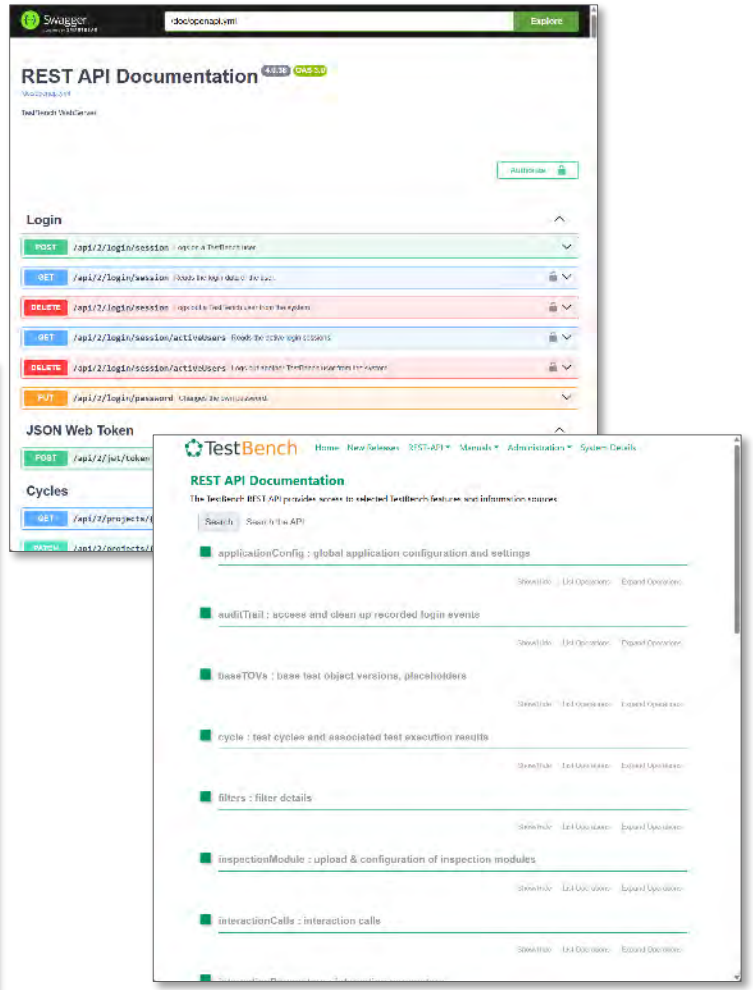
- The online help is now opened in the browser and can be opened and read independently of the TestBench client.
- New features
  - Switching languages within the help section
  - Light and dark mode
  - Comprehensive full-text search
  - Glossary



Administration Pages

- The administration pages in the backend contain important information:
  - Logged-in users, system accounts
  - Information about the server
    - Server log files
    - Server version info
    - License info
- REST API documentation
- Release notes



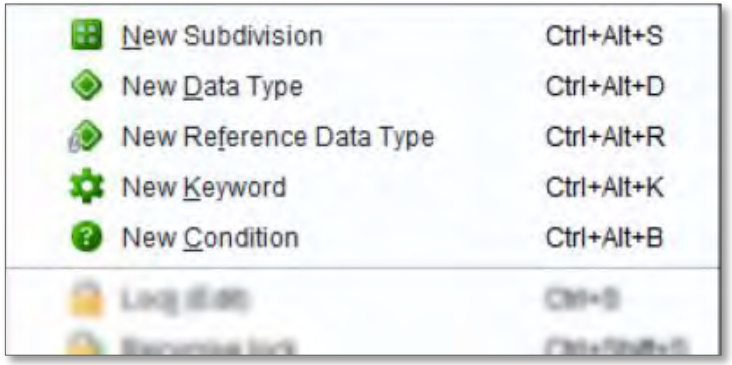


TestBench Terminology

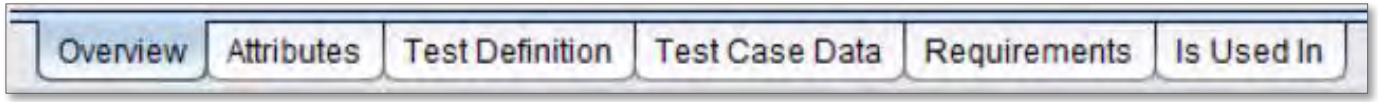
TestBench by imbus

Adaptation to the ISO 29119-5 standard Keyword Driven Testing

Previous term	New term
Interaction	Keyword
Keyword	Tag
Test Cases	Test Case Data
Definition	Test Case Definition



Detail of the context menu for test elements



Tab cards of a test case set

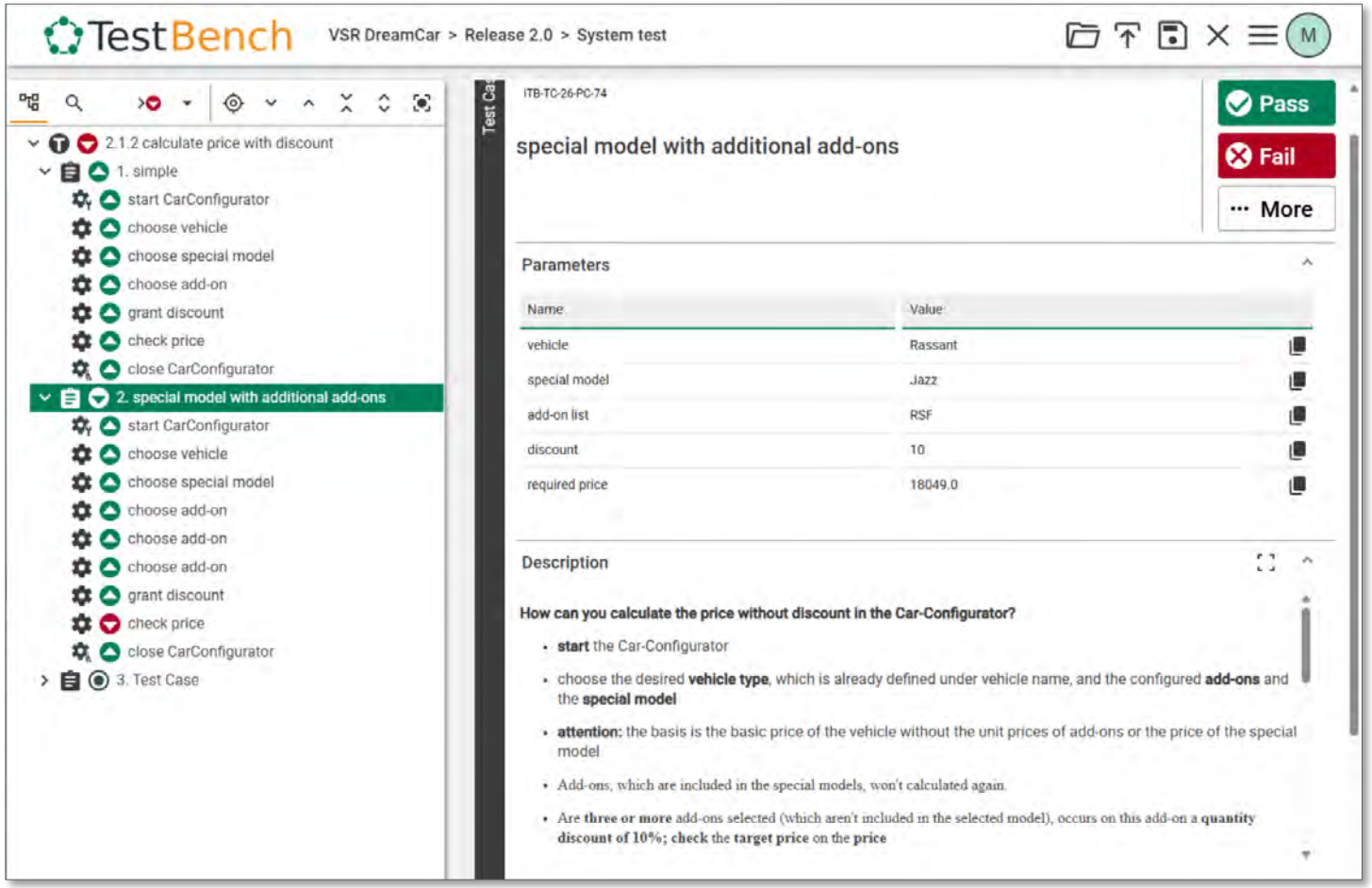
No changes have been made to technical export formats, e.g. the XML full report or project export/import, so backward compatibility is maintained.

