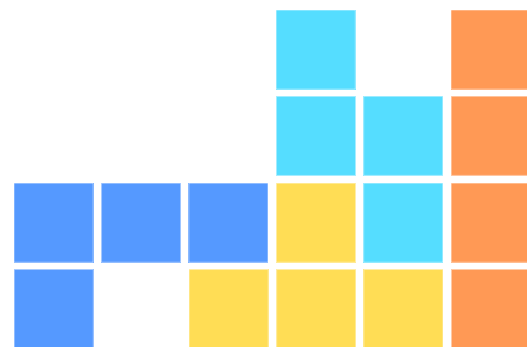
CWRAP

Testing complex software in CI

January 30th, 2016

Andreas Schneider and Jakub Hrozek

Red Hat Inc.



Who are we?

ABOUT US

We are Free and Open Source Software developer working on:

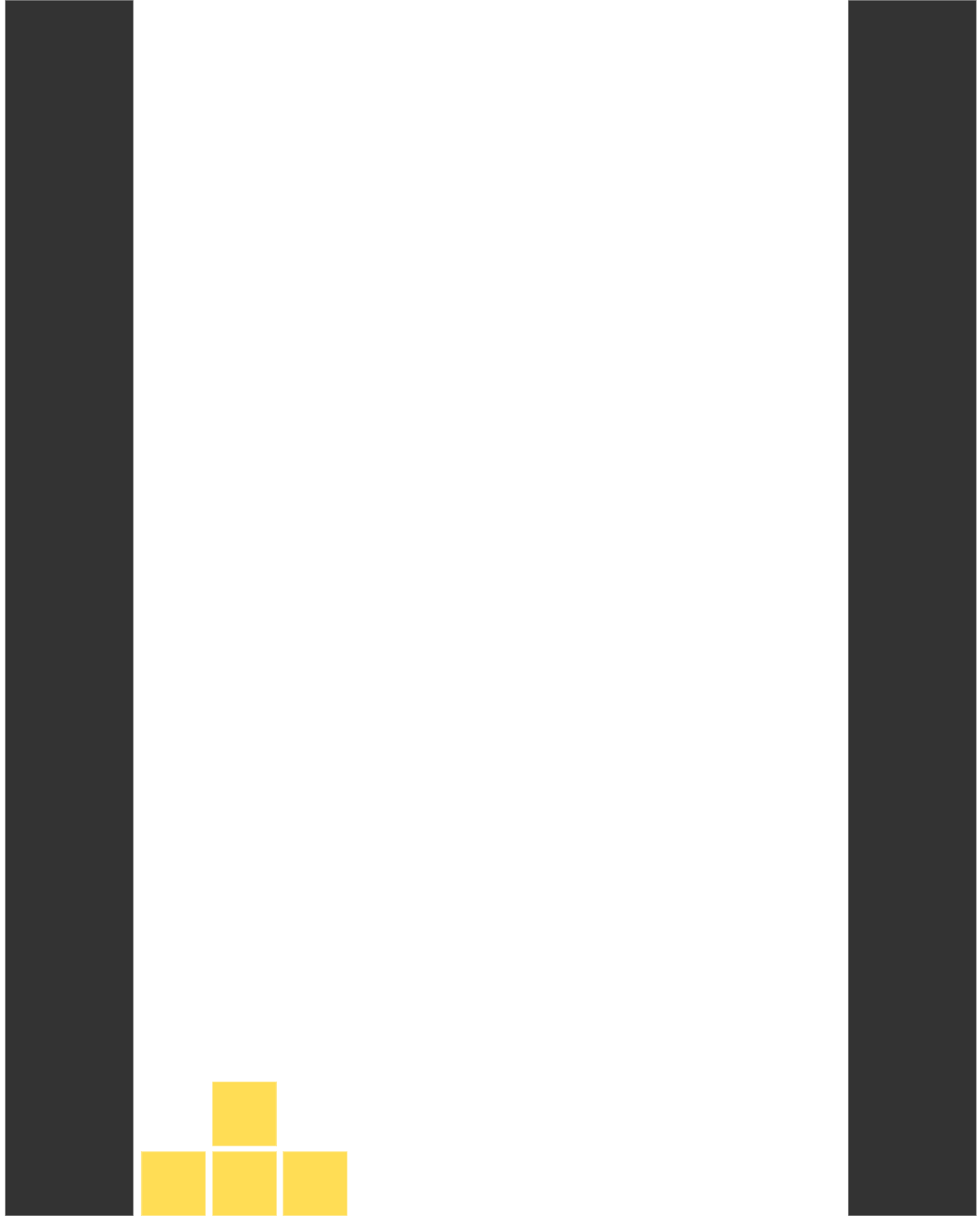
Andreas

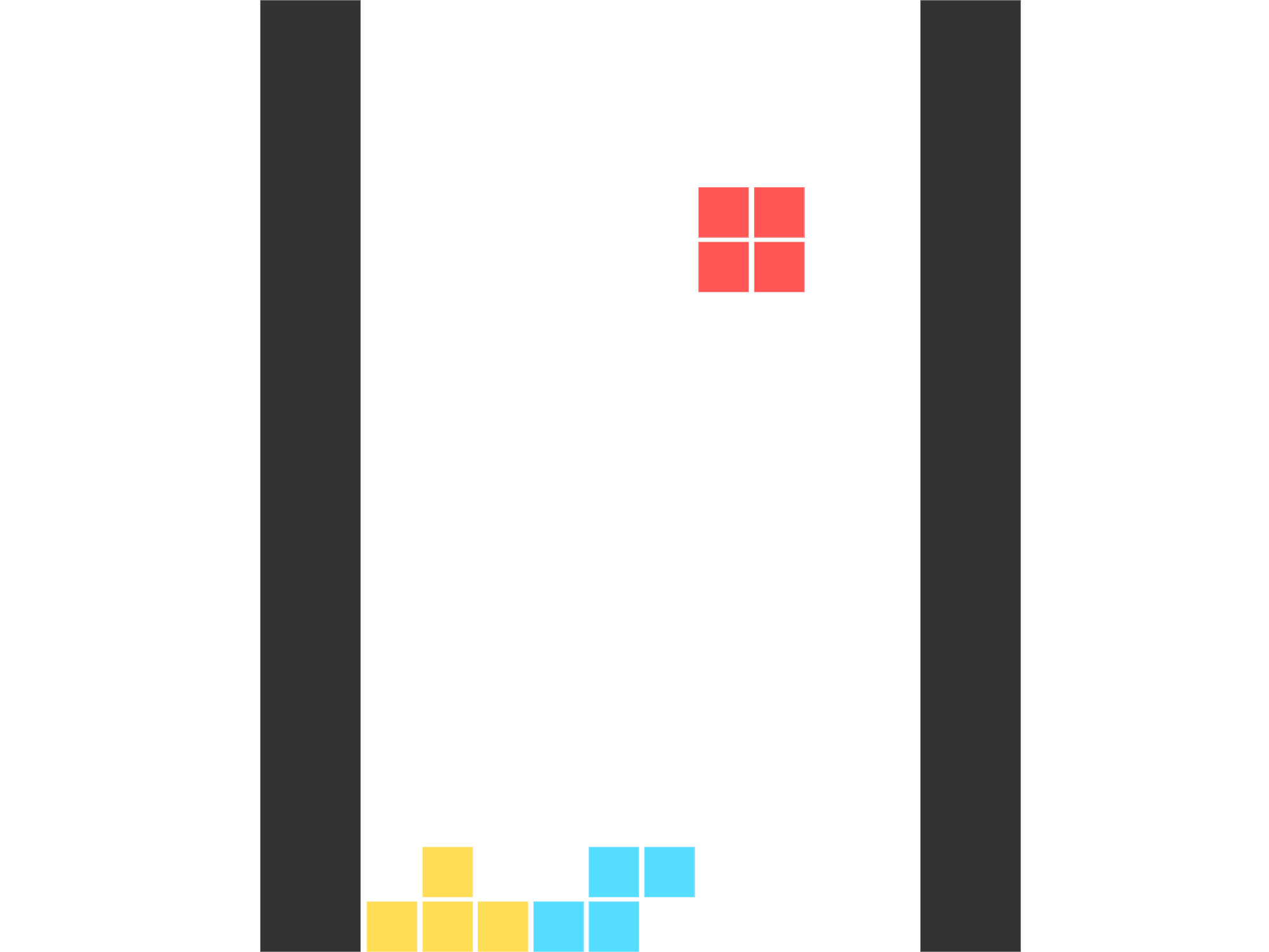
- → Samba - The domain controller and file server
- → libssh - The SSH Library
- → cmocka - a unit testing framework for C

