

# Interactive and Collaborative Source Code Annotation

**Ryo Suzuki:** The University of Tokyo, Japan  
- Email: [1253852881@mail.ecc.u-tokyo.ac.jp](mailto:1253852881@mail.ecc.u-tokyo.ac.jp)  
- Website: <http://ryosuzuki.org>

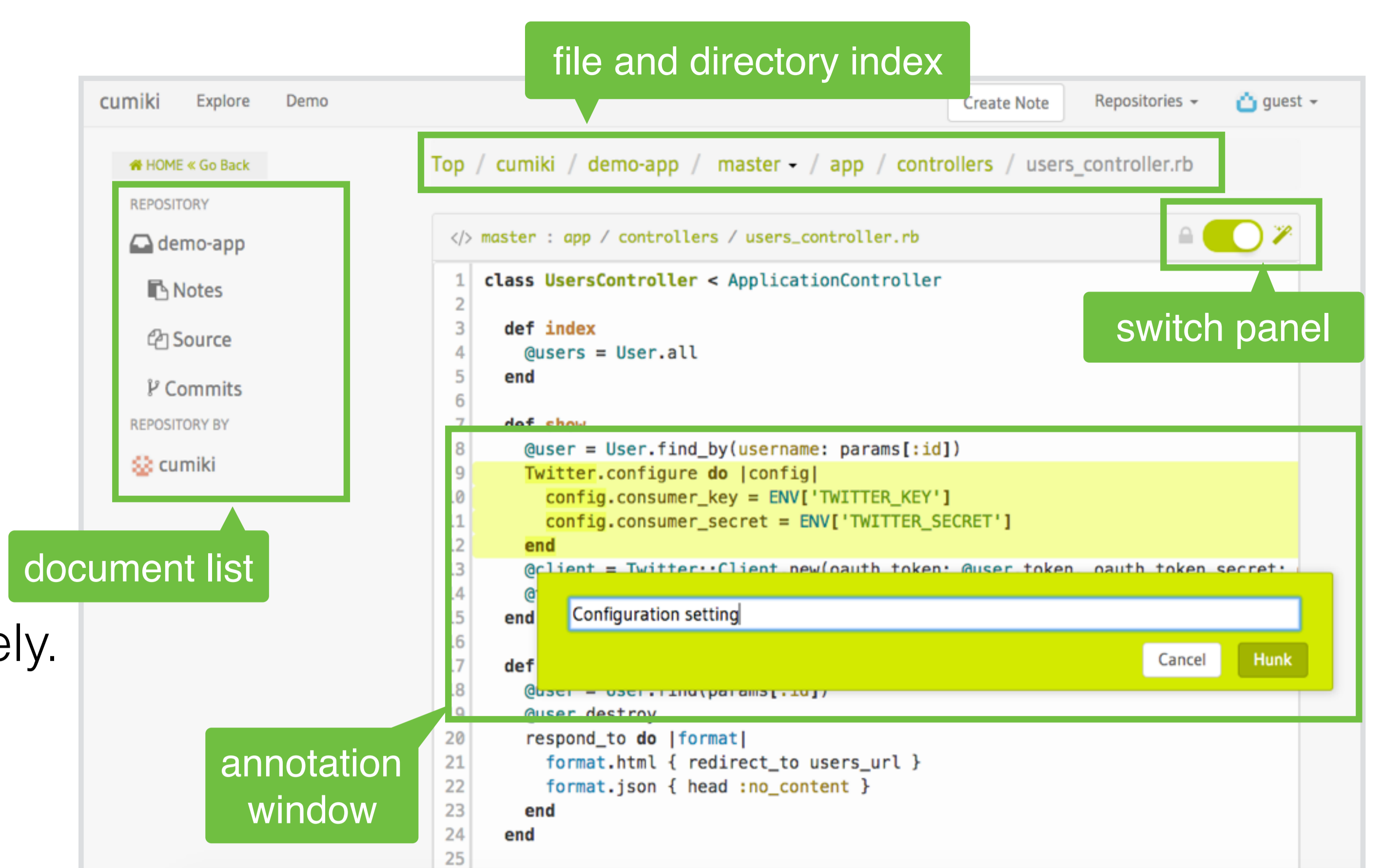
## Motivation and Background

### Problem:

Software documentation is helpful, but it needs to be constantly updated. Therefore, developers need a tool that makes it easier to make up-to-date documentation as well as to edit collaboratively.

