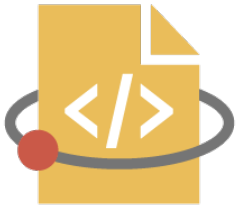
29th June 2022
Compiler-Research weekly meeting

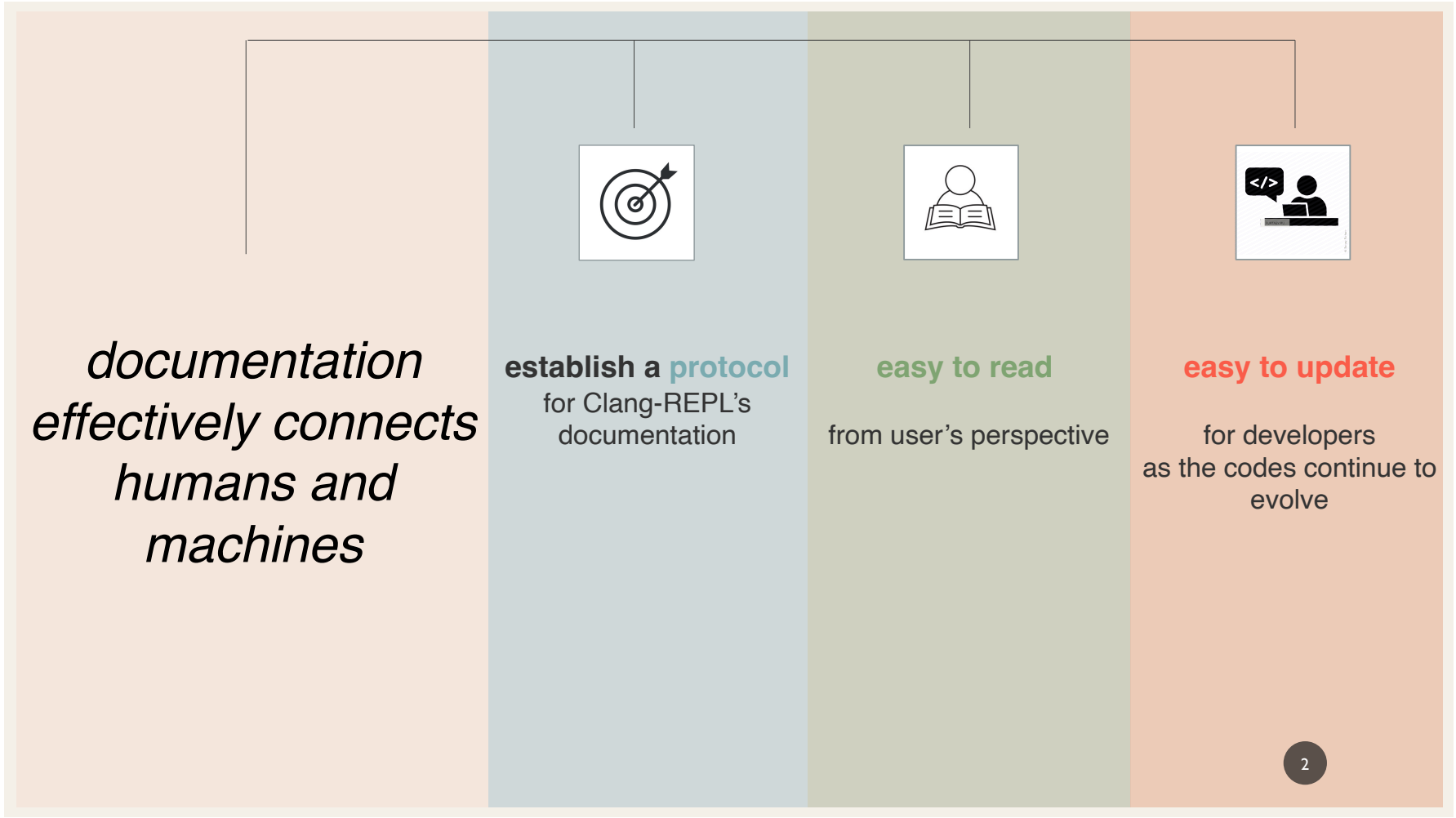
Improving the Clang-REPL documentation
(June—October 2022)

Author: Sara Bellei
Ph.D in Physics, Polytechnic University of Milan, Italy (2017)

