Chapter 2
Cloud Labor Services

Despite all endeavors of rationalizing their business processes, organizations still face many types of tasks which cannot be fully automated but which require human intelligence or action to be completed. Cloud labor services provide human workforce on demand as a scalable service in order to meet the needs of organizations that deal with huge but varying workloads of such tasks.

After introducing the concept of cloud labor, identifying its characteristics and deducing key challenges in Sect. 2.1, Sect. 2.2 explores how cloud labor compares to related concepts. Section 2.3 then raises a series of challenges and general research questions. As the core contribution of the chapter, Sects. 2.4–2.5 provide an overview of the state of the art, organized into three perspectives: The perspective of the application benefiting from the cloud labor service, the perspective of the technical coordination platform and the perspective of the workforce that is represented by the people performing the actual work.

2.1 Overview

After putting the concept of cloud labor services into a historical context in Sect. 2.1.1, Sect. 2.1.2 formally introduces the basic concept of cloud labor in general and of cloud labor service specifically.

2.1.1 History

Cloud labor can be seen as a new form of outsourcing that has emerged driven by the advances in information and communication technology. While in general, "outsourcing is simply the farming out of services to a third party" (Overby n.d.), cloud labor ties in with two trends regarding the granularity of what is being outsourced and whom it is outsourced to. In its first wave in the late 1980s, the concept of

R. Kern, *Dynamic Quality Management for Cloud Labor Services*, LNBIP 192,
DOI 10.1007/978-3-319-09776-3_2

outsourcing was merely applied on a total business unit level, e.g. when Eastman Kodak outsourced its IT department to IBM, DEC and others (Hermes and Schwarz 2005, p. 17). Along with advances in business process re-engineering and business process management, the granularity of the outsourced services became finer, which resulted in the concept of business process outsourcing (BPO) (Hermes and Schwarz 2005, pp. 30–31). Cloud labor continues this development by further reducing the granularity down to "microtasks" which represent rather atomic manual tasks with a typical execution time in the order of minutes.

A parallel development concerns the growing number of outsourcing providers being involved. While traditionally large deals with only a small number of providers were closed, the development of the Internet and the Web 2.0 era opened the doors for outsourcing tasks to large groups of individual Internet users rather than to organizations. In his 2006 Wired–article, Howe (2006a) shaped the term crowdsourcing for this phenomenon which he later defined as the "act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call" (Howe 2006b). Kleemann et al. (2008) point out that crowdsourcing takes place "over the Internet".

From a computing perspective, passing work to a crowd of workers has already been a common approach for dealing with complex and extensive calculations long before the invention of electronic computers (Grier 2010). The largest computing office before the invention of the electronic computer has been the *mathematical tables project* started in 1938 in New York with 450 people working on mathematical calculations; the most renowned legacy of the project is the *Handbook of Mathematical Functions with Formulas, Graphs, and Mathematical Tables*, which was released in 1964 and which quickly became the "most widely circulated scientific text ever published" (Grier 2007, pp. 314–317). Originally, the term computer referred to a human being. In fact, the first *automatic* computers had the acronym "AC" in their names to distinguish them from human computers (Grier 2010). With the advances in computer technology, more and more tools have been developed that assisted humans in doing their work, as indicated by the emergence of terms like *computer aided design* (CAD) and *computer supported cooperative work* (CSCW). The degree of automation has increased step by step to a level where a large portion of business processes is fully represented and executed electronically. Of course, within those processes still a great number of tasks are performed by humans; nevertheless, to an ever-growing extent, the computer is setting the pace. This can be seen as a fundamental shift from *computer assisted humans* to *human assisted computers*.

2.1.2 Concept

Cloud labor is a specific form of crowdsourcing. The industry website Crowdsourcing.org (2012a) defines cloud labor as the act of "Leveraging of a distributed virtual labor pool, available on-demand to fulfill a range of tasks from simple to complex. Crowdsourcing is used to connect labor demand and supply. Virtual workers perform activities that range from simple to specialized tasks". A cloud labor platform is typically implemented as an electronic marketplace in the Internet or in an Intranet, on which a requester can submit work requests. Workers can browse work requests and choose according to their capabilities and interests what they want to work on. Depending on the business model of the platform, there may be a fixed compensation per task or a prize for submitting the best response. Commercial examples include MTurk[1] and oDesk[2].

