Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024

BASIL The FuSa Spice to spot gaps in kernel testing and more

How to track existing LTP tests for a set of syscalls to their man pages and spot gaps?
How can the community be notified by a change to the kernel source code or to the man page?
How to provide a Test environment that is integrated in an automated CI workflow?
We'll discuss how BASIL can answers some of these questions.

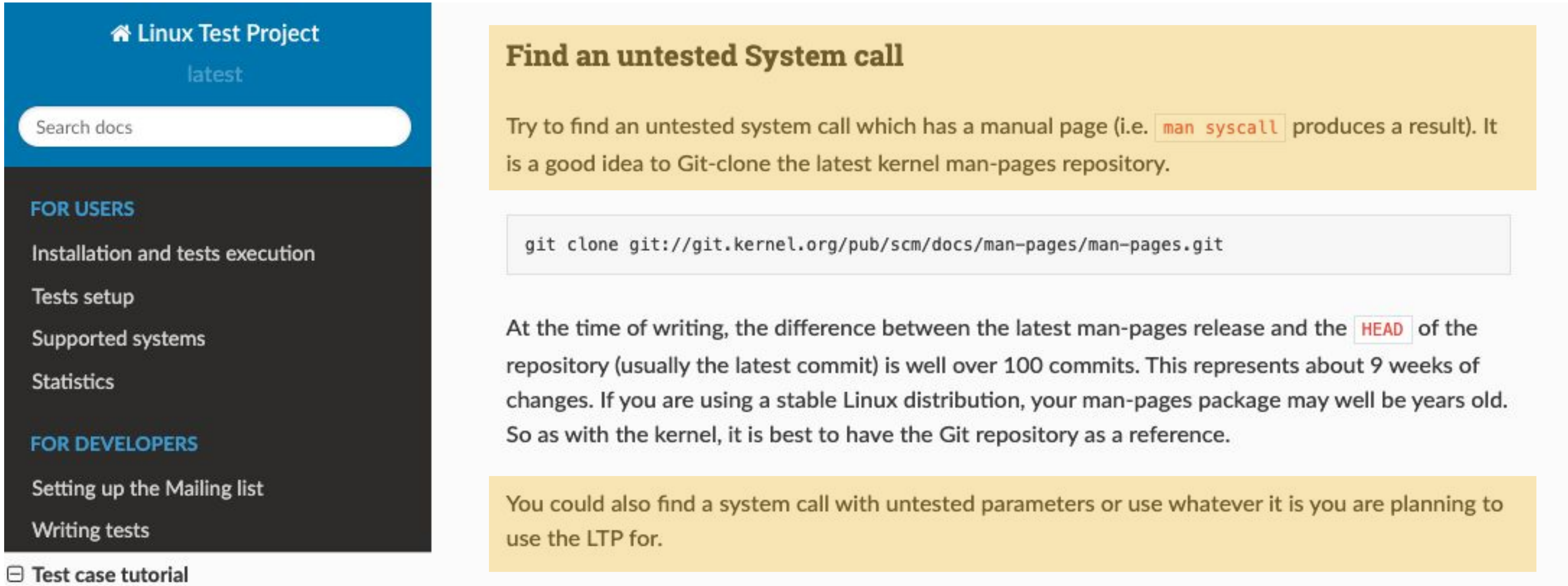
Luigi Pellecchia
Principal
Software Quality Engineer
Quality Engineering
In-vehicle OS

LINUX PLUMBERS CONFERENCE | Vienna, Austria
Sept. 18-20, 2024

BASIL The FuSa Spice

An Open Source Tool for Quality Management
Development driven by the ELISA project
Web Application that can be deployed via docker containers
Supports user management, user roles and permissions at sw component level.
Its main purpose is to establish a traceability between work items such as Software Specification, Software Requirements, Test Specifications, Test Cases, Test Results, Bugs, Assumption of Use, Source Code and so on.
It comes also with a Test Execution Environment that can be used with any test suite to run test against containers, virtual machines or physical hardwares.

LTP Documentation



Find an untested System call

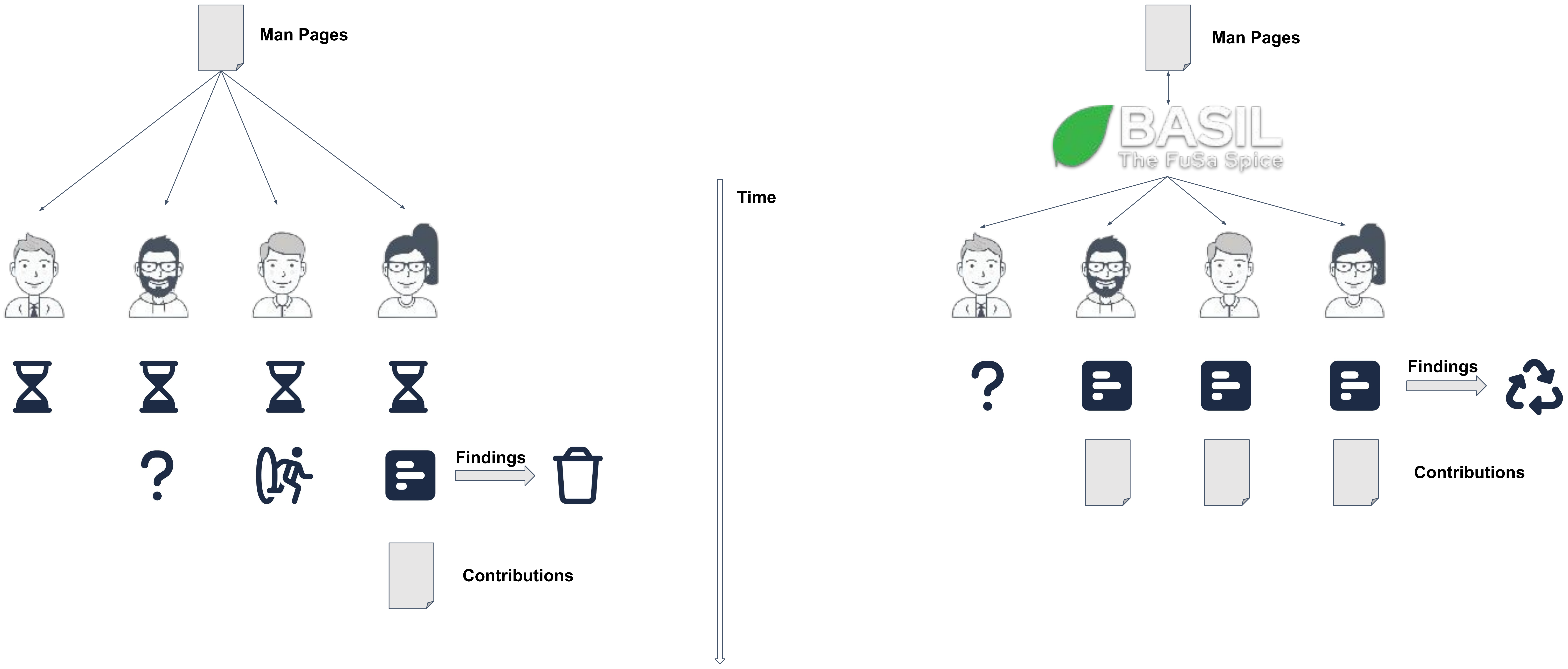
Try to find an untested system call which has a manual page (i.e. `man syscall` produces a result). It is a good idea to Git-clone the latest kernel man-pages repository.

```
git clone git://git.kernel.org/pub/scm/docs/man-pages/man-pages.git
```

