

# BSc./MSc. Competition Based Human Computation Framework for CrowdFlower.



## Problem statement

Human input is one of the most important data sources in computer science. Although advances in machine learning are making computers capable to solve more and more difficult problems, in some tasks human are still outperforming machines. Also, one of the key requirements for successful machine learning is a large amount of accurately labelled training data. Microtask crowdsourcing platforms like Crowdflower or Amazon Mechanical Turk offer a convenient and affordable way of obtaining contributions from a huge crowd of workers to a variety of tasks and competitions were proven very successful in this context. Both platforms offer a variety of templates that can be reused to natively implement common crowdsourcing tasks such as image labelling, audio transcription, and translations. However, the design of the crowdsourcing platforms assume a single worker and advanced crowdsourcing settings, such as competitions or collaboration with group interactions, are not natively supported. Therefore, there is a need for a crowdsourcing framework to conduct such crowdsourcing tasks on external platforms, while still leveraging the power of these platforms for recruitment and payment of workers.

## Thesis description

The goal of this thesis is therefore to extend and generalize our existing external crowdsourcing competition framework, to function as a base framework for implementing such external crowdsourcing applications on crowdflower. Specifically, users' accounts on the external platform need to be linked to their worker accounts on the crowdflower platform. After completing the tasks, based on their work on the external platform the workers need to be compensated afterwards via bonus payment. Although the crowdflower API offers methods for doing so, our crowdsourcing framework currently lacks the functionality to interact with it to perform the payment in a (semi-)automated manner. Further, for effective administration of the external application, an admin interface is required for monitoring and management of the task.

## Tasks and deliverables

In particular, following modules need to be designed, implemented and evaluated.

- Integration with the Crowdflower API: user management, payment management, etc.
- Admin-View: competition settings, user management, statistics view, etc.
- Payment management: money distribution, interaction logging, API
- Testing environment: test tasks, test payments

**Contact:** Dr. Sergej Zerr [szerr@l3s.de](mailto:szerr@l3s.de), MSc. Markus Rokicki [rokicki@l3s.de](mailto:rokicki@l3s.de)