Java iTORX becomes Web iTORX

TestBench by imbus

Existing iTORX becomes a Progressive Web App (PWA)

- Starts in the web browser
- Opens archives or is called from the client
- PWA = can be installed like an app via the browser, but still updates automatically



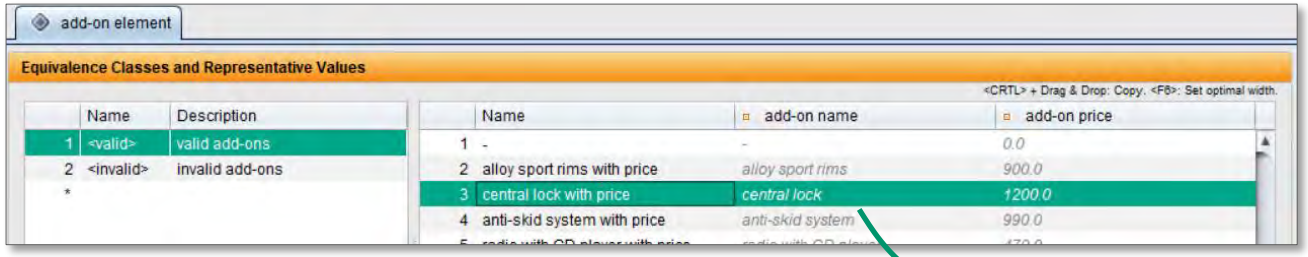
Compound data types in Web iTORX (1/2)

TestBench by imbus

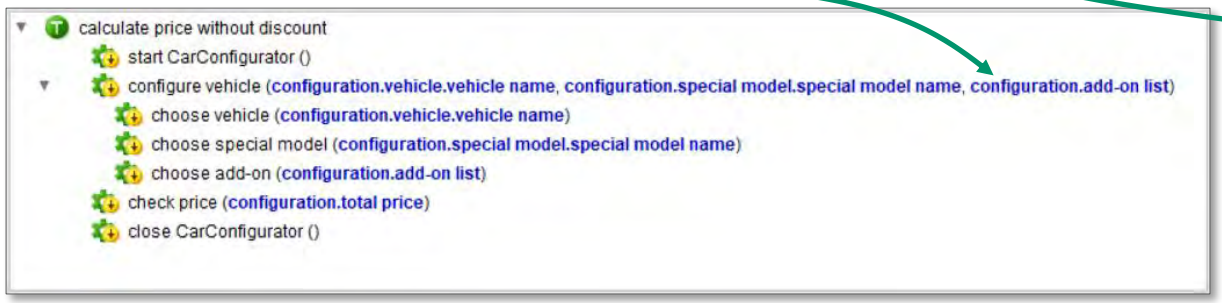
Values of compound representatives are displayed

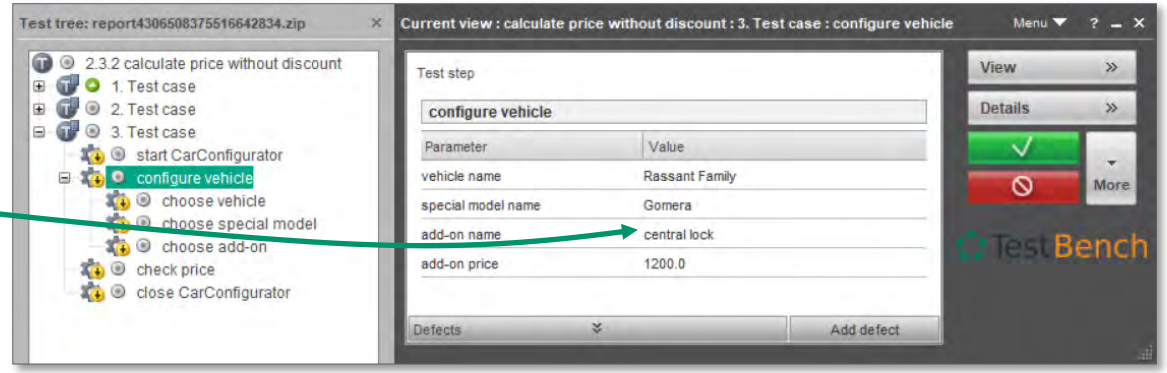
Initial situation :
Names of representatives of compound data types are not displayed at the higher level; instead, the detailed values of the sub-data types are displayed.

Compound data type in TestBench



Usage in test sequence





Detailed values in iTORX

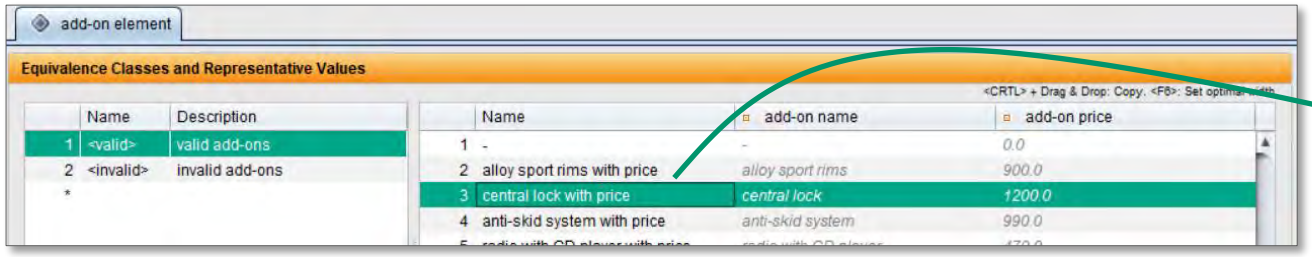
Compound data types in Web iTORX (2/2)

TestBench by imbus

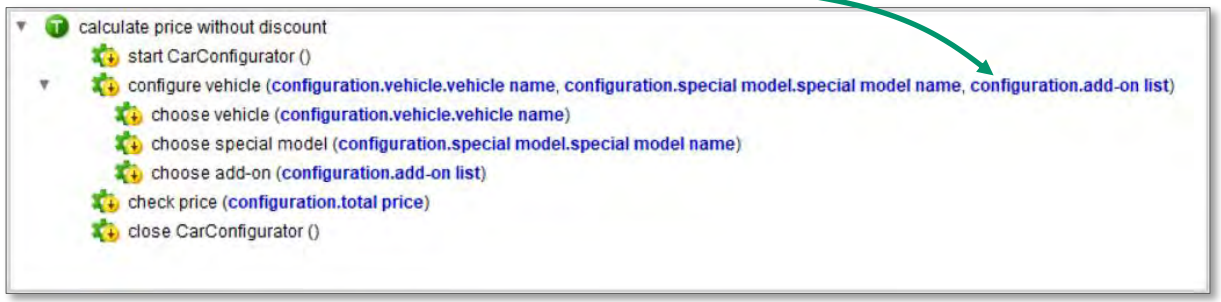
Values of compound representatives are displayed

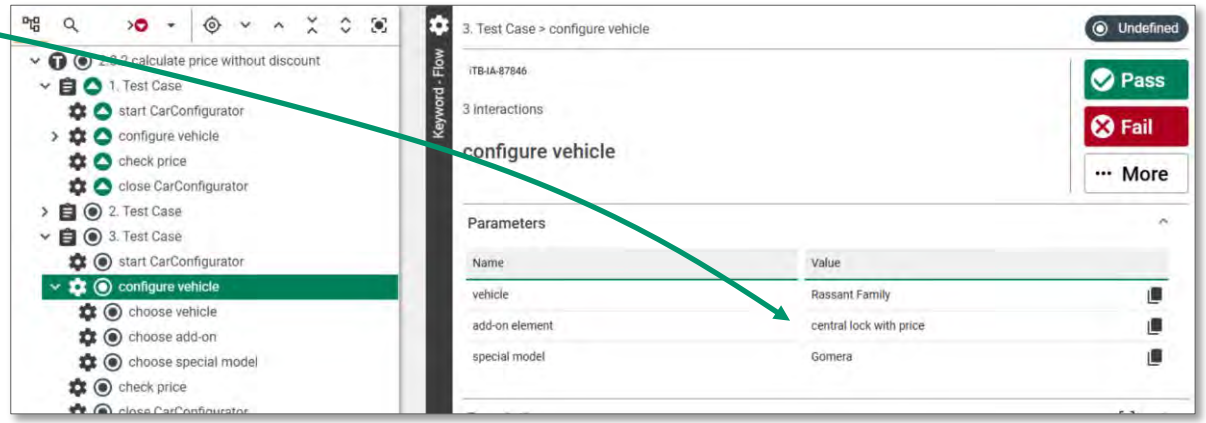
New:
Web iTORX displays the names of representatives of compound data types at the parent level.