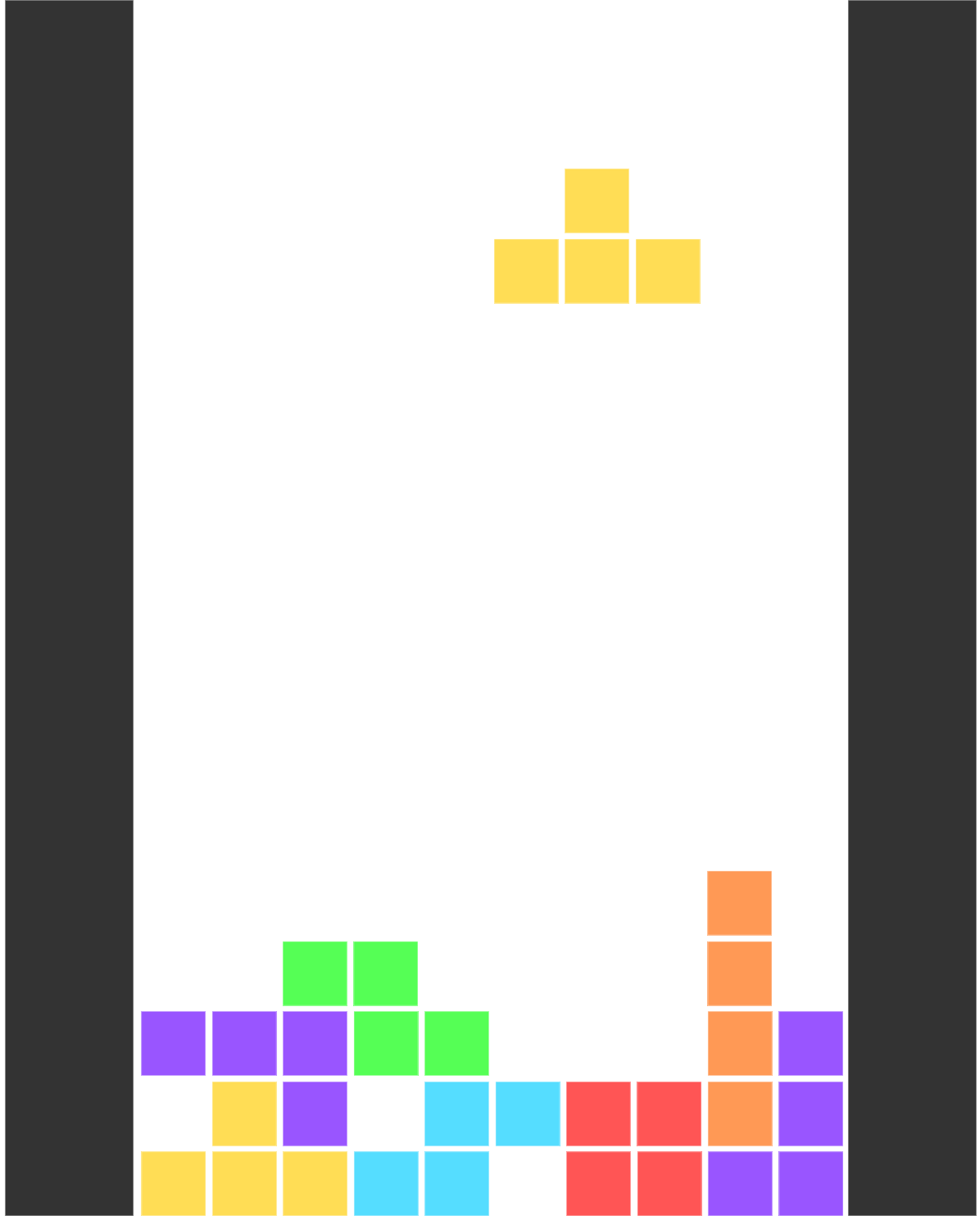
Jakub

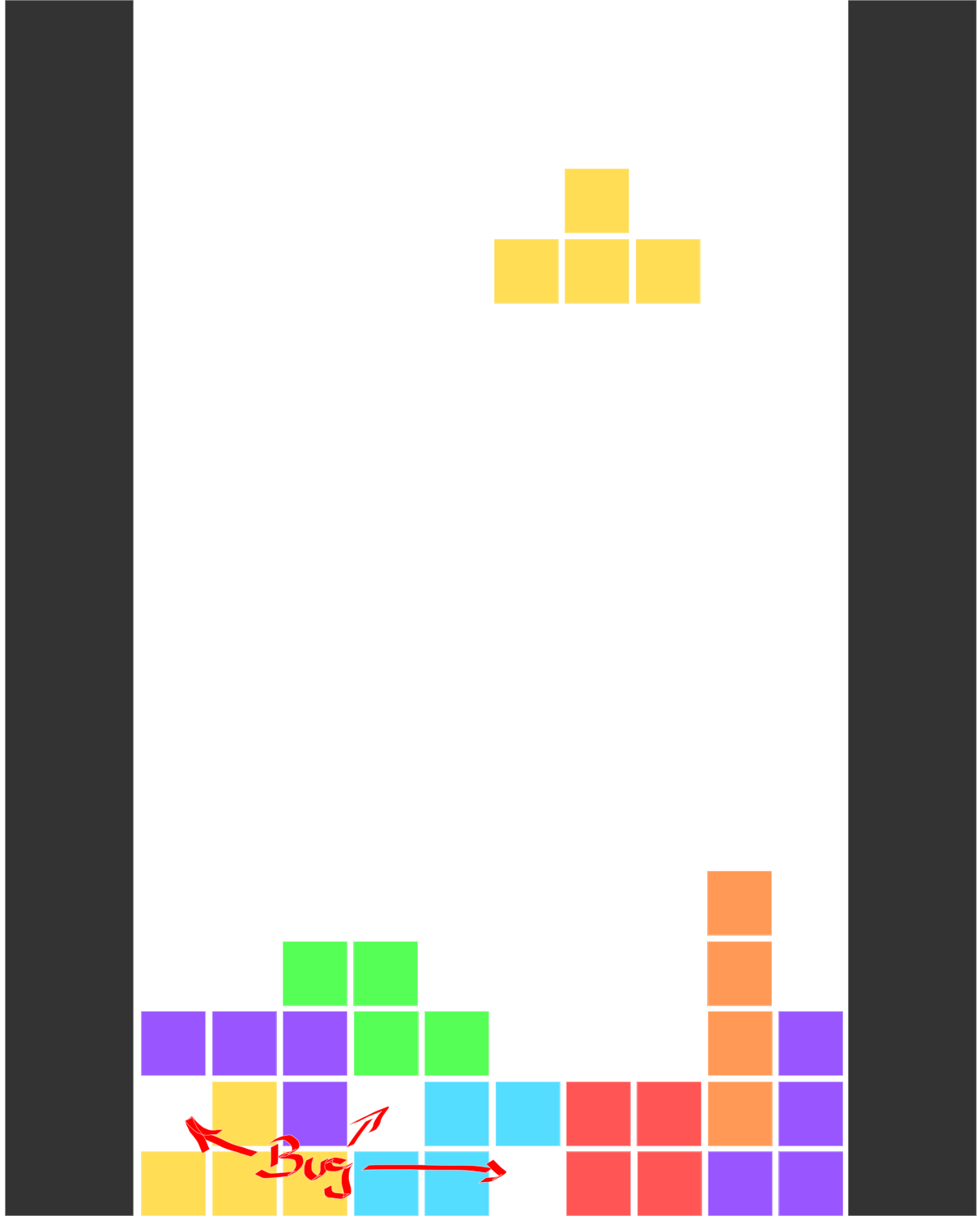
- → SSSD - A system deamon to integrate Linux machines with Directory services
- → FreeIPA - A free Directory Service implementation
- → cmocka - a unit testing framework for C