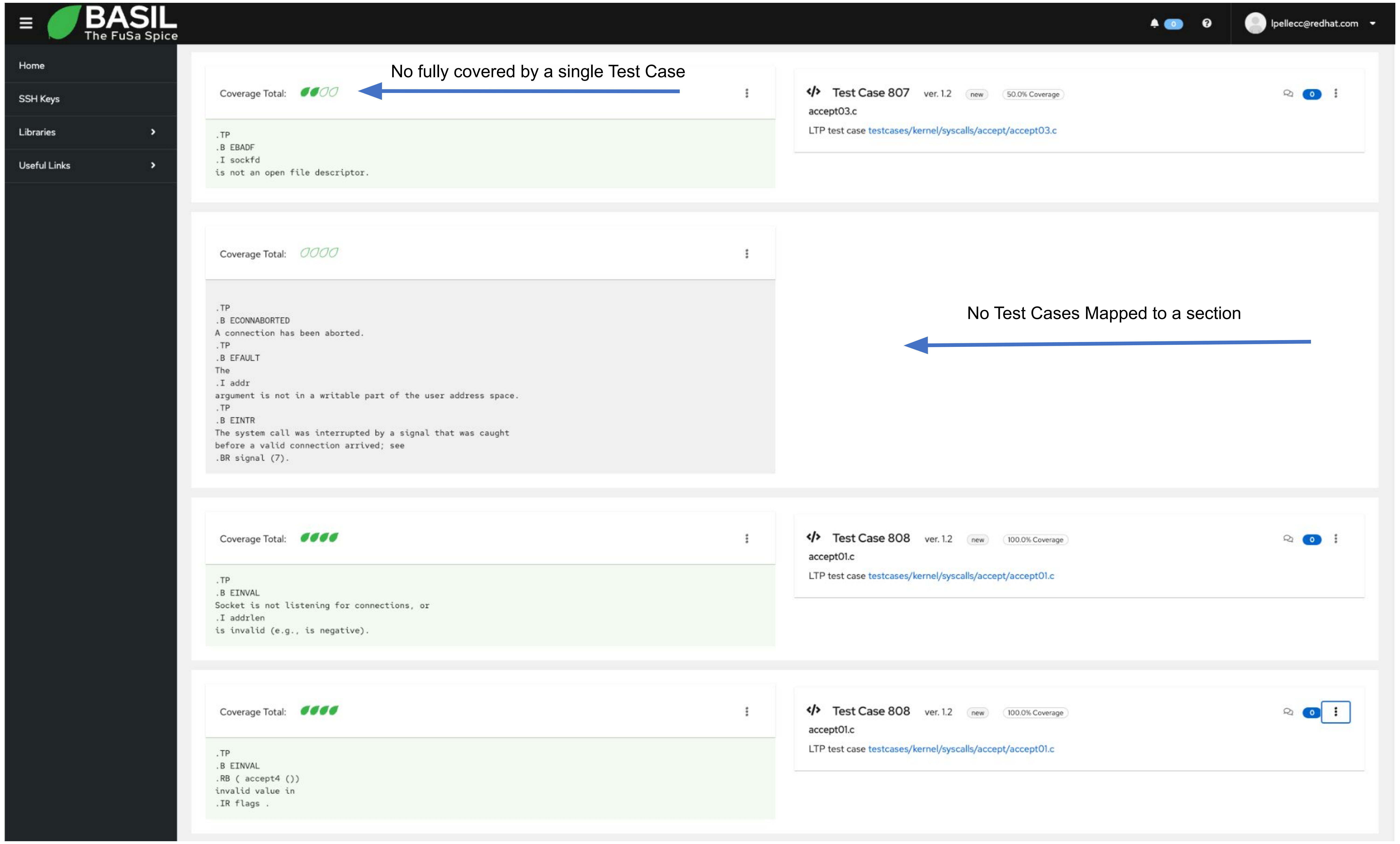
At the time of writing, the difference between the latest man-pages release and the `HEAD` of the repository (usually the latest commit) is well over 100 commits. This represents about 9 weeks of changes. If you are using a stable Linux distribution, your man-pages package may well be years old. So as with the kernel, it is best to have the Git repository as a reference.

You could also find a system call with untested parameters or use whatever it is you are planning to use the LTP for.

Value every contribution



Highlight gap in BASIL



No fully covered by a single Test Case

Coverage Total:

```
.TP
.B EBADF
.I sockfd
is not an open file descriptor.
```

Test Case 807 ver. 1.2 new 50.0% Coverage

accept03.c

LTP test case testcases/kernel/syscalls/accept/accept03.c

Coverage Total:

```
.TP
.B ECONNABORTED
A connection has been aborted.
.TP
.B EFAULT
The
.I addr
argument is not in a writable part of the user address space.
.TP
.B EINTR
The system call was interrupted by a signal that was caught
before a valid connection arrived; see
.BR signal (7).
```

No Test Cases Mapped to a section

Coverage Total:

```
.TP
.B EINVAL
Socket is not listening for connections, or
.I addrlen
is invalid (e.g., is negative).
```

Test Case 808 ver. 1.2 new 100.0% Coverage

accept01.c

LTP test case testcases/kernel/syscalls/accept/accept01.c

Coverage Total:

```
.TP
.B EINVAL
.RB ( accept4 ())
invalid value in
.IR flags .
```