Compound data type in TestBench



Usage in test sequence



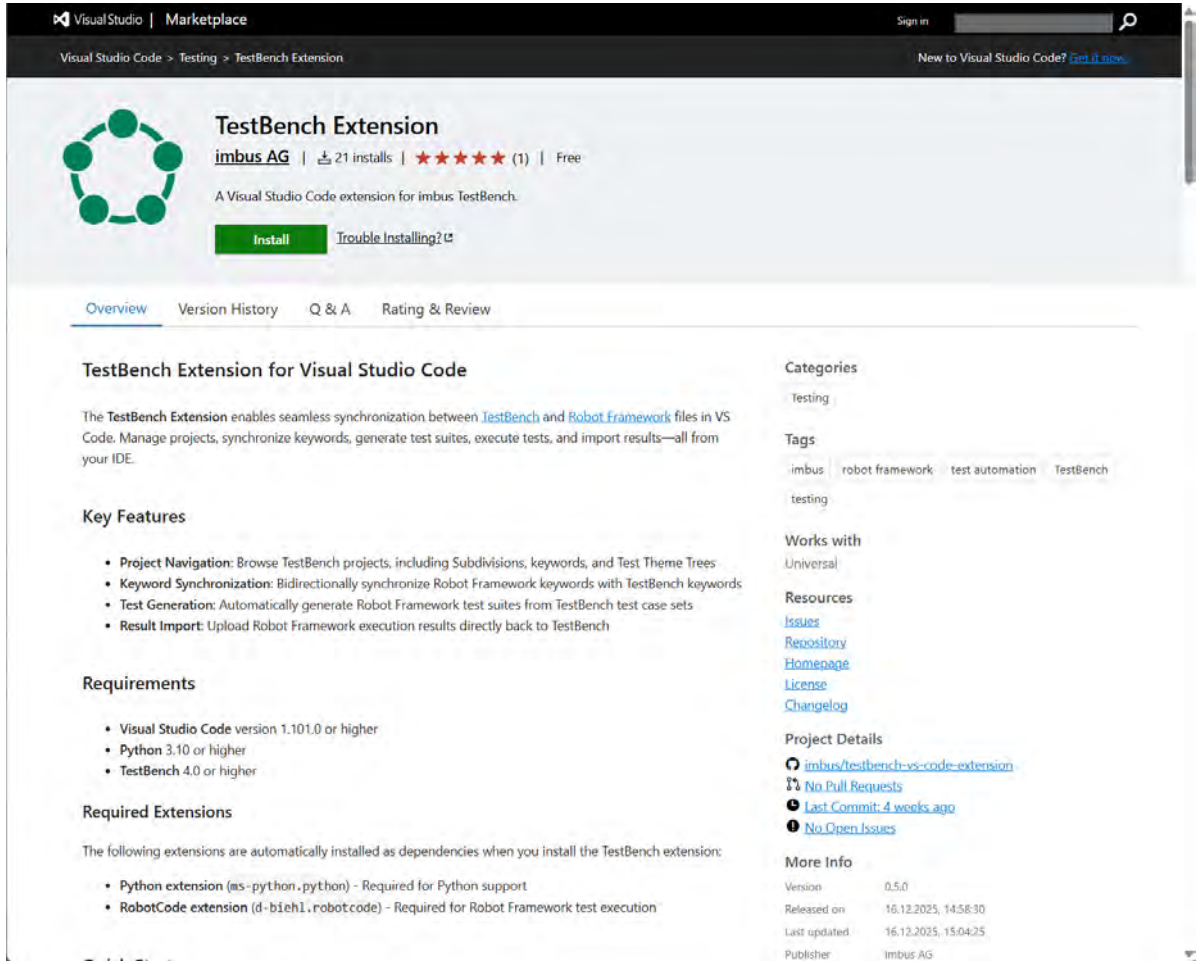


Name of representative

Visual Studio Code Extension

Specification = Automation

- Expert specifications for keyword-driven tests and their test data in TestBench
- Technical automation, coding and debugging of keywords in VS Code Extension
- Expert testers work exclusively in TestBench
- Test automation engineers work exclusively in VS Code.



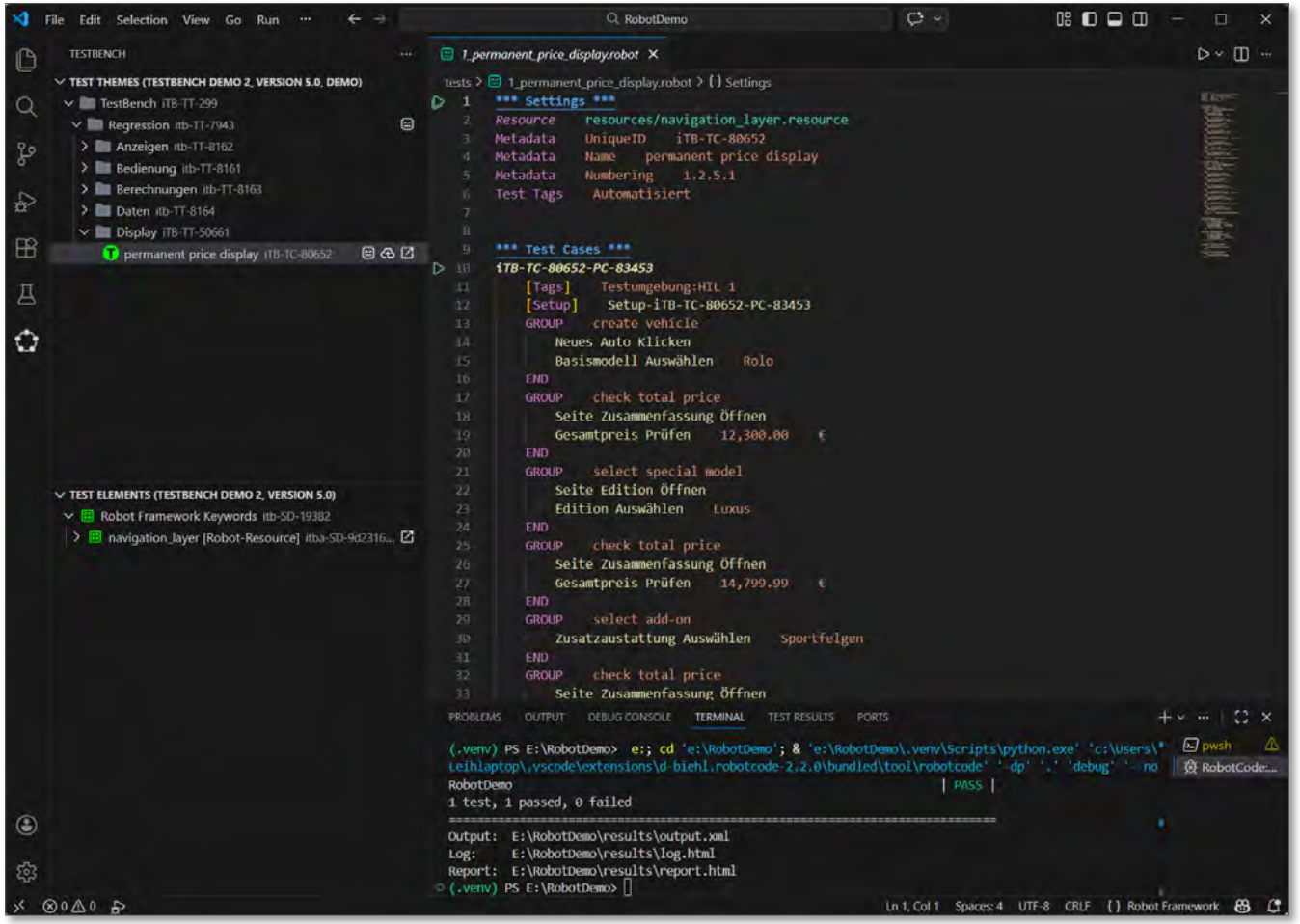
Visual Studio Marketplace:
https://marketplace.visualstudio.com/items?itemName=imbus.testbench-extension

Visual Studio Code Extension

TestBench by imbus

Integration of Robot Framework based test automation

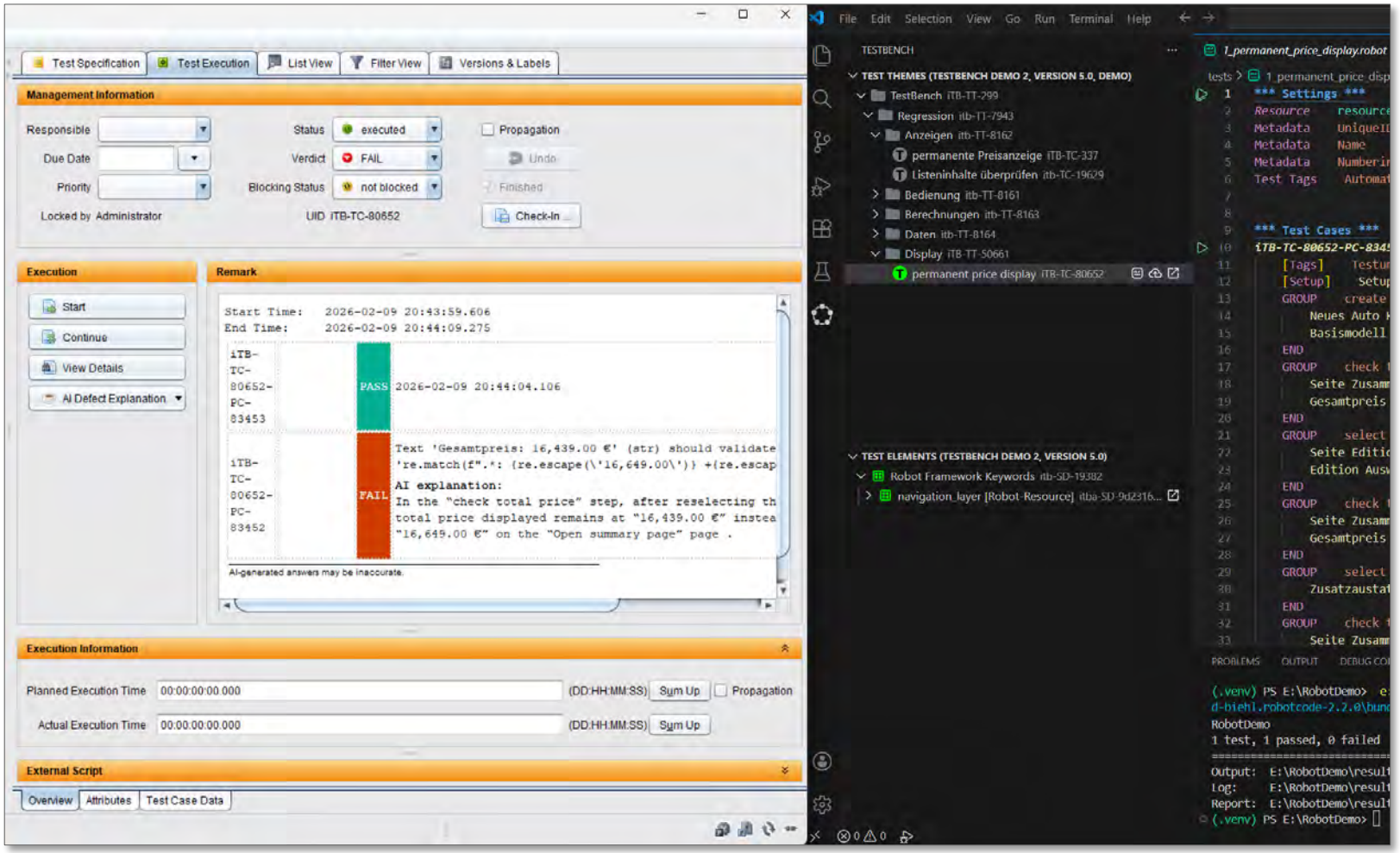
- TestBench tests are generated as Robot Framework tests.
- Keywords are implemented in VS Code
- New keywords are synchronised with TestBench (keywords can be created on both sides)
- Names and descriptions are synchronized with TestBench