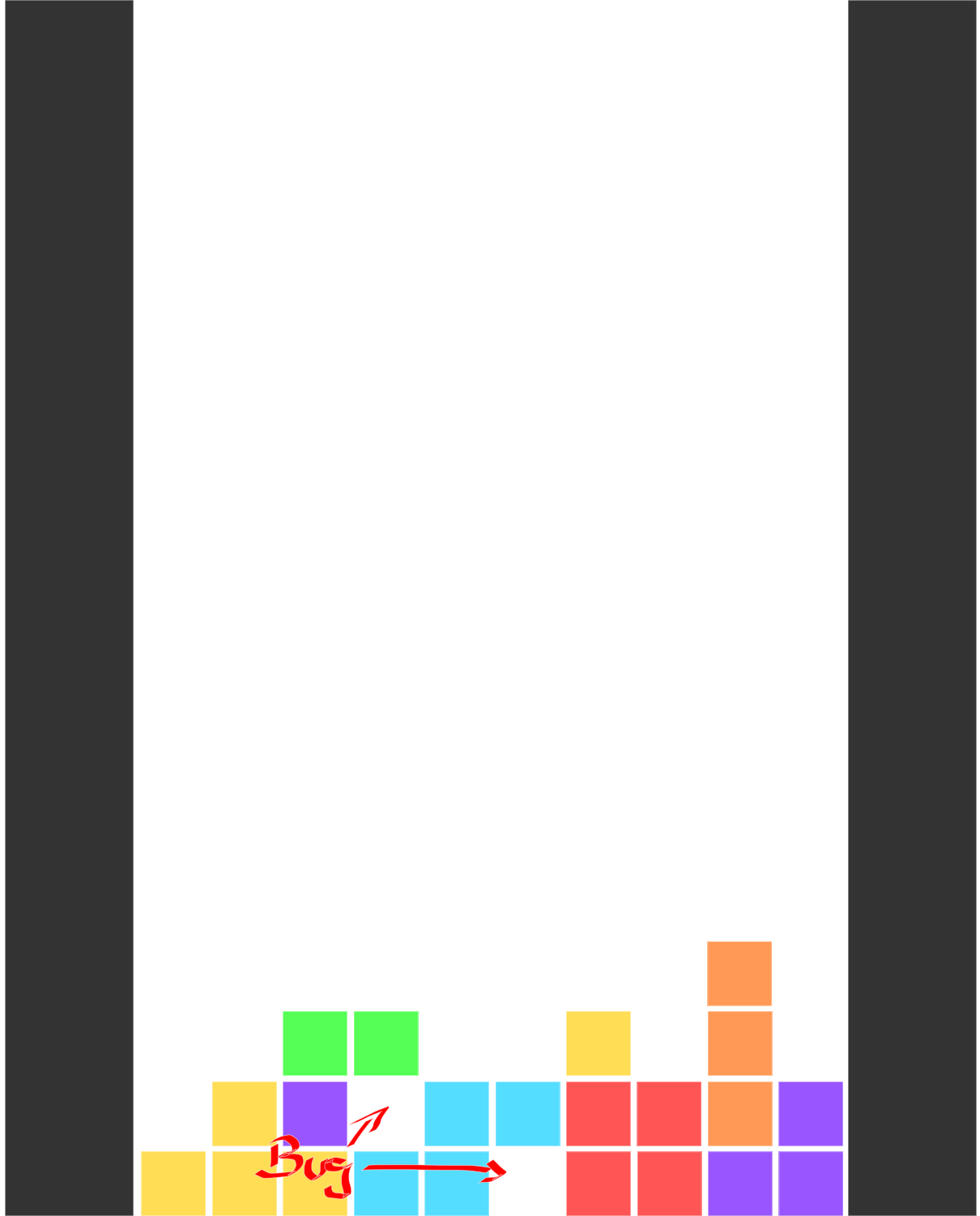
Do you remember this?