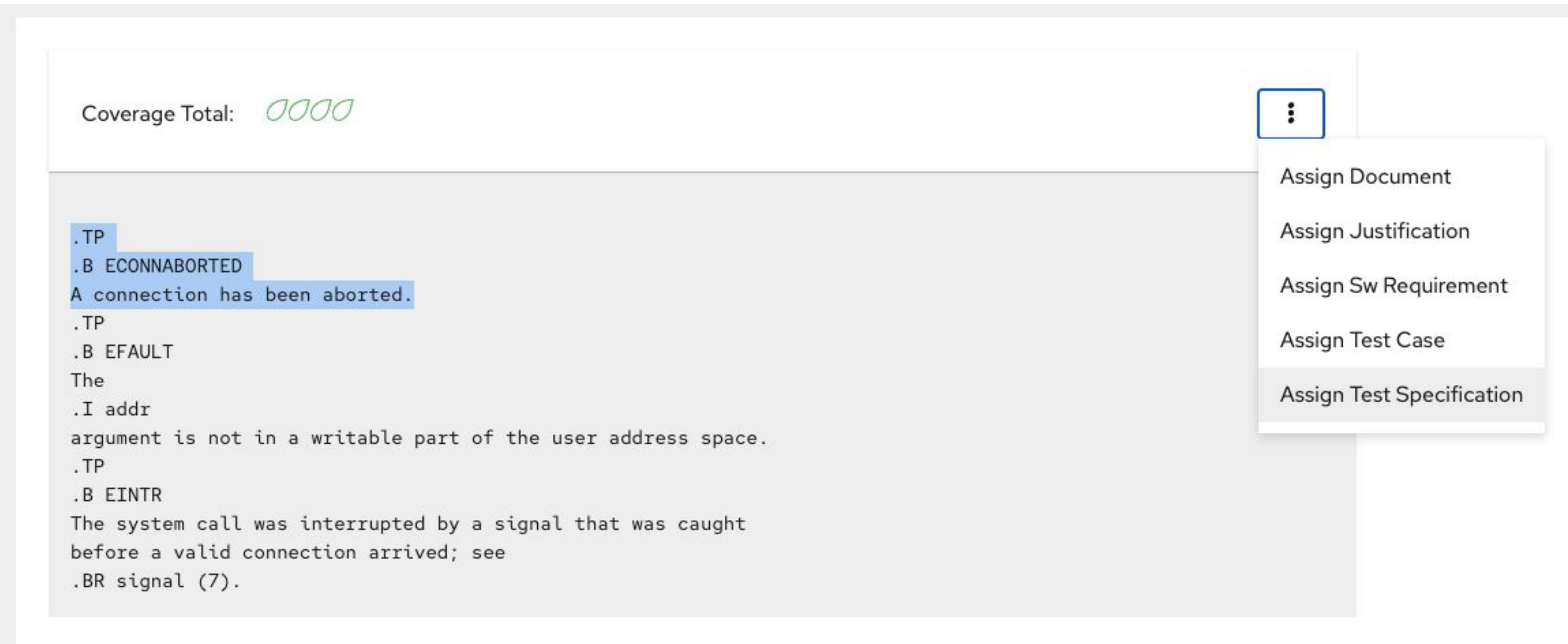
Test Case 808 ver. 1.2 new 100.0% Coverage

accept01.c

LTP test case testcases/kernel/syscalls/accept/accept01.c

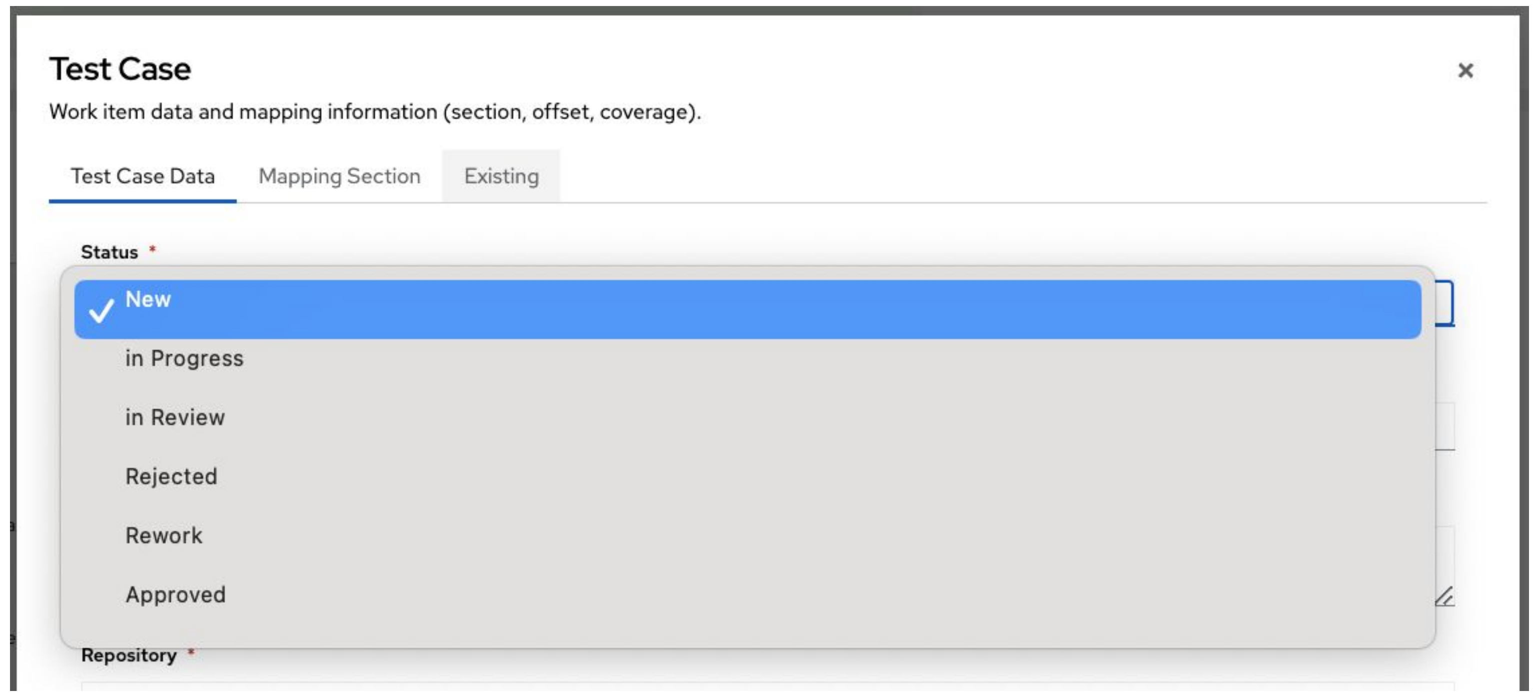
Collaborative tool for communities

Users can work in parallel on different work item types



```
.TP
.B ECONNABORTED
A connection has been aborted.
.TP
.B EFAULT
The
.I addr
argument is not in a writable part of the user address space.
.TP
.B EINTR
The system call was interrupted by a signal that was caught
before a valid connection arrived; see
.BR signal (7).
```