Mentors: David Lange, Vassil Vasilev



Season of Docs



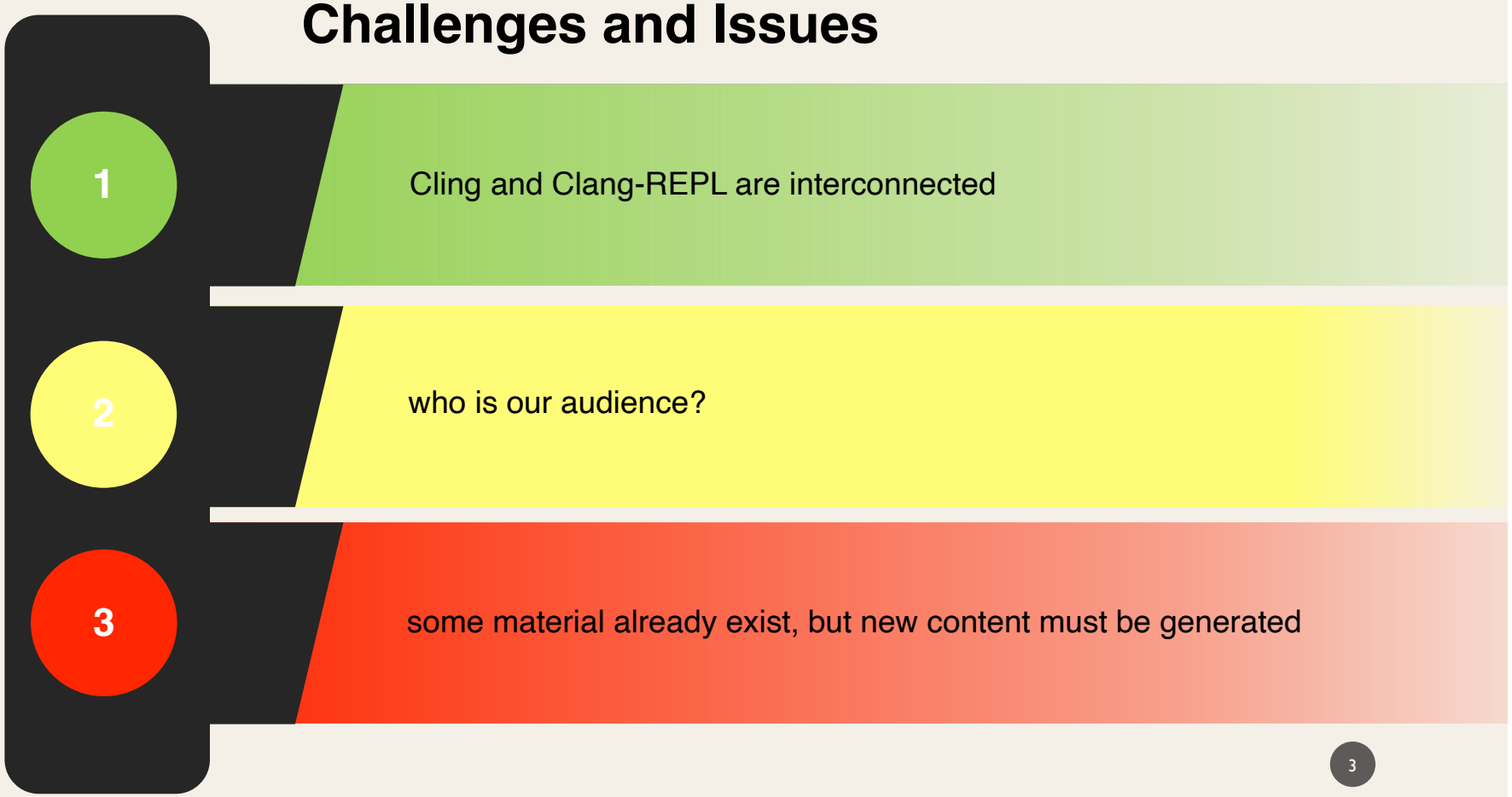
Documentation effectively connects humans and machines
(https://guides.lib.berkeley.edu/how-to-write-good-documentation)

Technical Writing is a form of communication that attempts to take a technical field (like developing a new software) and convey ideas from that field as efficiently as possible, to a diversified audience, sometimes to non experts, sometimes to experts.