VS Code Extension – Additional Functions

Save test results directly

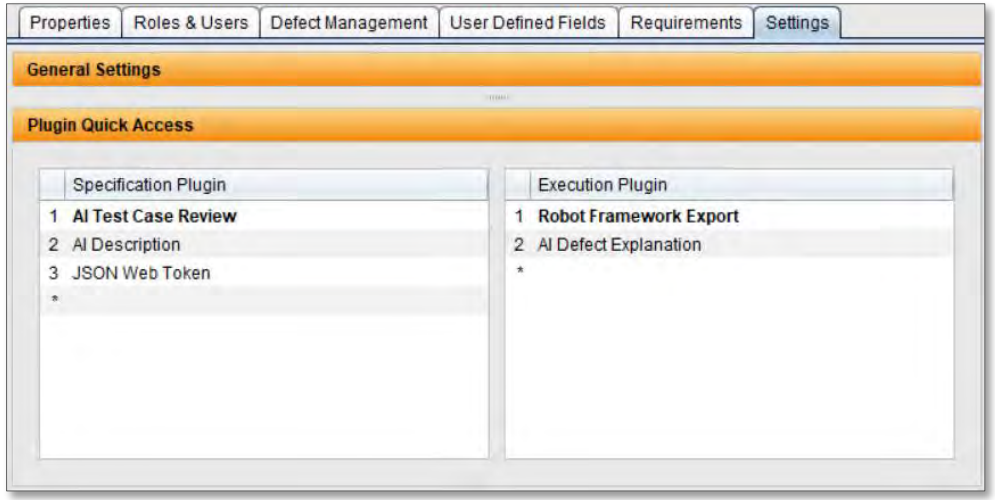
- Tests can be performed in VS Code(*)
- Test results can be saved directly back to TestBench



(*) with the help of free RobotCode Extension

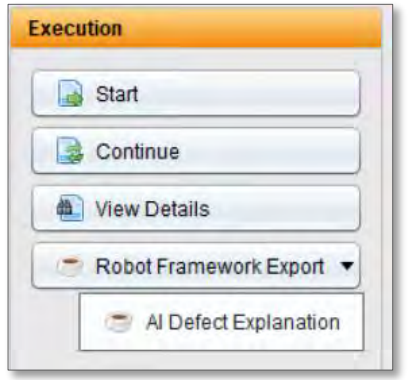
Plugin Quick Access

- The buttons for calling up plugins introduced in the last release have been expanded so that several plugins can be configured behind each button:

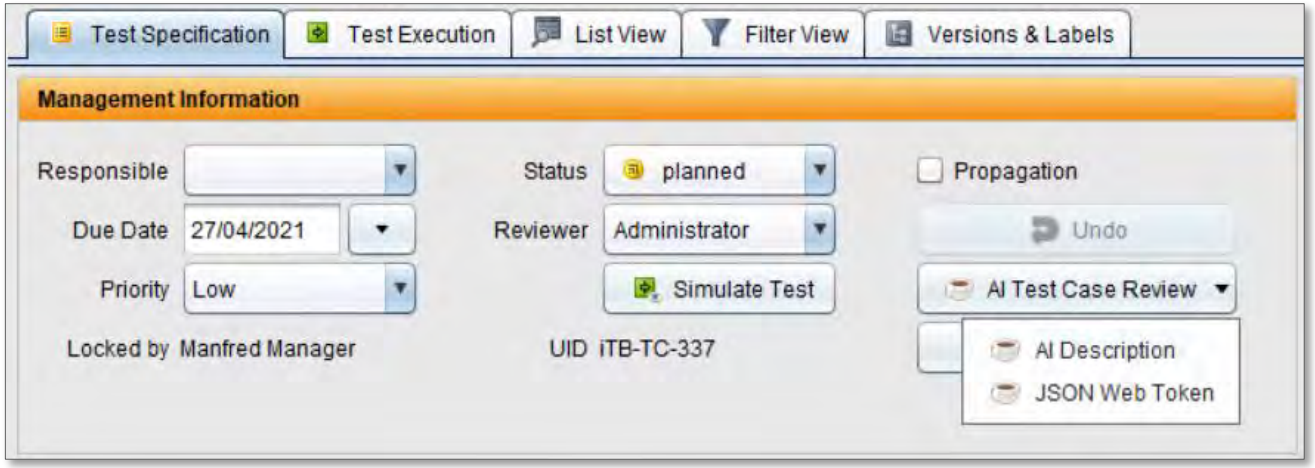


Definition of the buttons

Execution buttons



Specification buttons

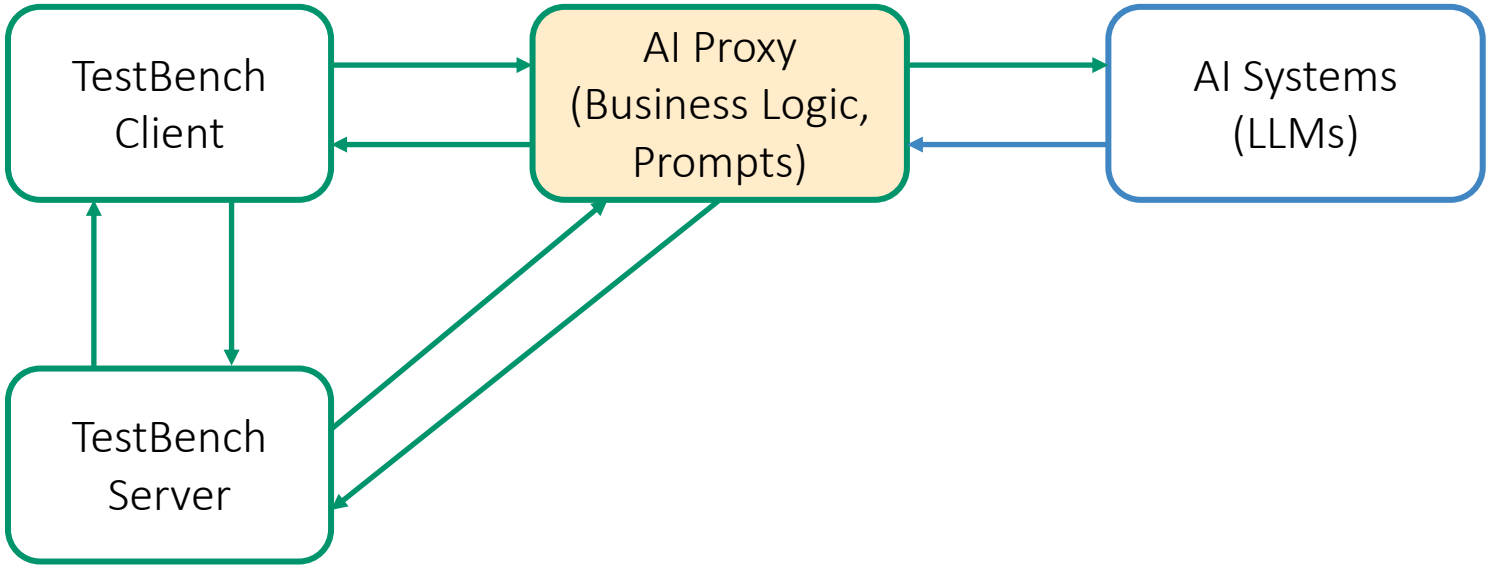


TestBench AI Assistants

Structure of the AI service(AI proxy)