Work item lifecycle and approvals



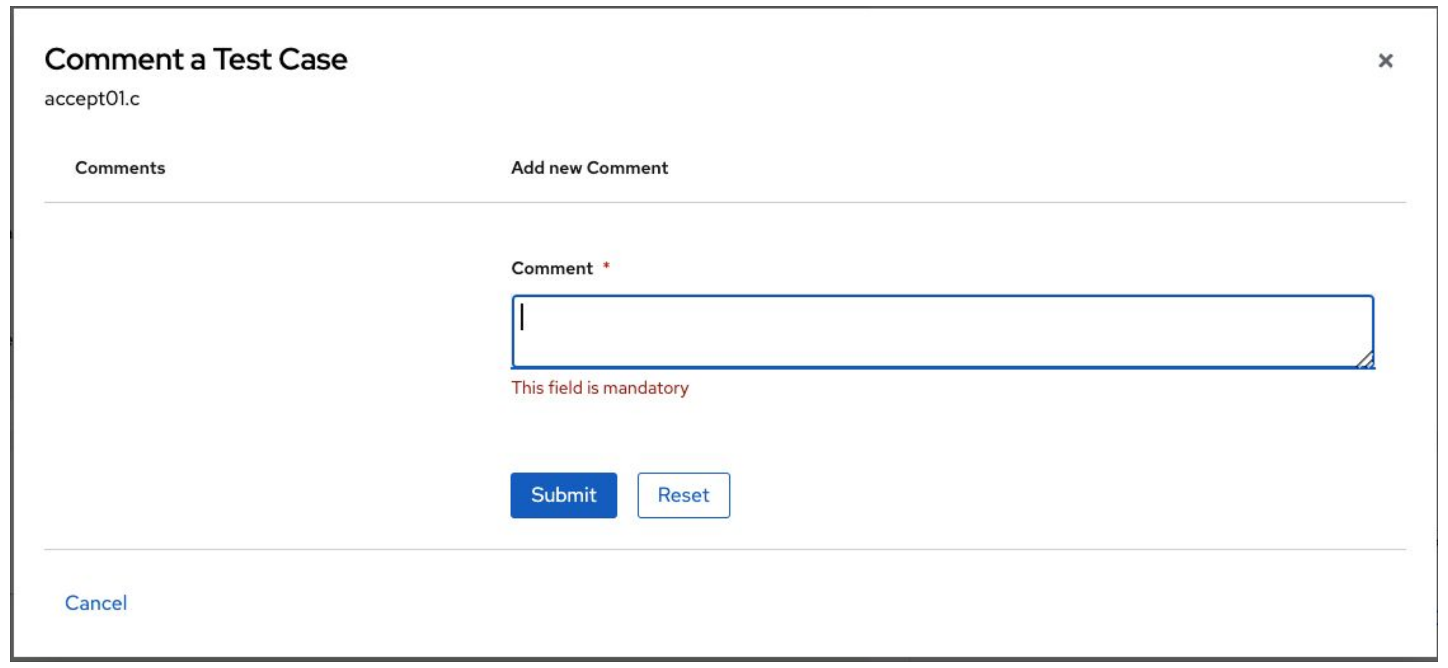
Test Case

Work item data and mapping information (section, offset, coverage).

Status *: ✓ New

Repository *

Possibility to share comments



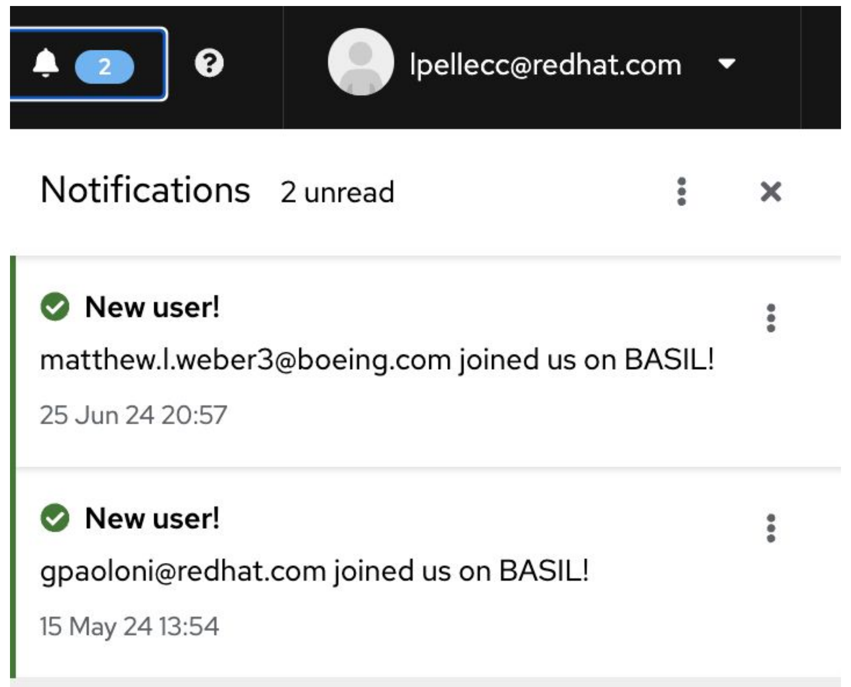
Comment a Test Case

accept01.c

Comment *

This field is mandatory

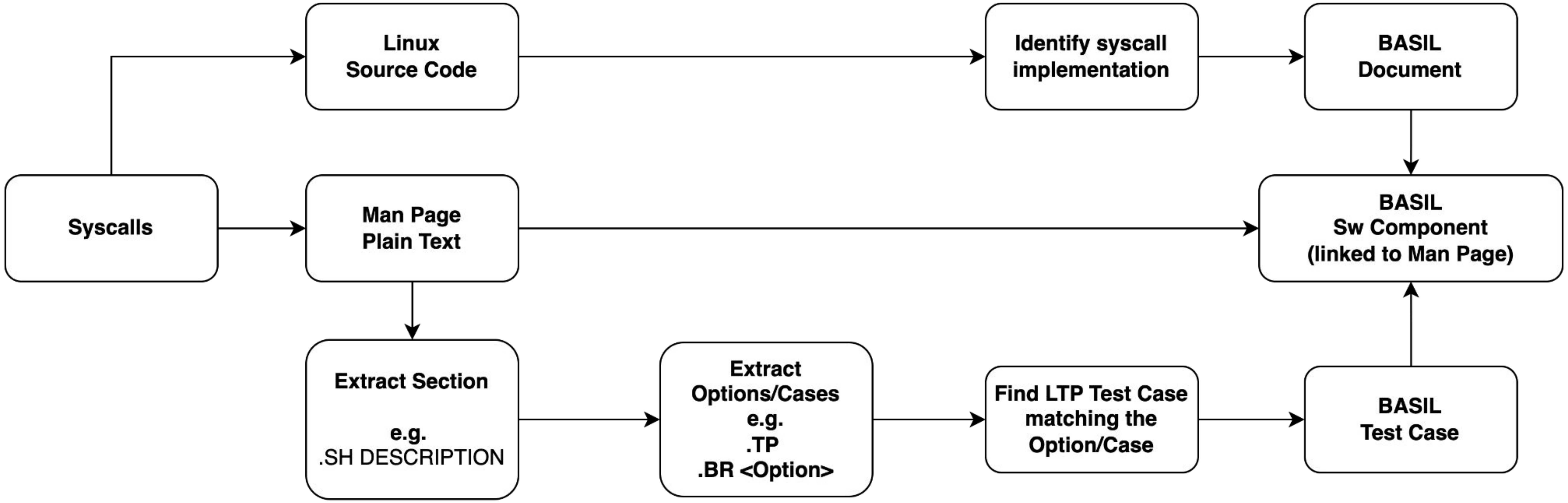
Notifications



New user!
matthew.l.weber3@boeing.com joined us on BASIL!
25 Jun 24 20:57

New user!
gpaoloni@redhat.com joined us on BASIL!
15 May 24 13:54

Automated Analysis



Syscalls

Linux Source Code

Identify syscall implementation

BASIL Document

Man Page Plain Text

BASIL Sw Component (linked to Man Page)