### Related literature:

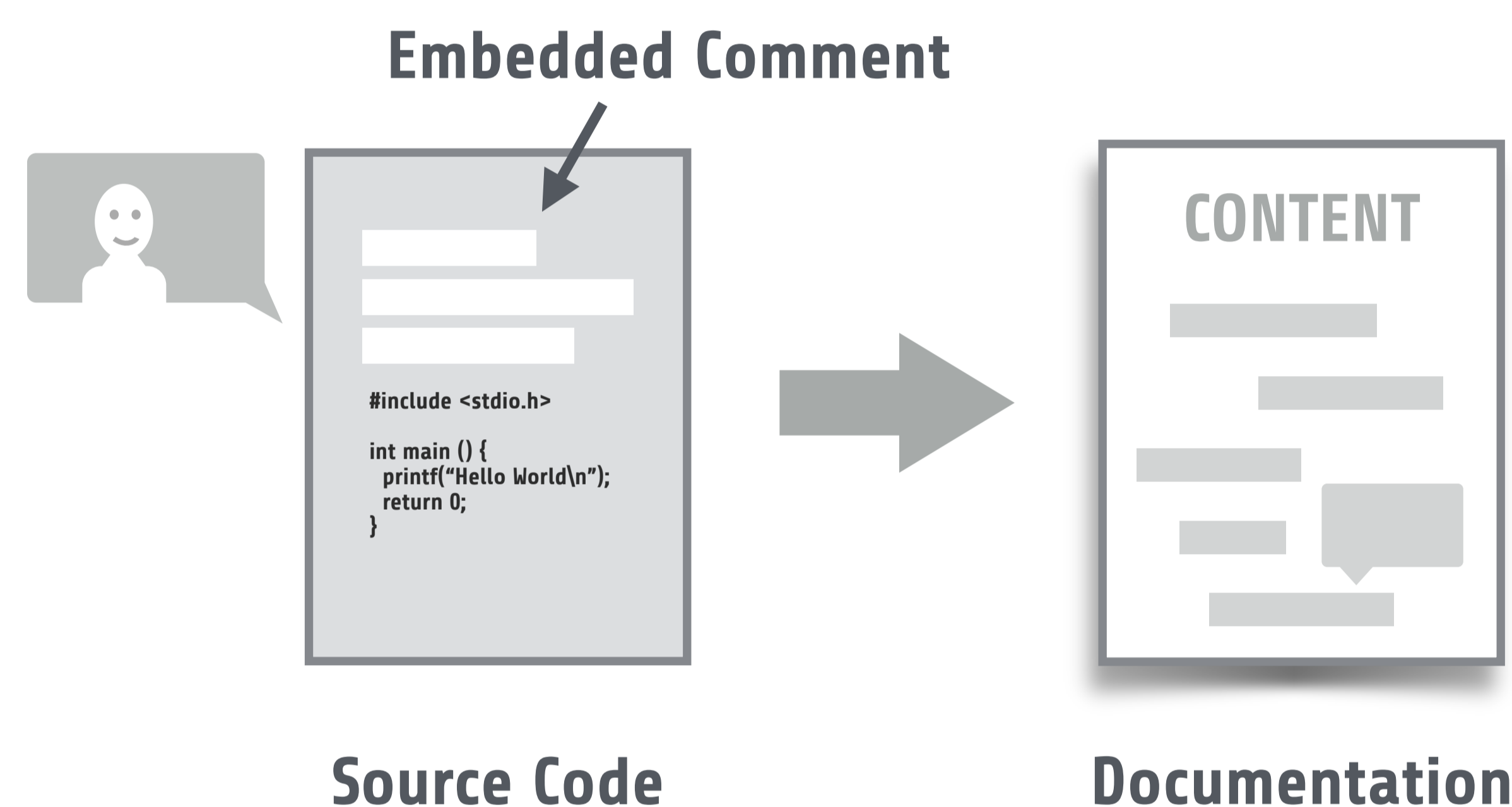
- Collaborative software development: Collabode [UIST'11]
- Crowdsourcing in software engineering: CrowdCode [UIST'14]
- IDE with web resources: Codelets [CHI'12], HyperSource [CHI'11]



User interface of Cumiki (<http://cumii.com/demo>)