Strong focus on flexibility and adaptability

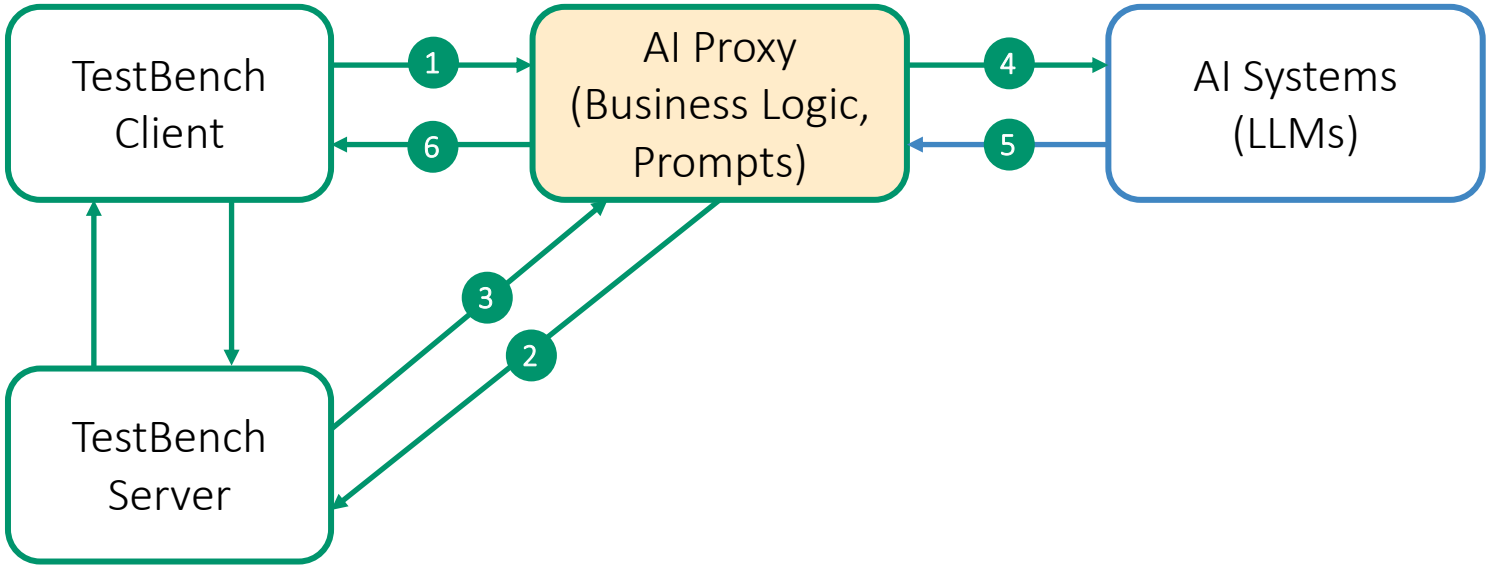
- Modular prompts can be customized and configured according to customer/project requirements
- open source
- Can be connected to various AI systems: local systems, private clouds, public clouds
- Standard:
  Public OpenAI ChatGPT



TestBench AI Assistants

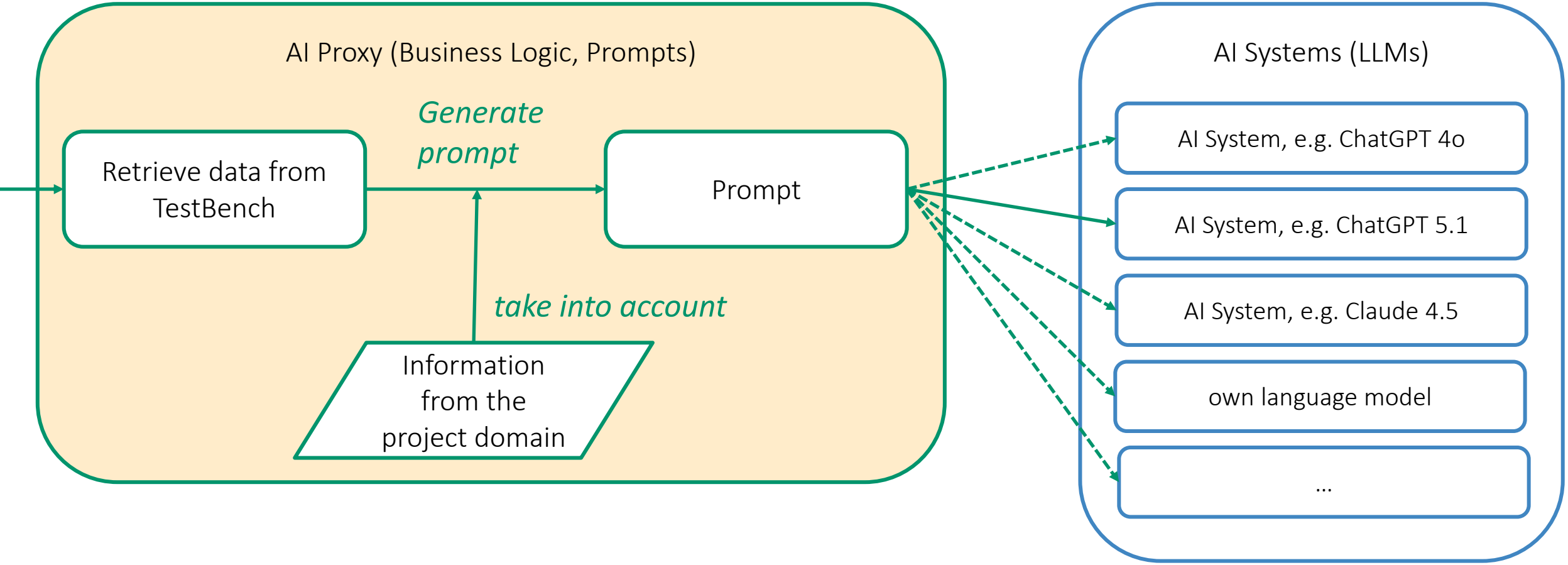
Internal flow

1. The client initiates the request for a specific use case to the AI bridge in the AI proxy and provides basic information, such as the ID of a test case set.
2. The AI Bridge retrieves the information required to generate prompts, etc., from the TestBench server.
3. The TestBench server delivers the requested data to the AI bridge.
4. The AI Bridge generates the request and prompt and sends them to the configured AI.
5. The AI bridge processes the AI's response.
6. The AI Bridge prepares the AI's response and sends the result to the client.



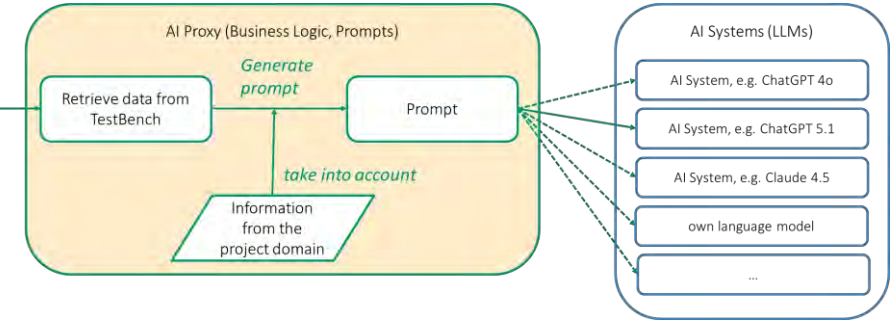
TestBench AI Assistants

Inner architecture



TestBench AI Assistants

AI functions (use cases) can be flexibly adapted and expanded



- Improving and more specialized language models are enabling more and more new AI solutions.
- Use cases each use the optimal(*) language model instead of all using the same one
- Integrating domain knowledge delivers better results
- Involvement of internal AI experts

(*) e.g. with regard to capabilities, licensing, location, data protection regulations

TestBench AI Assistants

Generation of the test case set description

- Generation of general description from test sequence and test case data
- Variants can be stored for each assistant:
  Here: Detailed and brief test case description

General Description

AI description - 2026-02-09 20:56:20

This test case set checks that the total price in CarConfig is continuously updated and displayed after each configuration step (create vehicle, select special model, select add-on).

Checks:
- After create vehicle, check total price is validated with total_price_after_vehicle.
- After select special model, check total price is validated with total_price_after_special_model.
- After the first select add-on, check total price is validated with total_price_after_add-on_1.
- After the second select add-on, check total price is validated with total_price_after_add-on_2.

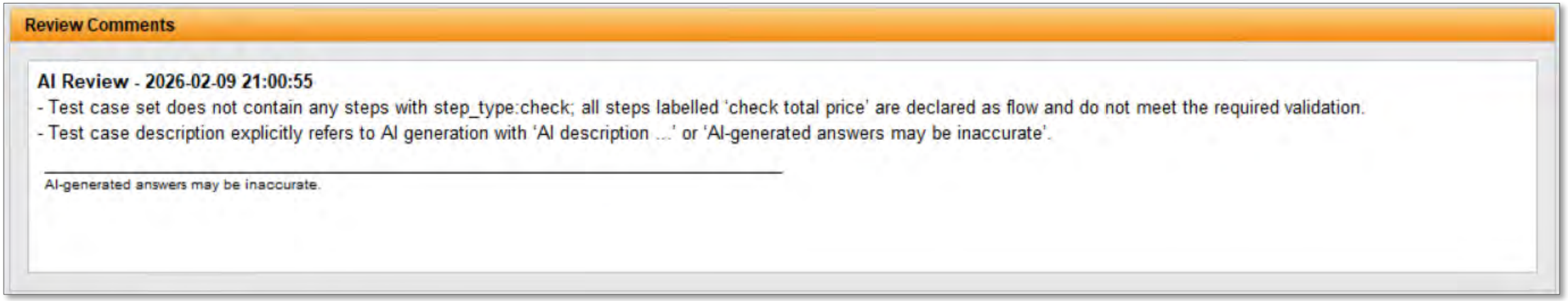
Test cases:
- iTB-TC-80652-PC-83453: vehicle=Rolo, special model=Luxury, add-on - 1=Sports rims, add-on - 2=Leather steering wheel; Expected prices: total_price_after_vehicle=12,300.00, total_price_after_special_model=14,799.99, total_price_after_add-on_1=15,699.99, total_price_after_add-on_2=15,699.99.
- iTB-TC-80652-PC-83452: vehicle=mini golf, special model=Gomera, add-on - 1=floor mats, add-on - 2=heated exterior mirrors; expected prices: total_price_after_vehicle=15,000.00, total_price_after_special_model=16,413.00, total_price_after_add-on_1=16,439.00, total_price_after_add-on_2=16,649.00.