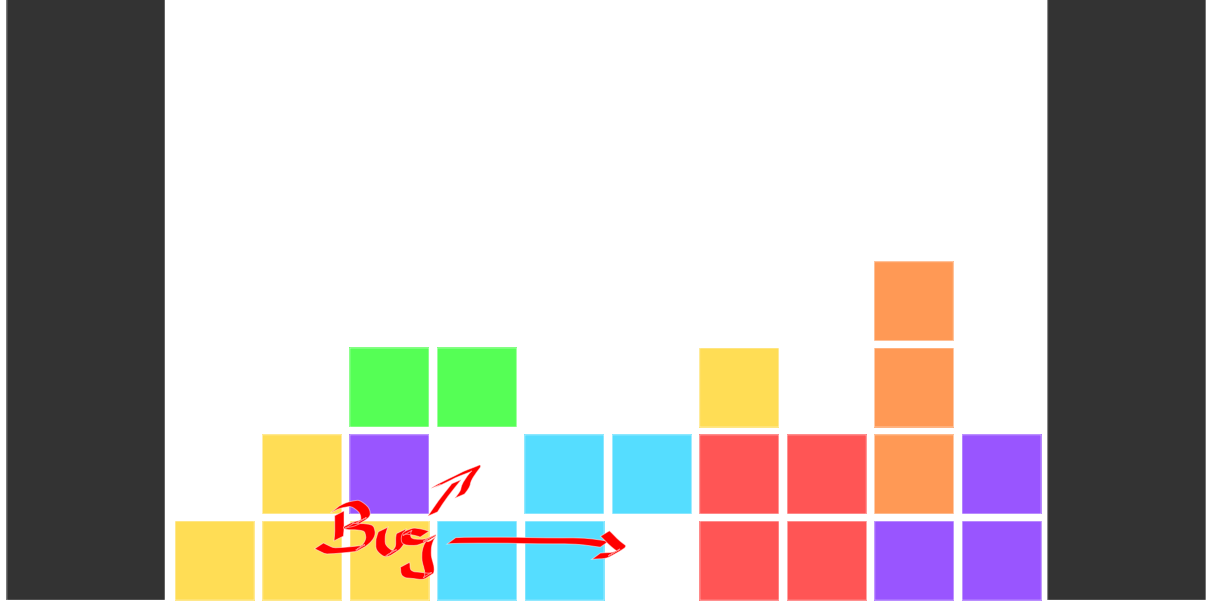


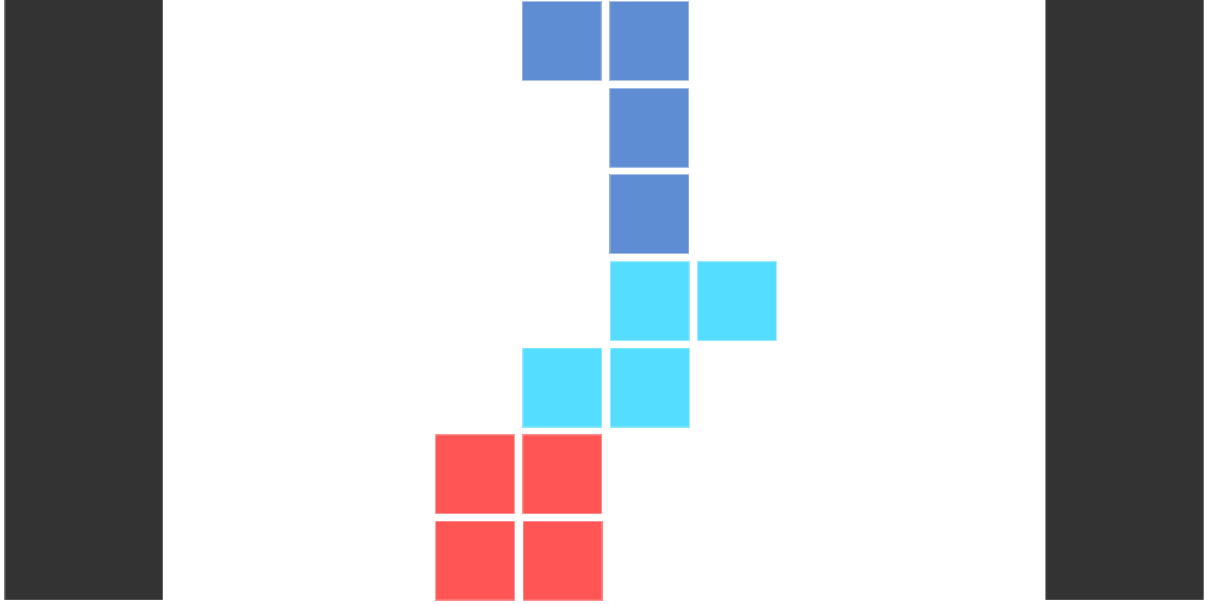





Bug


Bug


Bug



WHAT IS THIS TALK ABOUT?