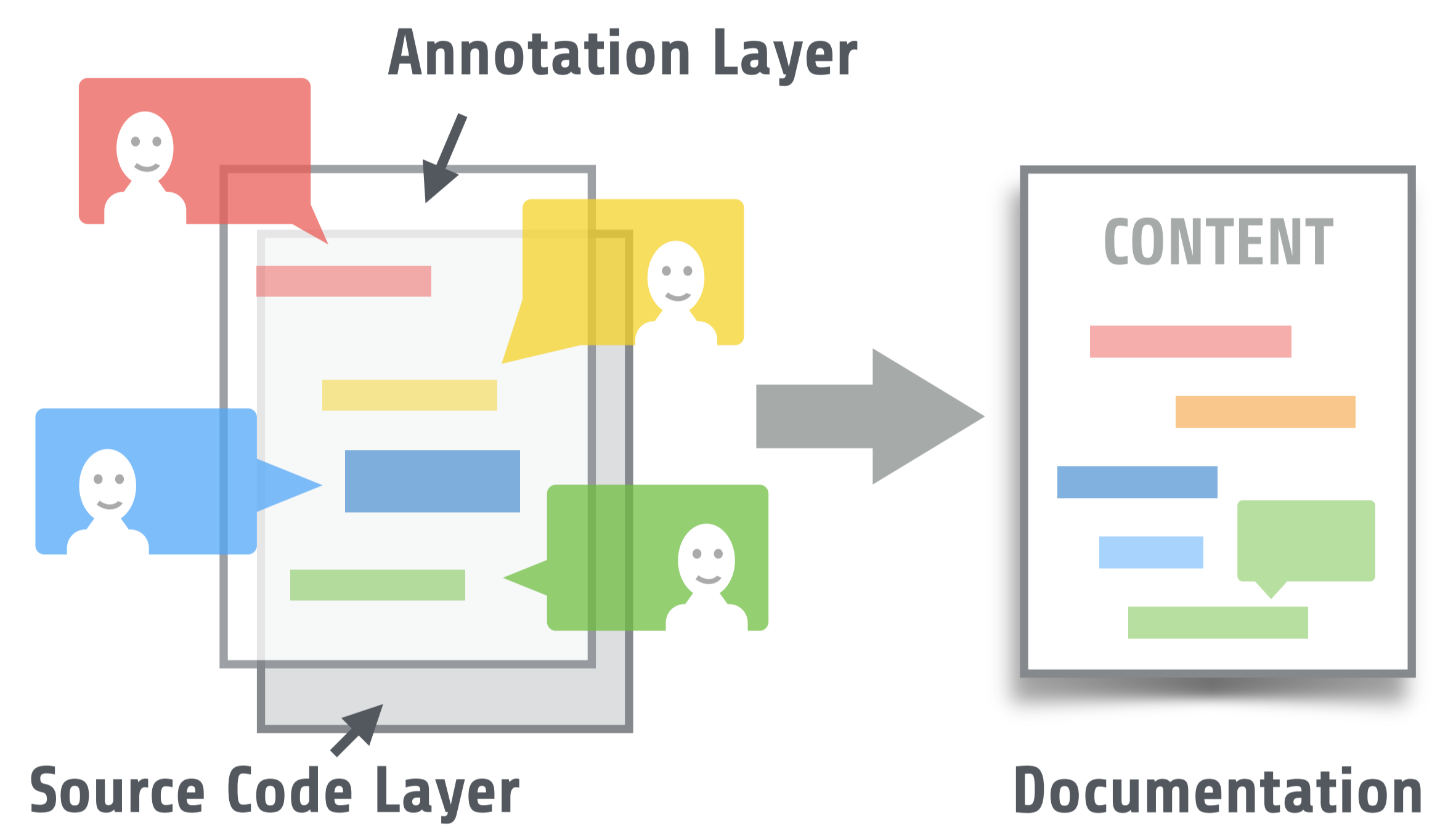
## Existing approach

Embedded documentation generator (e.g. RDoc, JSDoc) makes it easier to make up-to-date documentation, but developers cannot edit it without the owner's permission, thus it could hinder collaboration.