Extract Section

e.g.
.SH DESCRIPTION

Extract Options/Cases
e.g.
.TP
.BR <Option>

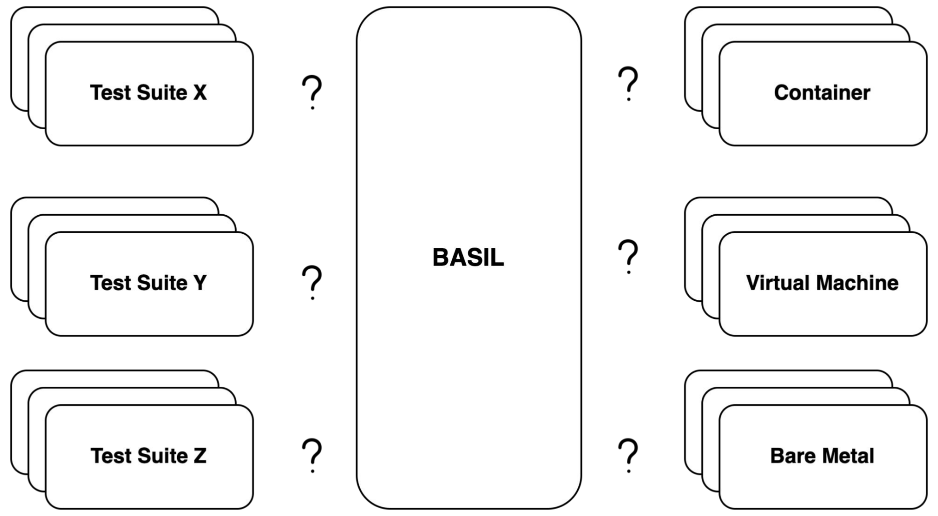
Find LTP Test Case matching the Option/Case

BASIL Test Case

ELISA BASIL Instance

http://elisa-builder-00.iol.unh.edu:9056

Test Execution

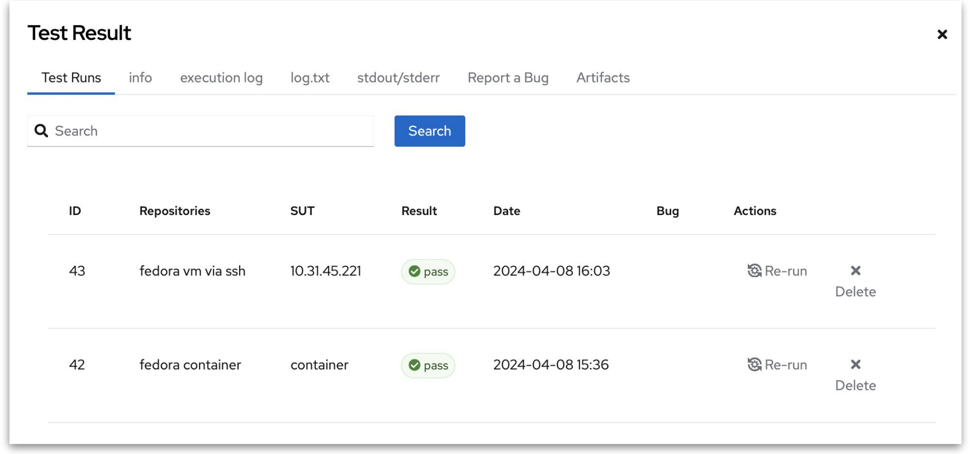


```
summary: Example fmf file
description: This is an example of an fmf metadata file
contact: Luigi Pellecchia <lpellecc@redhat.com>
component: basil
recommend: python3
test: python3 test_entrypoint.py
framework: shell
duration: 5m
```

tmt introduce an abstraction layer that allow BASIL to interact with any test suite.

tmt uses the flexible metadata format (fmf) to describe test cases, test plans, and user stories.

github.com/teemtee



ID	Repositories	SUT	Result	Date	Bug	Actions
43	fedora vm via ssh	10.31.45.221	✓ pass	2024-04-08 16:03		Re-run × Delete
42	fedora container	container	✓ pass	2024-04-08 15:36		Re-run × Delete

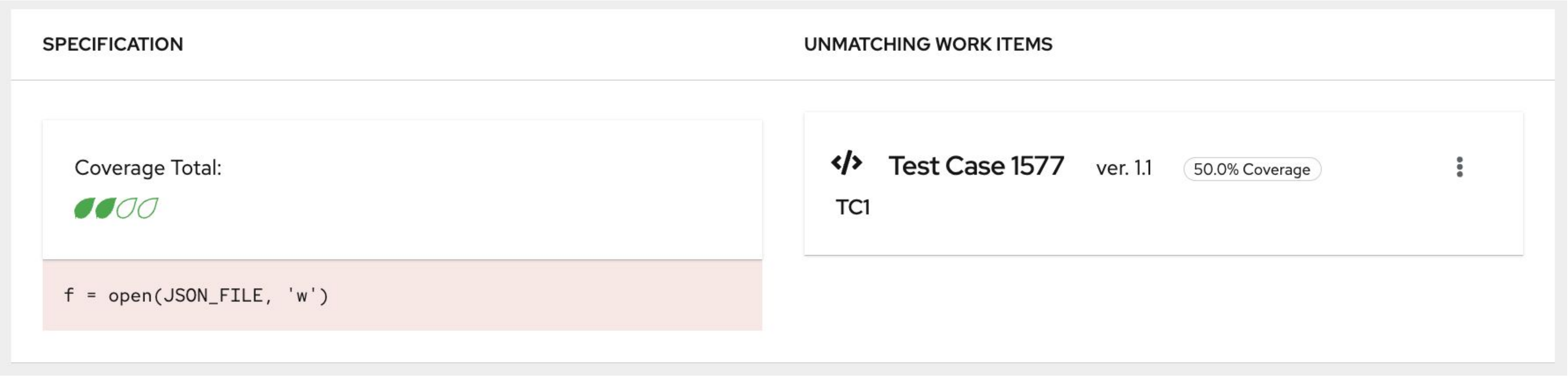
tmt supports multiple provisioning mechanisms as described at https://tmt.readthedocs.io/en/stable/spec/plans.html#provision