HOW TO DO COMPLEX TESTS IN A LIMITED ENVIRONMENT

This talk is about improving your tests and enable you to test even the complex scenarios of your software.

WHAT DO WE MEAN BY COMPLEX SOFTWARE

Complex software

- → A simple software to test would be the command **`whoami`**
- → An example of complex software is a system service that runs as root ...
- → ...or a service that communicates over the network ...
- → ...or a software that logs in users ...

LETS LOOK AT THE PROBLEMS FIRST

PROBLEMS TESTING COMPLEX SOFTWARE AT FOR THE ENTERPRISE

Problems

- → Not every software offers a test suite
- → Only unit tests for some low level functions are offered
- → Test suites often require machine provision or provisioning of a complete network with multiple machines

HOW DO WE DO TESTING AT RED HAT

Traditional approach => Development and testing are seperate

→ Red Hat Quality Engineering (QE) creates tests with bare metal or VMs

RED HAT QE DEVELOPS TESTS TO AVOID REGRESSIONS IN FUTURE

→ This is complex to set up

→ They do testing based on RPM packages produced by development

OUR GOAL FOR DEVELOPER-FRIENDLY TESTING

→ Complete integration tests

→ Possible to run without a complex environment

→ Enable full Continuous Integration (CI) tests during code review

THE SAMBA TESTSUITE

The Samba project had a complex testsuite in place but hard to use outside Samba

Awesome testsuite, but

- → Special code for testing, not usable by other project
- → Did not require any special set up machines or network
- → Goal: Make it useable for every project out there => the birth of cwrap
- → This enables other software to develop great tests easily, too

WHAT IS CWRAP?

WHAT IS CWRAP?

cwrap is..

a set of tools to create a fully isolated network environment to test client/server components on a single host, complete with synthetic account information, hostname resolution and privilege separation support. The heart of cwrap consists of four libraries you can preload to any executable.

WHAT IS CWRAP?

Think of cwrap like it is

"The Matrix"

where reality is simulated and everything is a lie.

WHAT IS PRELOADING?

- → Preloading is a feature of the dynamic linker.
- → It is available on most Unices.
- → It loads a user specified, shared library before all others.
- → The library to preload is defined by the environment variable `LD_PRELOAD=libwurst.so`.
- → The symbols of the preloaded library are bound first.
- → See ›man ld.so‹

THE PRELOADABLE LIBRARIES OF CWRAP

cwrap provides several libraries you can preload:

- → socket_wrapper
  Allows network simulation
- → nss_wrapper
  Fakes user/group information
- → uid_wrapper
  Fakes privilege separation
- → resolv_wrapper
  Allows you to talk to your DNS server or fake DNS responses
- → pam_wrapper
  Enables testing of PAM modules and appliations

THE WRAPPERS IN DETAIL

SOCKET_WRAPPER

When preloaded and enabled:

- → Redirects all network communication to happen over Unix sockets.
- → Support for IPv4 and IPv6 socket and addressing emulation.
- → Ablility to capture network traffic in pcap format.

The idea and the first incarnation of socket_wrapper has been written by Jelmer Vernooij in 2005.
It made it possible to run the Samba torture suite against smbd in 'make test'.

SOCKET_WRAPPER DEMO

NSS_WRAPPER

When preloaded and enabled:

- → Provides account information for users and groups.
- → Can do network name resolution using a hosts file.
- → Allows loading and testing of NSS modules.