## Our approach

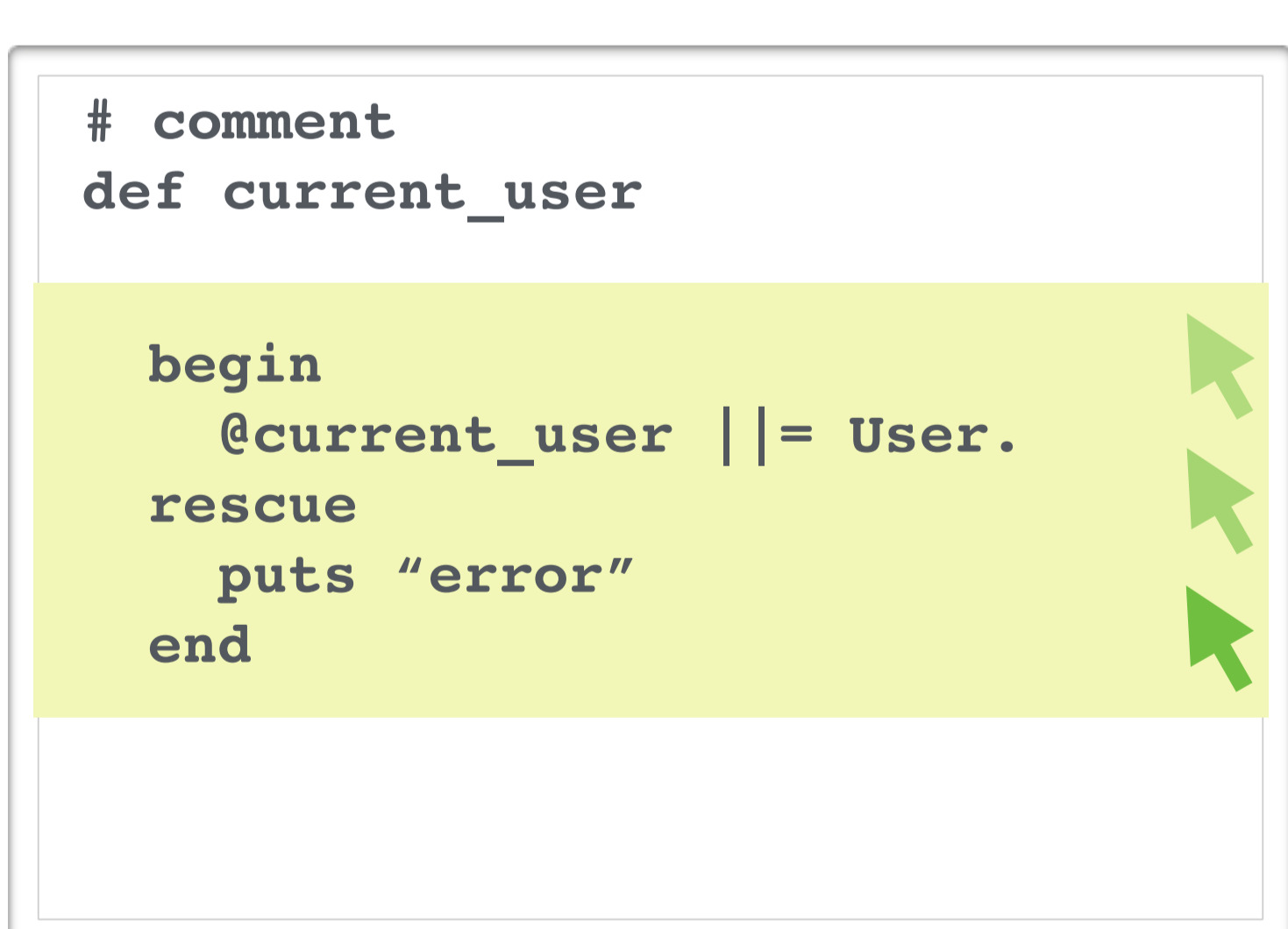
We propose another approach that separates the layer of annotation from the source code. Our approach makes it possible for crowds of developers to **annotate collaboratively** and **accumulate the knowledge** on the source code.