AI-generated answers may be inaccurate.

TestBench AI Assistants

Review of test sequences

- Review of test case sets to determine whether they correspond to the implementation and comply with a defined set of rules.
- Rules can be defined in the assistant's prompting.

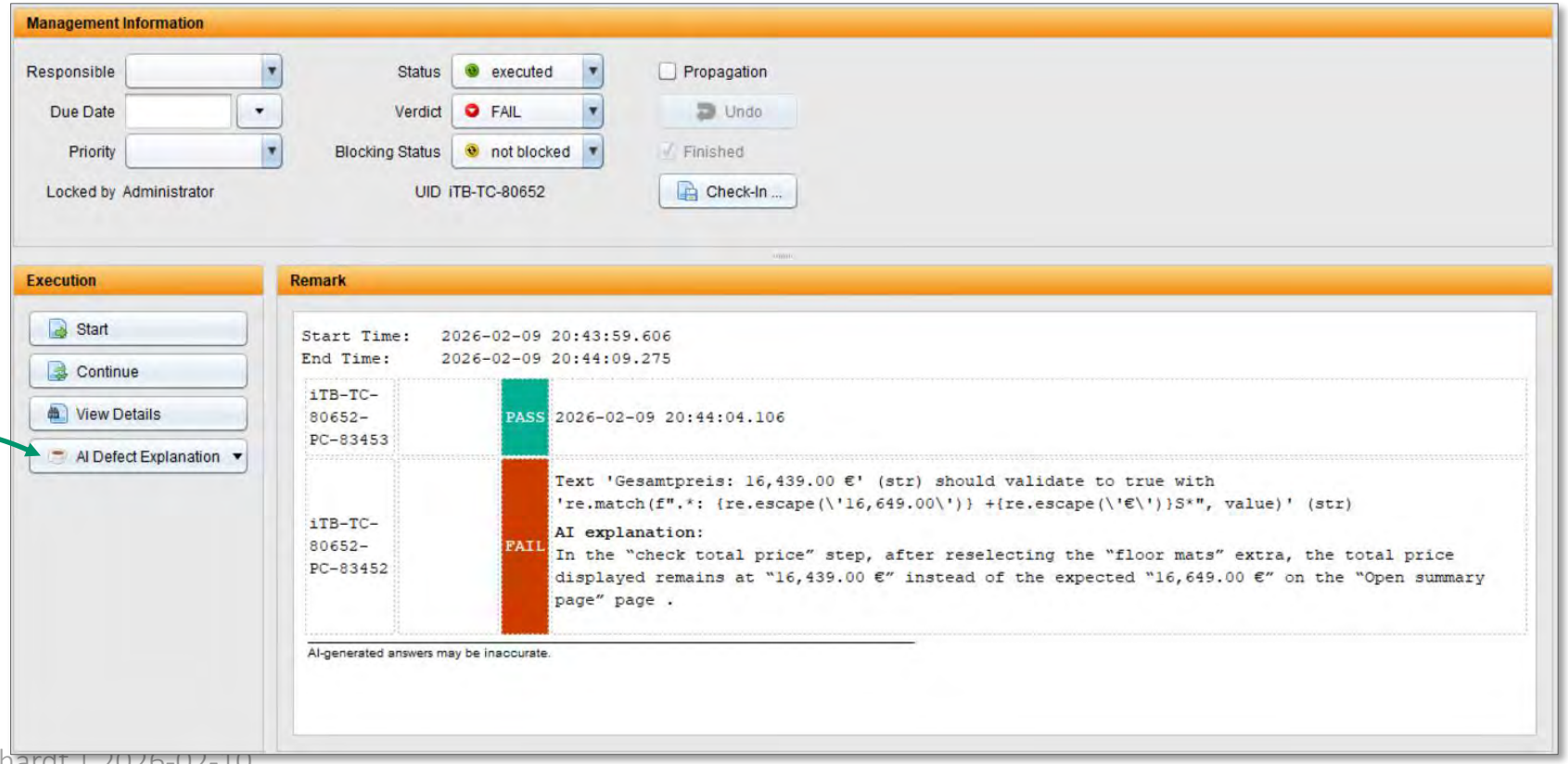


TestBench AI Assistants

Explanation of error messages from automation

- Translation of the technical details of an error message into “domain specific” language
- The original error message remains unchanged.

Assistants are called up via plugins



Requirements Management Integration

TestBench by imbus

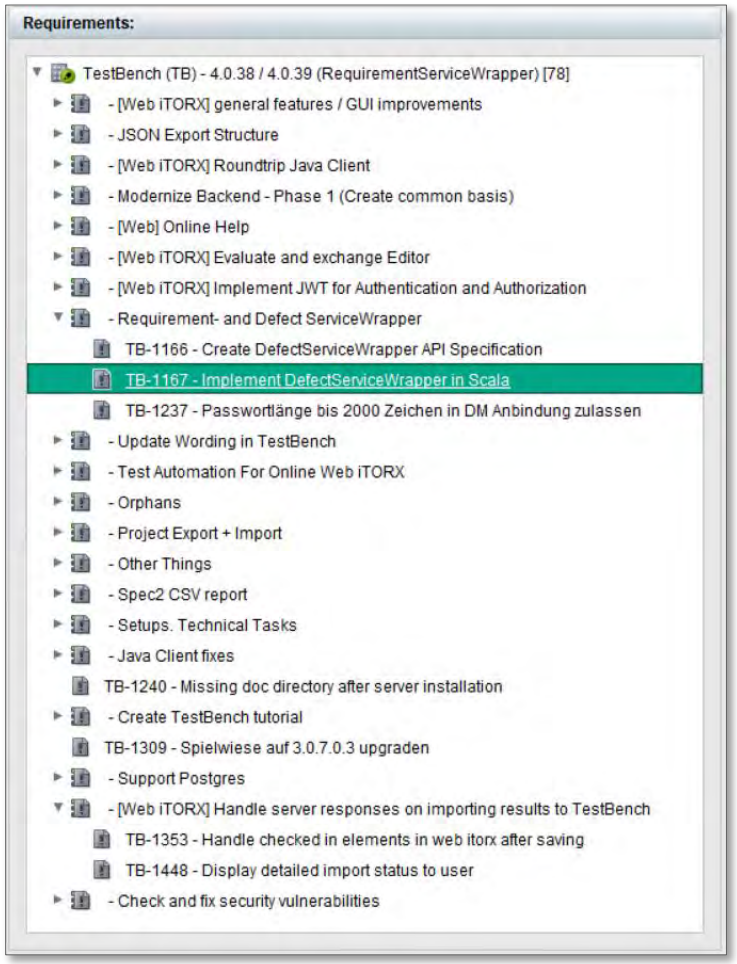
Modernization of the integration of requirements management systems

- New REST API for requirements management proxy
- Initial implementations: XLSX files, JSON line files, Atlassian Jira
- New features :
  - takeover of requirement trees, e.g. for epics and their user stories
  - change histories
- Open source implementation

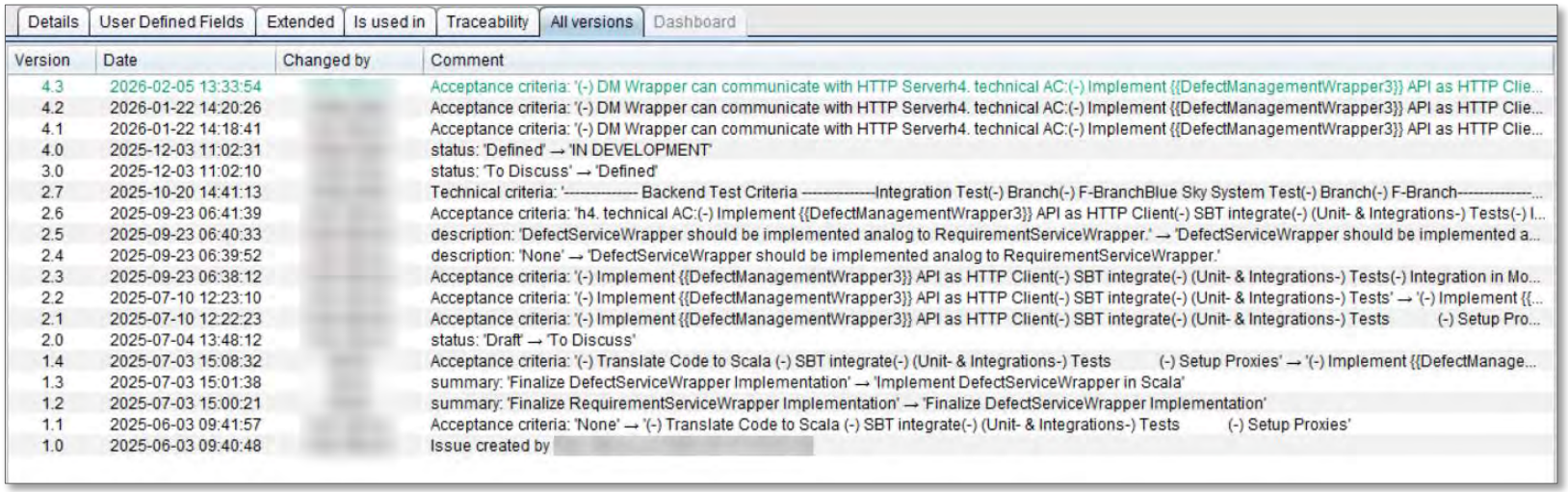
- Can also be used with TestBench version 3.0.x.