We have two goals:

1. to create new content which will ensure that people who want to use Clang-REPL have the information they need to understand it, decide if it matches their needs, and operate it
2. to promote **in-team communication**, by creating content that will allow to advance better and faster, and **out-group communication**, encouraging other developers to contribute to this code

Challenges and Issues



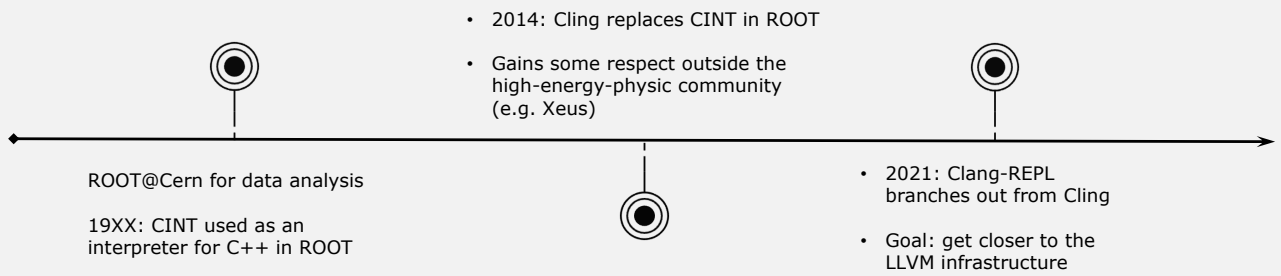
Challenges and Issues:

Three main challenges will have to be faced

Challenges and Issues

1

Cling and Clang-REPL are interconnected



in order to create Clang-REPL documentation, some of the Cling's content also have to be rewritten (or written from scratch)

Cling and Clang-REPL are interconnected:

At this point, is impossible to talk about Clang-REPL without talking first about Cling.
This means that some parts of the documentation, the ones that introduce Clang-REPL should be, to some extent, written in a consistent way with the Cling documentation, because most likely a reader who wants to know what Clang REPL is will have to read something about Cling.

Challenges and Issues

2

who is our audience?

the community of LLVM developers

- encourage new developers to step in
- give insights about every new developments
- use a language register more technically detailed

everybody else:
LLVM users, data scientist, Jupyter users, …

- give a general overview of Clang-REPL
- keep the language register easier (they don't know Clang-REPL)
- provide tutorials tailored to different possible users (Rohit)

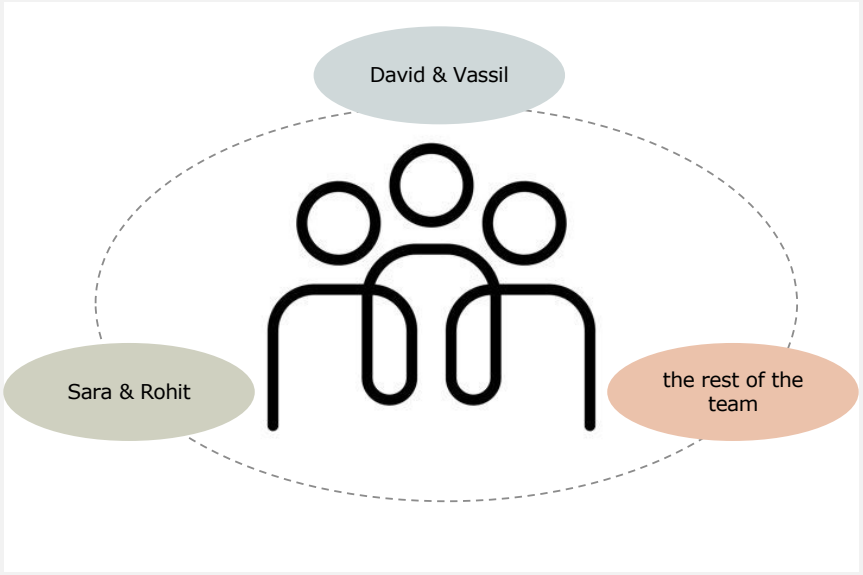
Who is our audience?

for both audiences we need to create guides and other forms of documentation which allows us to integrate new features into the core project as you develop them, and to describe these features in a way that makes sense to users.