BASIL is leveraging:

- **container** (default Fedora)
- **connect** (SSH)

Follow Sw evolution

Highlight broken mappings



SPECIFICATION

UNMATCHING WORK ITEMS

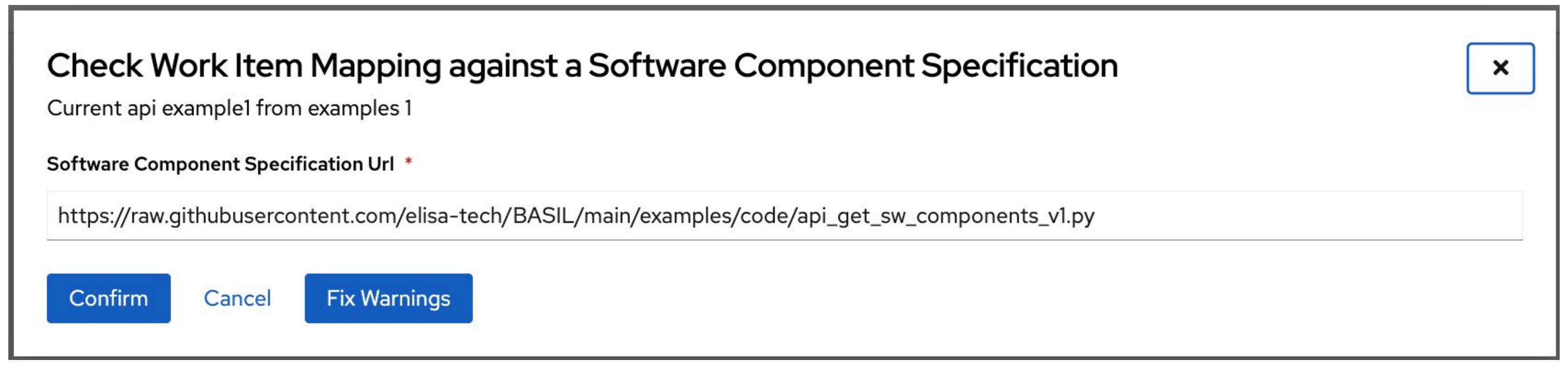
Coverage Total:

```
f = open(JSON_FILE, 'w')
```

</> Test Case 1577 ver. 1.1 50.0% Coverage

TC1

Evaluate in advance the effect of changes and Help to automatically fix warning



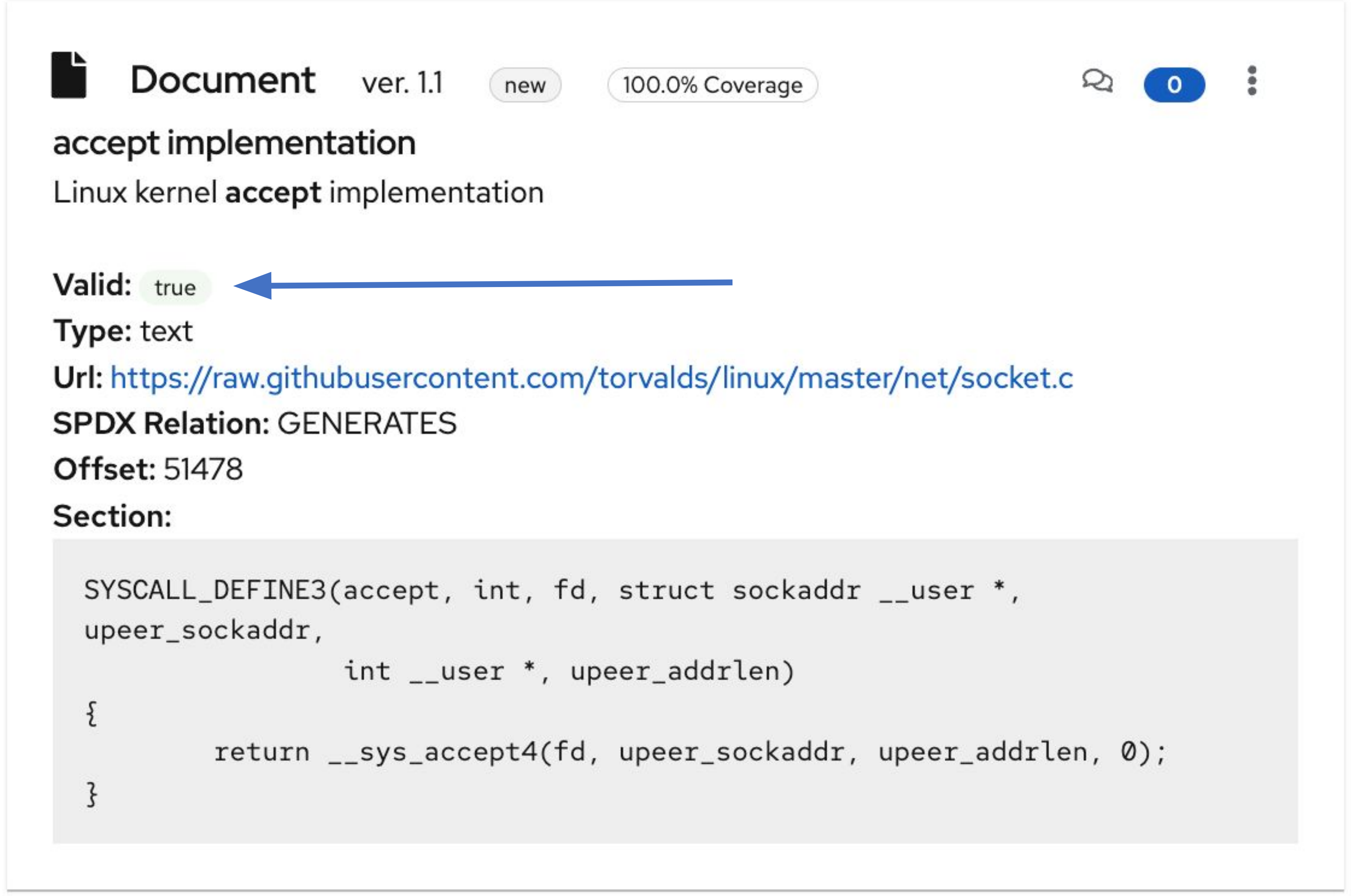
Check Work Item Mapping against a Software Component Specification

Current api example1 from examples 1

Software Component Specification Url *

https://raw.githubusercontent.com/elisa-tech/BASIL/main/examples/code/api_get_sw_components_v1.py

Automatic validation of linked textual document



Document ver. 1.1 new 100.0% Coverage 0

accept implementation

Linux kernel **accept** implementation

Valid: true

Type: text

Url: https://raw.githubusercontent.com/torvalds/linux/master/net/socket.c

SPDX Relation: GENERATES

Offset: 51478

Section:

```
SYSCALL_DEFINE3(accept, int, fd, struct sockaddr __user *,
upeer_sockaddr,
                int __user *, upeer_addrlen)
{
        return __sys_accept4(fd, upeer_sockaddr, upeer_addrlen, 0);
}
```