Requirements Management Integration

Example: Atlassian Jira



Requirements tree with epics, their user stories, tasks and bugs



Change history of a user story in Jira

Defect Management Integration

TestBench by imbus

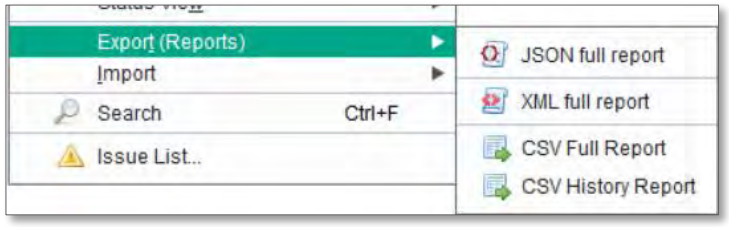
Modernization of the integration of defect management systems

- New REST API for defect management proxy
- Initial implementations: XLSX files, JSON line files, Atlassian Jira
- New features :
  - Creating bugs from TestBench (also for Jira)
  - Synchronization of all custom fields (= user defined fields in TestBench)
- Open source implementation

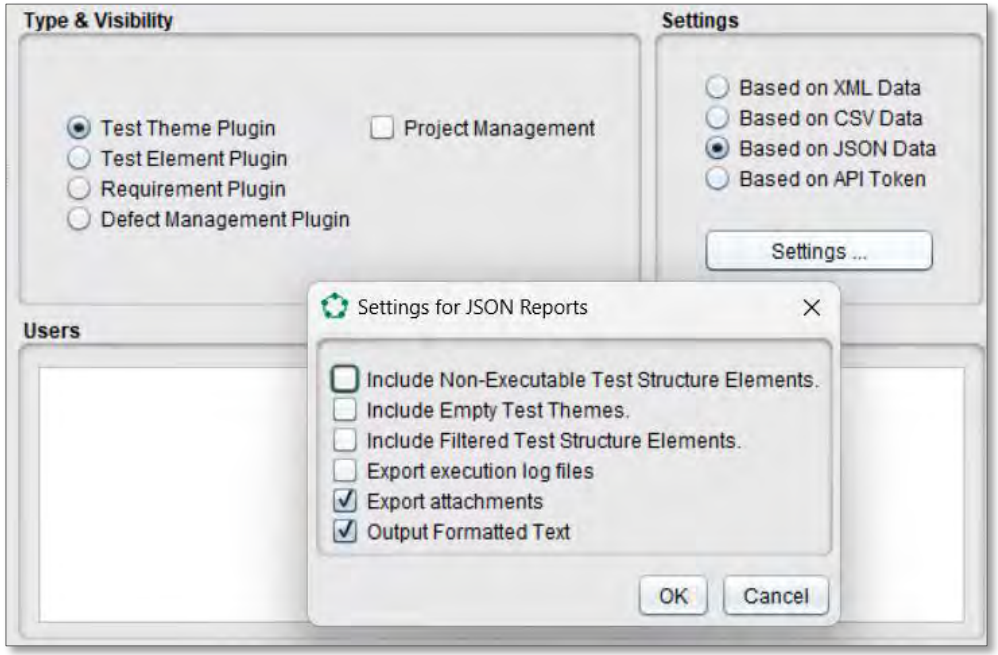
- Can also be used with TestBench version 3.0.x.

JSON as a New Export/Import Format

- JSON is another export and import format for TestBench.
- Web iTORX and other new integrations use this format.
- No log files are generated during the test run; instead, the test result data is added to the report.
- TestBench therefore provides a new JSON-based plugin type based on a JSON full report.



JSON full report in the report menu



Plugin definition based on JSON data

New Authentication via JSON Web Token

TestBench by imbus

- A JSON Web Token (JWT) is an access token standardised according to RFC 7519
- General structure: xxxxx.yyyyy.zzzzz
  - Header xxxxx: **Metadata**, such as signature algorithm and token type
  - Payload yyyyy: **Operational data** (known as claims), such as resource, user, target system, expiry date, etc.
  - Signature zzzzz: **Signature** over header and payload, generated with secret or private key
- Signature prevents manipulation

A JWT is a signed but unencrypted data packet consisting of a header, payload and signature that securely transmits identity claims.

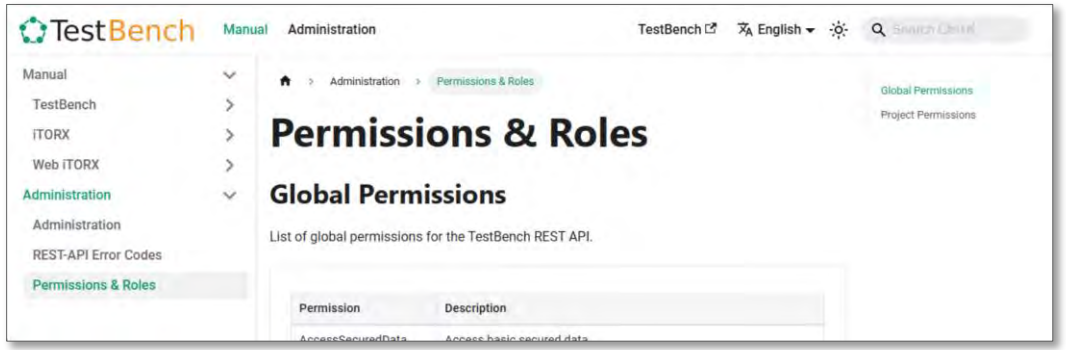
Benefits of JSON Web Token

- Enables secure data exchange between TestBench and other systems.
- Contains **all important information** for accessing data on the TestBench server.
- Determines who can access which data and for how long.
- No exchange of authentication data, such as login and password, is necessary.

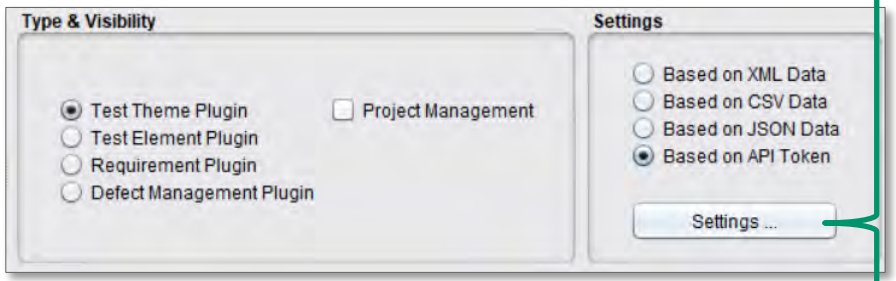
- Web iTORX is connected to TestBench via JWT.
- Web iTORX can be accessed via a URL that incorporates the JWT and loads the data stored in the token from the TestBench server.
  This allows integration into systems such as Jira or Jenkins via a link with the corresponding URL.

Plugin for JSON Web Token

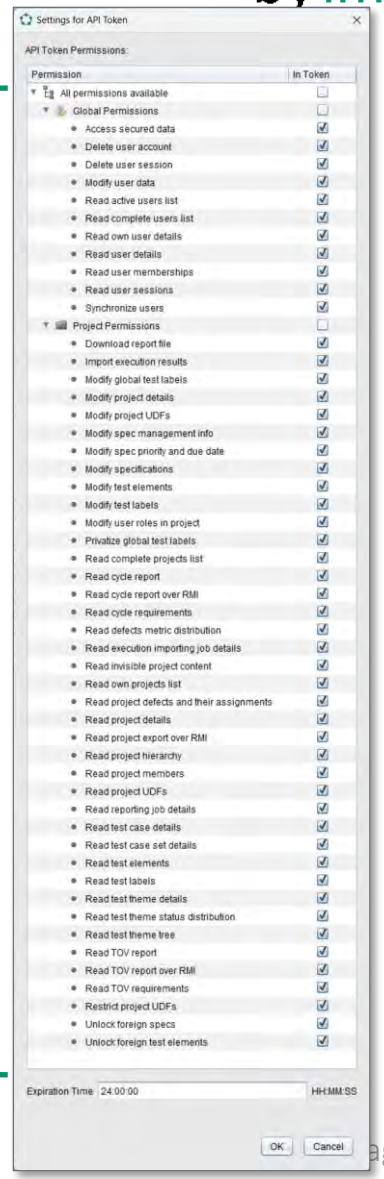
- To enable the use of JSON Web Tokens, TestBench provides a new plugin type. : **Based on API-Token**
- The plugin definition specifies which rights the token passes on.
- Connection to external systems can be established without sharing highly privileged access data.