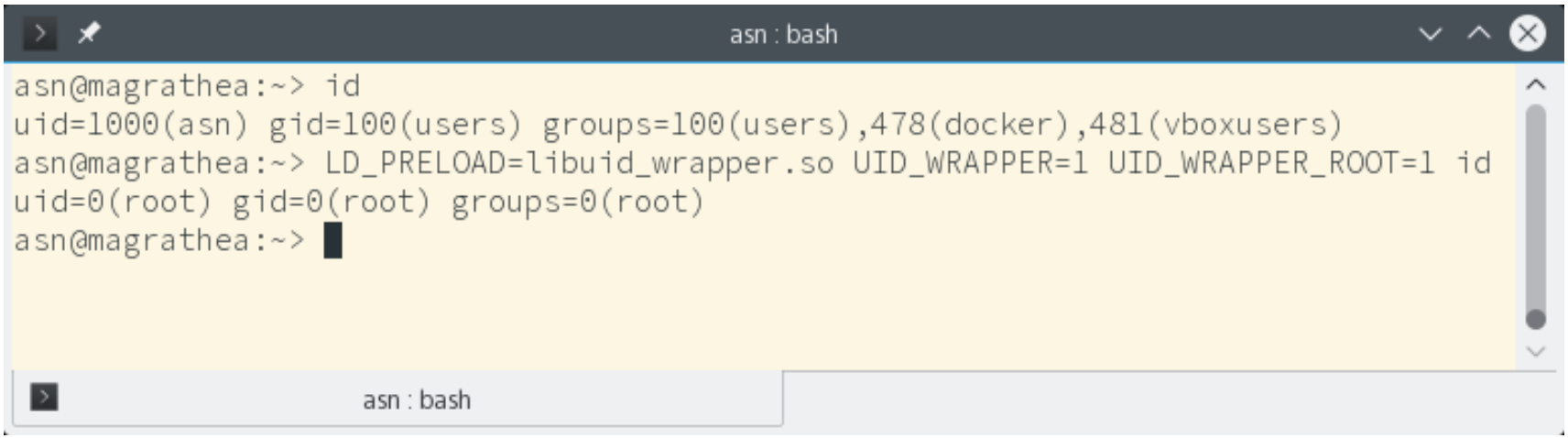
UID_WRAPPER

When preloaded and enabled:

- → Allows uid switching as a normal user.
- → You can start any application making it believe it is running as root.
- → Support for user/group changing in the local thread using the syscalls (like glibc).

uid_wrapper has been completely rewritten to support the last two features.

UID_WRAPPER


asn : bash
asn@magrathea:~> id
uid=1000(asn) gid=100(users) groups=100(users),478(docker),481(vboxusers)
asn@magrathea:~> LD_PRELOAD=libuid_wrapper.so UID_WRAPPER=1 UID_WRAPPER_ROOT=1 id
uid=0(root) gid=0(root) groups=0(root)
asn@magrathea:~>
asn : bash

RESOLV_WRAPPER

When preloaded and enabled:

→ It either redirects name queries to the nameservers specified in your own `resolv.conf`.

→ or can fake DNS queries using a simple formatted DNS hosts file.

PAM_WRAPPER

The latest addition to the cwrap project, released a couple of weeks ago.

WHAT IS PAM?

→ PAM is the Linux system-level authentication subsystem

→ abstracts authentication into modules where each service has its own configuration

TESTING PAM

→ There are two sides to testing PAM

→ Testing a PAM-aware application (like a SSH server)

→ Or testing a PAM module (like pam_unix or pam_sss)

PAM_WRAPPER

- → Enables loading of service files from a custom location
- → Provides a simple module that authenticates against a plaintext database
- → Offers a C and Python API to write tests easily

PUTTING IT ALL TOGETHER

- → Using these wrappers, we can set up a real-world test without a heavyweight environment
- → Example: Samba tests
- → Example: SSSD integration tests

SAMBA and SSSD demos

WHO USES CWRAP?

- → Samba
- → SSSD (uses all wrappers in the meantime)
- → libssh (Test libssh against OpenSSH)
- → Docker and OpenStack
- → Knot DNS (.cz root domain server)

LINKS

- → Website: **https://cwrap.org/**
- → Mailinglist: **https://lists.samba.org/mailman/listinfo/samba-technical**
- → IRC: #cwrap @ freenode

Questions & Answers

If you like this project tweet and blog about it!

https://cwrap.org/
https://cmocka.org/