Possible approaches

Fixed version of the Reference Document

Pros

- No changes into the Reference Document
- Low update frequency of the mapping that is only affected by changes in the work items
- We have a picture of a specific version of the software component
- No need to implement CI in the specification repo
- BASIL can analyze in advance the impact of a change

Cons

- Doesn’t reflect the state of the art of the software component
- Need to maintain multiple versions of the software component

Follow the HEAD of the Reference Document git repository

Pros

- Keep the Reference Document up to date
- CI can help to identify changes and to auto-fix warnings

Cons

- Need to implement CI (that’s also funny)
- We don’t have a picture of a specific version of the software

Possible approaches

Fixed version of the Reference Document

Pros

- No changes into the Reference Document
- Low update frequency of the mapping that is only affected by changes in the work items
- We have a picture of a specific version of the software component
- No need to implement CI in the specification repo

Cons

- Doesn't reflect the state of the art of the software component
- Need to maintain multiple versions of the software component

or both

Follow the HEAD of the Reference Document git repository

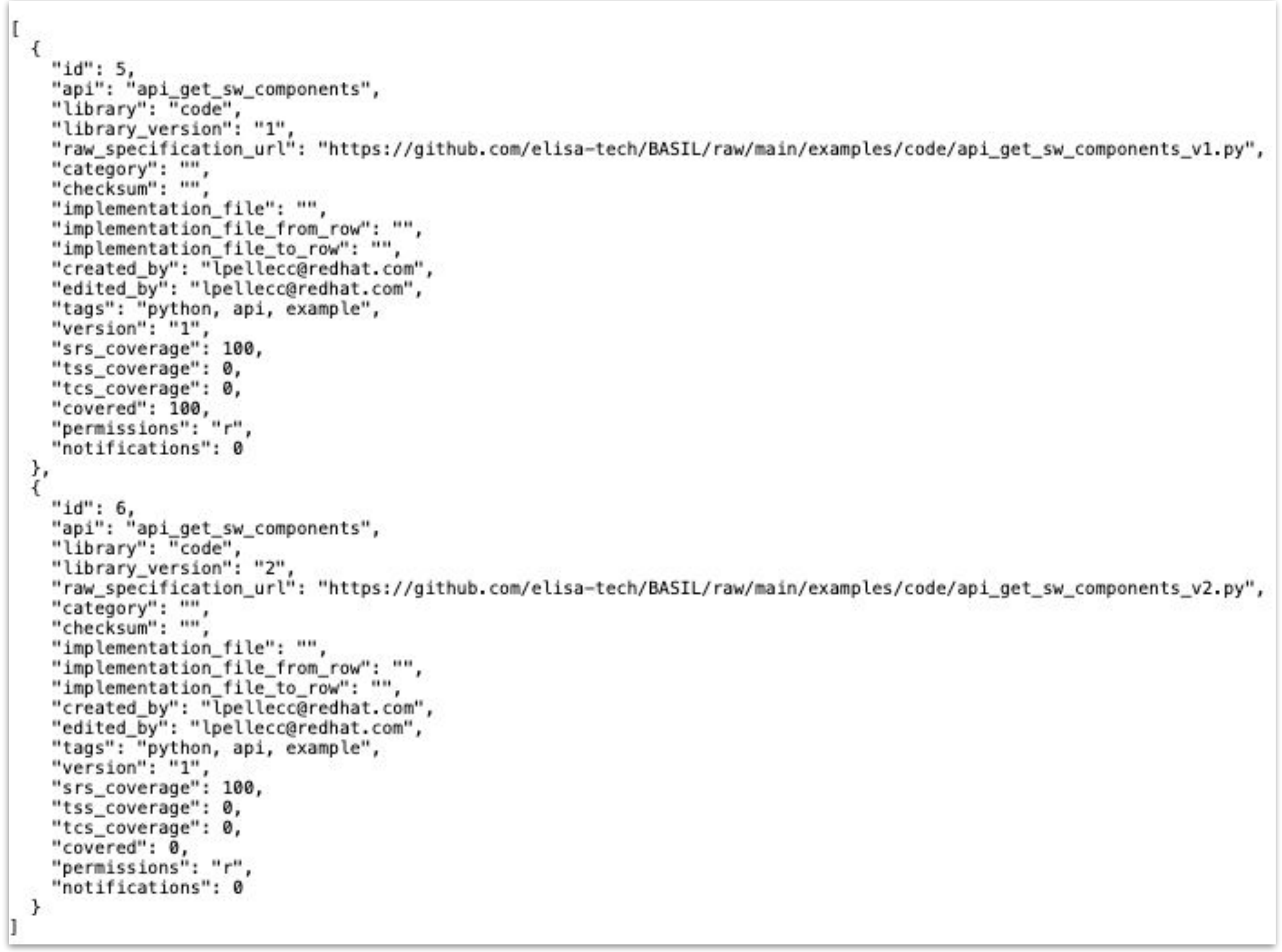
Pros

- Keep the Reference Document up to date
- CI can help to identify changes and to auto-fix warnings

Cons

- Need to implement CI (that's also funny)
- We don't have a picture of a specific version of the software