Description of possible permissions and roles

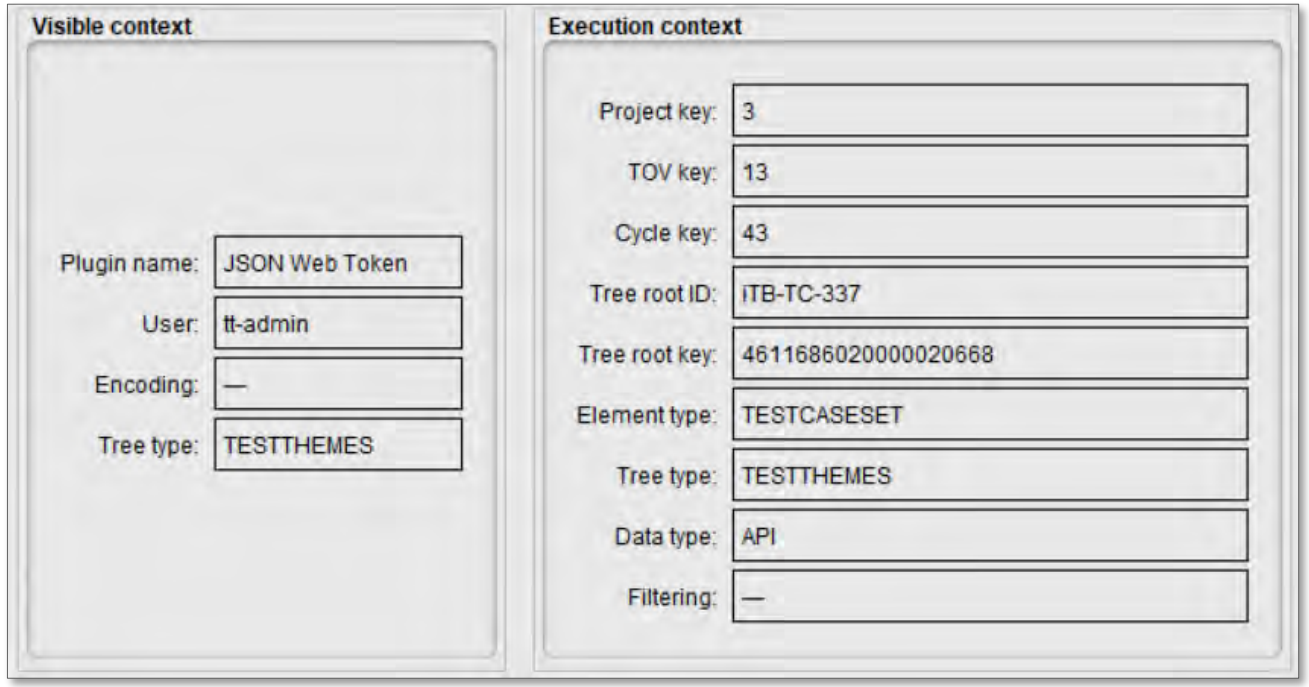


Plugin definition based on an API token with a list of possible permissions



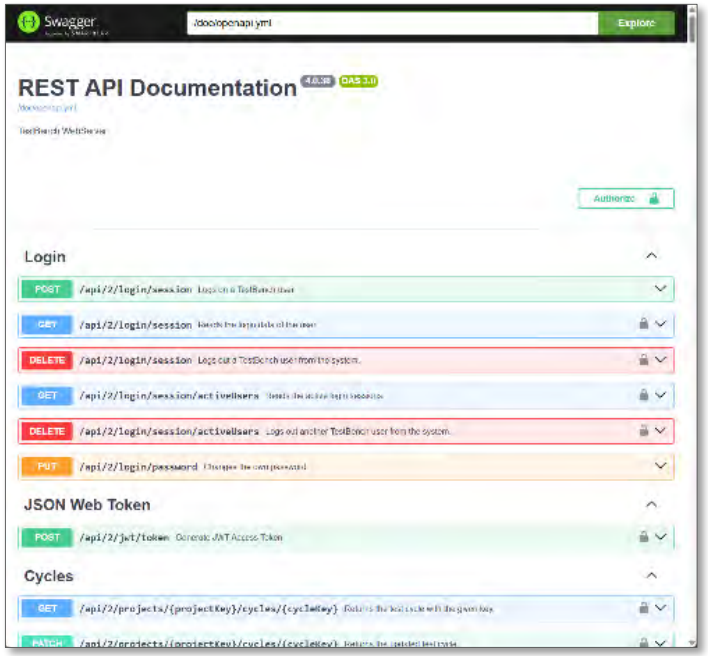
Plugins Know the Call Context

- The interface of all plugins has been expanded to include information describing the context of the call., e.g.:
  - Name and type of plugin
  - Key of test object version and test cycle
  - ID of the node from which the call is made

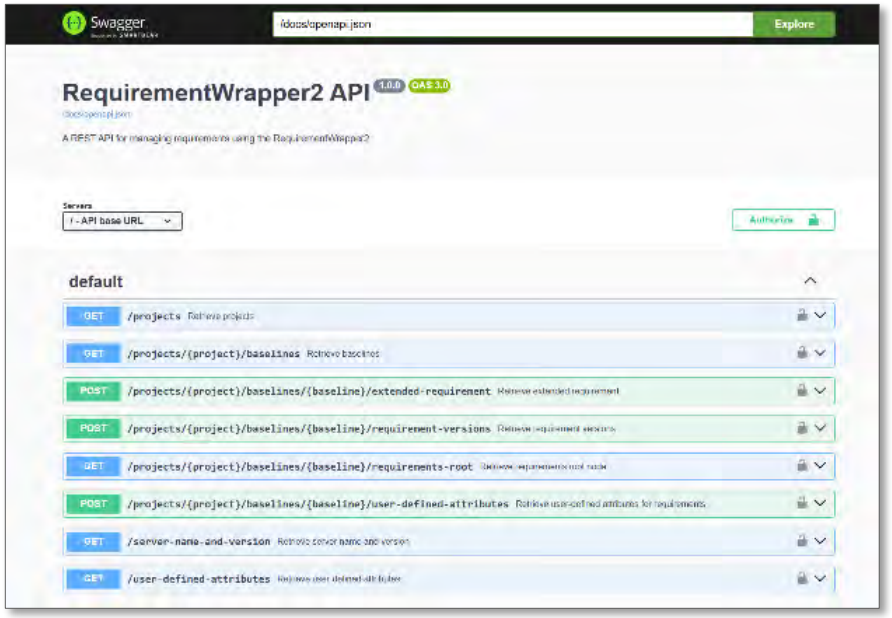


Extended REST-API

- The application server layer now consists of three servers. The third, new server provides a greatly expanded REST API.
- The new services for connecting requirements and defect management systems and the AI proxy also provide new REST APIs.



REST API of the new server



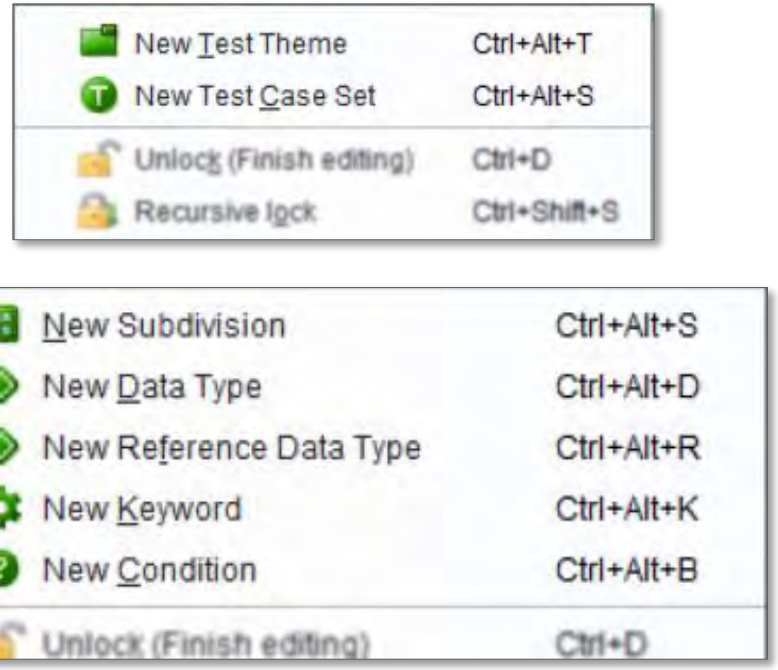
REST API of the requirements management proxy

Show Password in the Login Window

In the login screen, the password can be temporarily displayed by clicking on the eye icon in the password field:

Keyboard Shortcuts for creating elements

Test structure elements(*) can now also be created using keyboard shortcuts:



(*) Test structure elements are test subjects, test case sets, subdivisions, keywords, data types and conditions.

Time for Questions!

CONTACT

TestBench by imbus

imbus AG

Kleinseebacher Str. 9
91096 Möhrendorf
GERMANY

info@testbench.com

www.testbench.com

+49 9131 7518-0