Challenges and Issues

3

some material already exist, but new content must be generated



Some material already exist, but new content must be generated:

We need to make sure that the new content that the team of developer generate can be shared in the workplace so that we can subsequently write and communicate this information to the external public

Different Media Platforms

1. clang.llvm.org

2. Compiler-Researcher website

3. The LLVM Project Blog

Different media platforms:

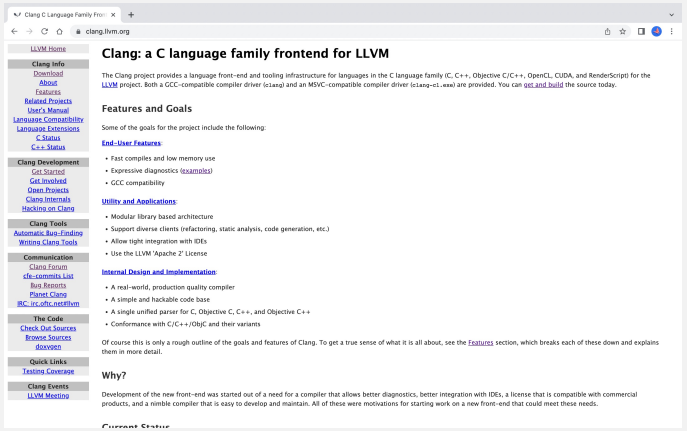
Here you can look at the media platform which will host Clang-REPL. The idea is that each of these platforms will host different content in an organized and consistent way

Different Media Platforms

1. clang.llvm.org

2. Compiler-Researcher website

3. The LLVM Project Blog



- Clang-REPL is part of Clang → LLVM release
  instead Cling: https://rawgit.com/root-project/cling/master/www/index.html
- follows the strict LLVM developer policy

The Clang webpage on LLVM.org:
https://clang.llvm.org/

Clang-REPL source code is found on Clang, because Clang-REPL is part of Clang, which comes as part of the larger LLVM release and not as a separate software.

LLVM has a strong policy-driven development code, which ensures that developers deliver software according to the platform's expectations.

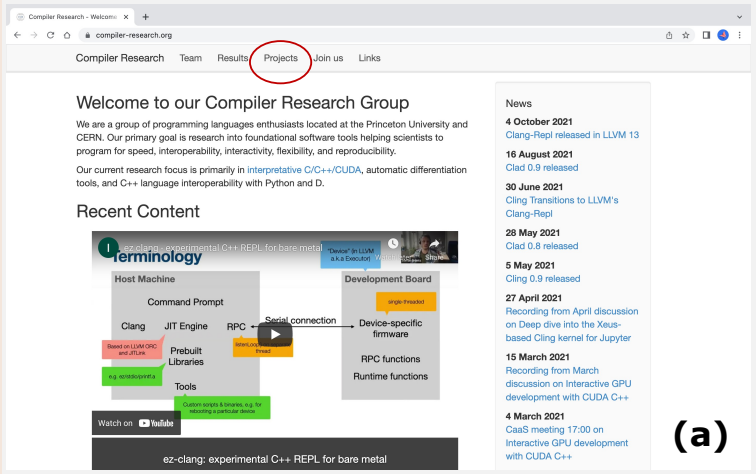
For all these reasons, the LLVM webpage will only host the source code and it will be kept clean and readable.

Different Media Platforms

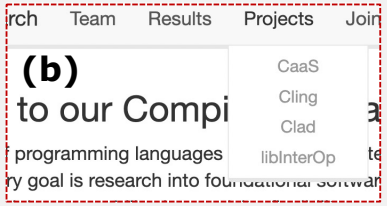
1. clang.llvm.org

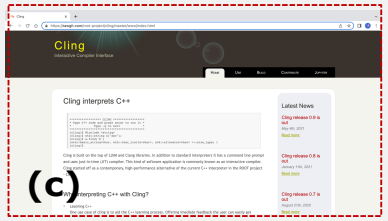
2. Compiler-Researcher website

3. The LLVM Project Blog



(a)





(c)

two separate links:

Projects:
- provide general information about them, but no details
- include link to the software webpage

Technical features
technical details
new developments
possible applications
tutorials

Compiler-Research website
https://compiler-research.org/

Cling and Clang-REPL project overview will be written under **Projects**:

1. provide general information about them, but no details
2. include link to the software webpage