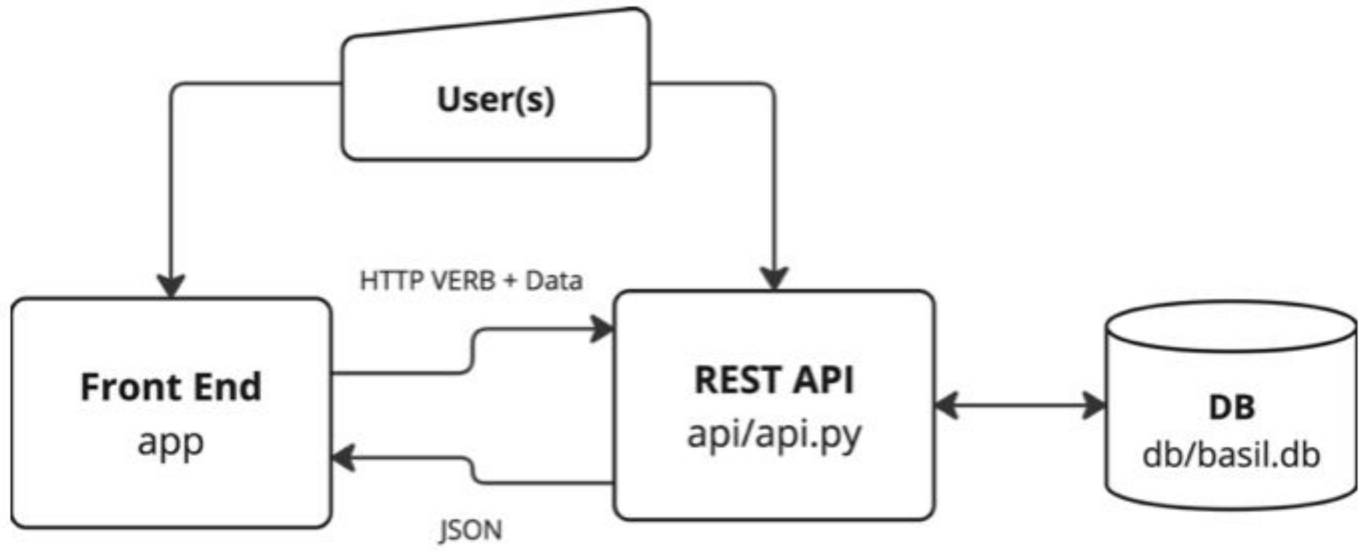
HTTP Api



```
[
  {
    "id": 5,
    "api": "api_get_sw_components",
    "library": "code",
    "library_version": "1",
    "raw_specification_url": "https://github.com/elisa-tech/BASIL/raw/main/examples/code/api_get_sw_components_v1.py",
    "category": "",
    "checksum": "",
    "implementation_file": "",
    "implementation_file_from_row": "",
    "implementation_file_to_row": "",
    "created_by": "lpellecc@redhat.com",
    "edited_by": "lpellecc@redhat.com",
    "tags": "python, api, example",
    "version": "1",
    "srs_coverage": 100,
    "tss_coverage": 0,
    "tcs_coverage": 0,
    "covered": 100,
    "permissions": "r",
    "notifications": 0
  },
  {
    "id": 6,
    "api": "api_get_sw_components",
    "library": "code",
    "library_version": "2",
    "raw_specification_url": "https://github.com/elisa-tech/BASIL/raw/main/examples/code/api_get_sw_components_v2.py",
    "category": "",
    "checksum": "",
    "implementation_file": "",
    "implementation_file_from_row": "",
    "implementation_file_to_row": "",
    "created_by": "lpellecc@redhat.com",
    "edited_by": "lpellecc@redhat.com",
    "tags": "python, api, example",
    "version": "1",
    "srs_coverage": 100,
    "tss_coverage": 0,
    "tcs_coverage": 0,
    "covered": 0,
    "permissions": "r",
    "notifications": 0
  }
]
```

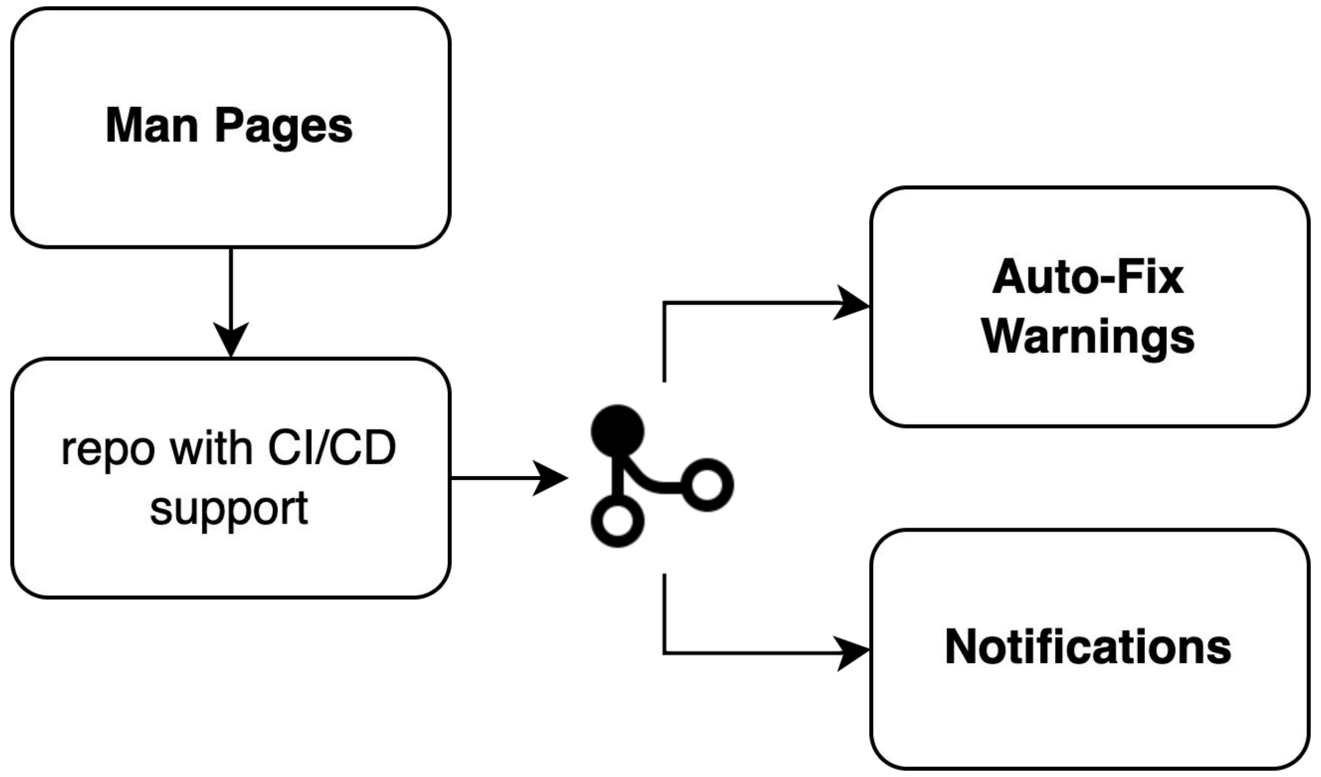
Any action in BASIL is performed via HTTP Api

It can be used to simplify the integration in CI/CD and other toolchain or to retrieve information to fill reports



Automation

If we track the HEAD



Add metadata in Test File to automatically update the mapping in CI

```
/*\
 * [Basil]
 *
 * - url:
https://git.kernel.org/pub/scm/docs/man-pages/man-pages.git/plain/man/man2/ch
mod.2?id=1e2d36deb2de1dbaf7084bb7dc3cb2c170cce852
 *   version: 1e2d36deb2de1dbaf7084bb7dc3cb2c170cce852
 *   coverage: 50
 *   coordinate: 2683,2780
*   sha1: 6fa01a0afe90dd46cbc1f5a43714a2a72fe5d442
 */
```

CI Job in the Source Code can:

- Fix BASIL Document Links in case of warnings
- Trigger Test for affected software components under specified conditions (and File Bugs)

Time for Questions

Source Code: https://github.com/elisa-tech/BASIL

Doc: https://basil-the-fusa-spice.readthedocs.io

ELISA Instance: http://elisa-builder-00.iol.unh.edu:9056

Linux Plumbers Conference

Vienna, Austria | September 18-20, 2024