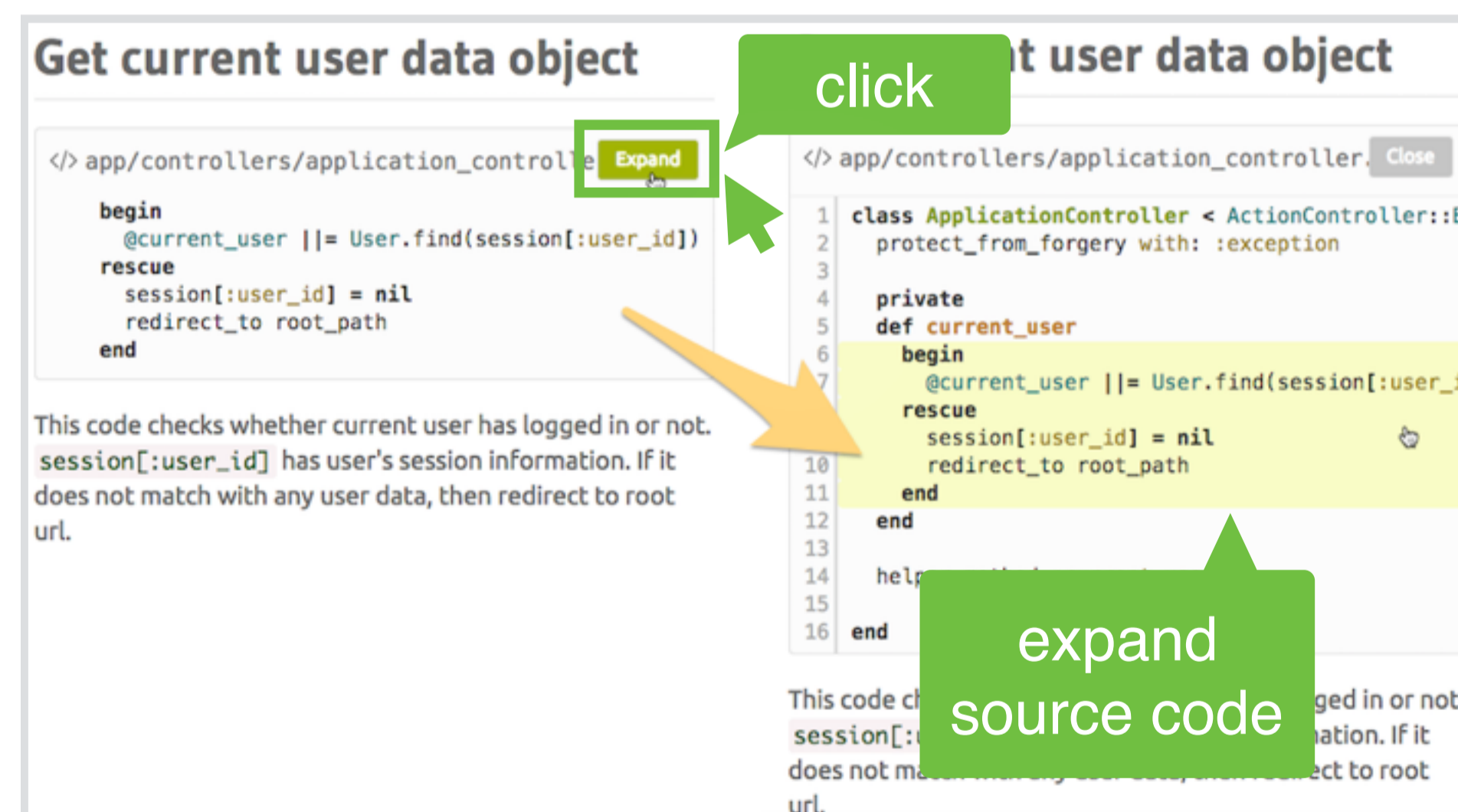
## Implementation: Cumiki, interactive and collaborative source code annotation

We introduce Cumiki, a web-based collaborative annotation tool. We implement the following three features:

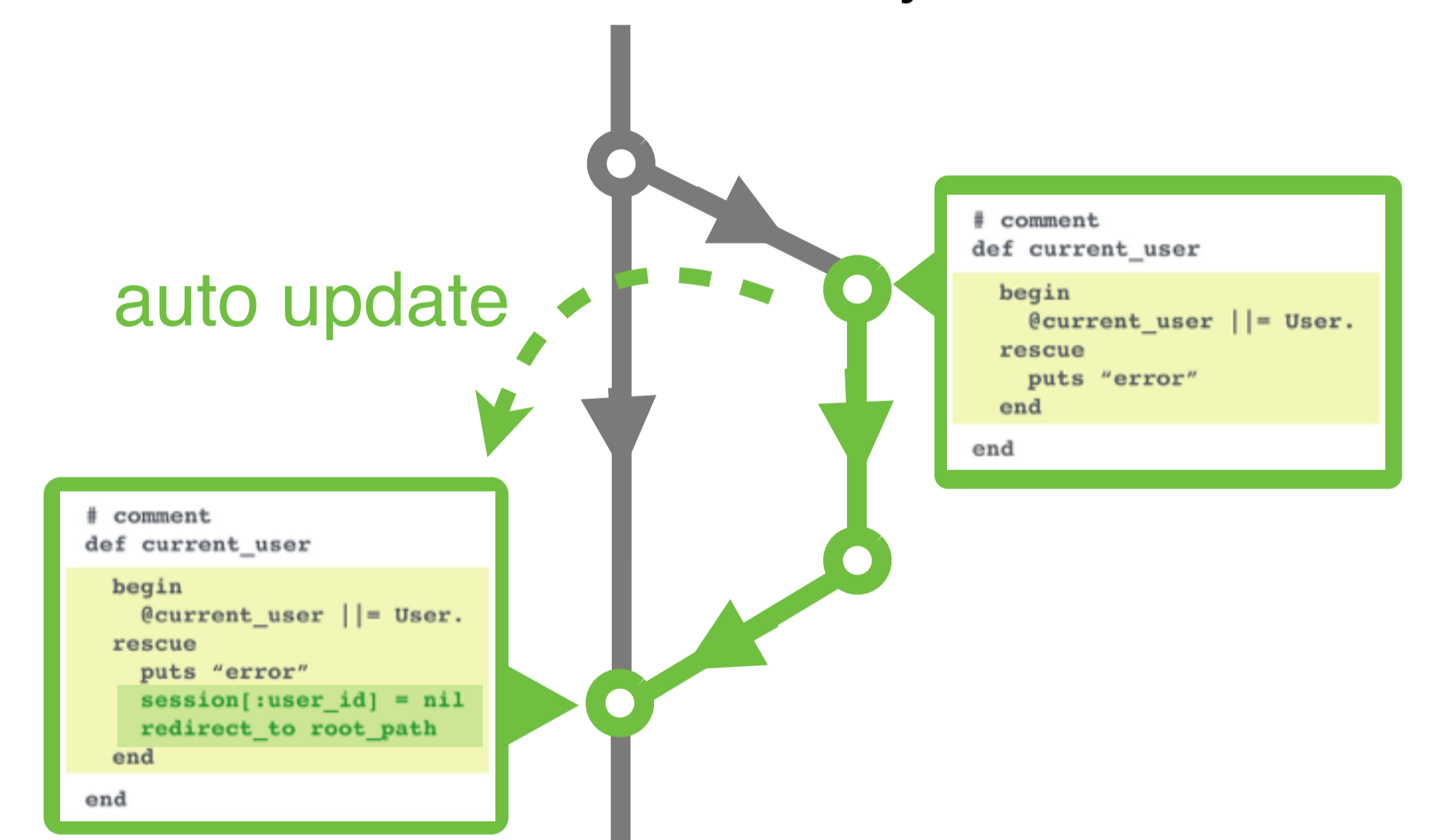
- (1) **interactive annotation:** annotate source code hosted on GitHub by simply dragging the mouse.
- (2) **source code traceability:** expand and highlight the annotated source code with one-click.
- (3) **automated updating:** analyze the history of change with git, and update the documentation automatically.



(1) mouse dragging annotation



(2) source code traceability



(3) automated updating with git

## Conclusion: possible usage scenarios and future work

We consider the following possible usage scenarios: (1) giving a comment on **a large open source project**, (2) accumulating and sharing the knowledge behind the source code within **a group or an organization**, and (3) creating a step-by-step tutorial for **computer science education**. As future work, we will evaluate our system with the user study related to these usage scenarios. To conclude, our contribution is to propose a concept of social and collaborative source code annotation and to explore it through the creation of our web-based annotation system.

## References

- M. Goldman, G. Little, and R. C. Miller. Real-time collaborative coding in a web ide. In Proc. of UIST'11, pages 155–164.
- B. Hartmann, M. Dhillon, and M. K. Chan. Hypersource: bridging the gap between source and code-related web sites. In Proc. of CHI'11, pages 2207–2210.
- T. D. LaToza, W. B. Towne, C. M. Adriano, and A. van der Hoek. Microtask programming: Building software with a crowd. In Proc. of UIST'14, pages 43–54.
- S. Oney and J. Brandt. Codelets: linking interactive documentation and example code in the editor. In Proc. of CHI'12, pages 2697–2706.