A link called **Technical Features** will be added next to Projects:

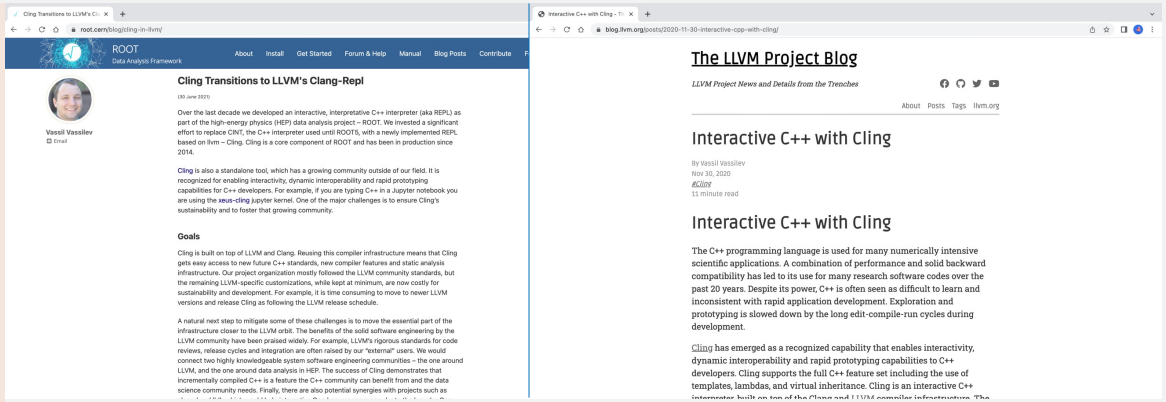
1. provide technical details
2. describe new developments
3. introduce to possible applications
4. include tutorials

Different Media Platforms

1. clang.llvm.org

2. Compiler-Researcher website

3. The LLVM Project Blog



Website → product-focused blog posts → **audience-focused**

goal: **to build a community around Clang-REPL**

how: by writing an interesting overview of the new developments then encourage the readers to follow-up by reading the official documentation

The LLVM project blog
https://blog.llvm.org/

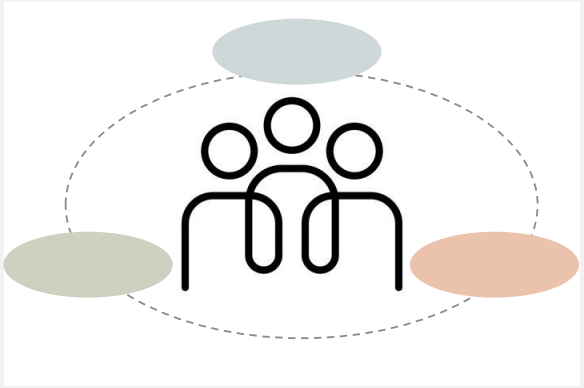
We will also keep posting on blogs, in particular on the LLVM blog.

A technical blog is the best form of advertising for the software, because unlike the documents describing technical features in detail, it will include plenty of keywords, visually appealing pictures, infographics and have a writing style that is more casual, as if the writer is talking to you. Their priority here is to reach a wider audience.

It will be used for describing Clang-REPL applications, and to keep the LLVM community updated regarding new Clang-REPL's features, and we will describe how these development will positively impact on the Clang project

Conclusion:

it is often only in the process of writing that certain critical aspects of the work become apparent



from the Clang webpage:

(…)evidence of a significant user community should be provided: This is based on a number of factors, including an existing user community, the perceived likelihood that users would adopt such a feature if it were available, and any secondary effects that come from, e.g., a library adopting the feature and providing benefits to its users

some questions I ask you to think about:

what is my contribution to Clang-REPL?

have I achieved the key result I want to build my work around?

how do I achieve the key result

how will the project benefit from my contribution?

is there a possible application for it?

Conclusion:

This presentation shows opportunities to improve the working relationship between the development teams and the technical writing teams.

From a developer point of view, having a consistent and clear written documentation about your software will force you to get into the habit of putting abstract concepts into understandable words.

This improves the way you conceptualize and deal with technical problems, and you communicate those problems to others.

For all these reasons, technical writing should be considered as a key factor to contribute to the overall development and expansion of the Clang-REPL project.

Thank you for your attention!