According to Frei (2009), existing cloud labor platforms can be divided into two categories: Those that focus on well defined *tasks* and those that deal with rather complex *projects*. For the first category, there is usually a high volume of similar tasks to be performed that are issued in an automated manner. Because the execution time for those tasks is usually rather short, there is naturally a relatively low payment per task. In the second category of platforms, requesters typically only submit a small number of similar work requests using a manual process. Negotiation and task execution usually require a direct interaction between the requester and the worker. As the execution effort is higher and often more specific skills are needed, payment per tasks is usually much higher than on task platforms. There is no agreed terminology yet for the two categories of cloud labor in the scientific literature. For the tasks, often the term *microtask* is used which is rather misleading because they do not necessarily have to be small. The primary difference between tasks and projects is that tasks are formally described while this is not necessarily the case for projects. Thus, tasks can be completed by the worker without inquiries to the requester. Everything that is needed to complete the task is usually clearly stated in the task description. This book proposes the term *programmatically managed* (or short *programmatic cloud labor*) for this type of cloud labor. In contrast, the project type of cloud labor is called *manually managed* cloud labor. The terms will be motivated in the following. Table 2.1 provides an overview of the cloud labor platform landscape.

The general objective of cloud labor is to deliver scalable workforce to organizations that have to deal with a huge, but varying workload of manual efforts. However, scalability is not only a characteristic of the cloud labor platform, but also must be supported by the requester who must have enough bandwidth in order to publish requests, coordinate work and handle responses quickly enough. On the second category of platforms, on which projects are *manually managed* by the requester, the scalability with regard to the number and complexity of projects that can be handled is limited to the requester's bandwidth. From that point of view, the scalability characteristic is more pronounced for programmatically managed cloud labor, for which such a general limit does not exist.

This book focuses on programmatically managed cloud labor. However, to some extent the considerations can also be applied to manually managed cloud labor. The concept of programmatically managed cloud labor involves three roles:

[1] http://www.mturk.com, last accessed on July 1, 2013.

[2] https://www.odesk.com, last accessed on July 1, 2013.

Table 2.1 Cloud labor landscape addressing a range from well-defined microtasks to complex projects; based on Frei (2009).

Platform type	Platform examples	Task examples
Task marketplace with programmatic interfaces (*programmatically managed* cloud labor)	▷ Amazon Mechanical Turk ▷ Crowdflower ▷ Clickworker	▷ Find email addresses or company websites ▷ Translate a product description to another language ▷ Find prices for competitive products ▷ Choose a category from a new catalog structure ▷ Write a product review ▷ Test this website and provide feedback ▷ Fill in the missing research citations in this report ▷ Build a list of universities conducting energy research
Project marketplace for hiring and managing a virtual workforce (*manually managed* cloud labor)	▷ oDesk ▷ Elance ▷ TopCoder	▷ Design a branded website ▷ Prepare an outline for a conference presentation ▷ Contact all confirmed attendees for an event ▷ Program a software module ▷ Design a new edible adhesive ▷ Develop a new security algorithm ▷ Develop an eCommerce website ▷ Inbound/outbound calls (sales, mrkt research, support)

1. A *requester*, ideally represented by an electronic business process that comprises tasks which either cannot be automated today or for which automation would be too expensive or too time consuming.
2. A large group of suitable internal or external remote *workers* (crowd workers), acting as a virtual workforce, available to work on the tasks.
3. A web-based *coordination platform* (cloud labor platform), acting as an interface between the requester and the workers.

Figure 2.1 illustrates the basic concept of programmatically managed cloud labor. The service requester can be seen as a customer while the coordination platform is providing a service by means of the workers. The requester issues tasks as Web service calls via the coordination platform on which they can be browsed and processed by workers using a personal computer or mobile device. After completion of the task, the workers pass back responses, which are consolidated by the platform and finally

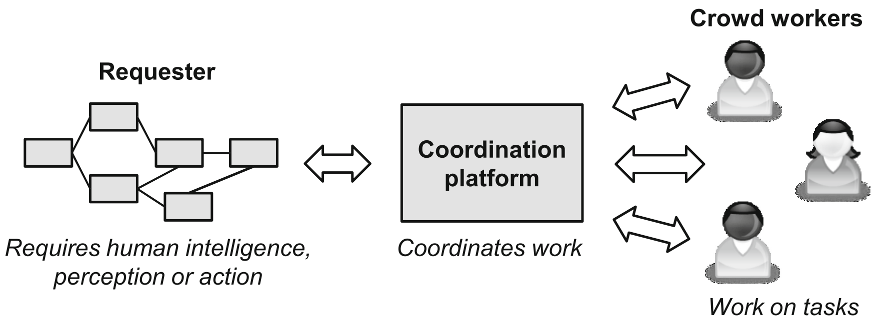

Fig. 2.1 Basic concept of programmatically managed cloud labor.

returned to the requester. Even though the tasks are issued electronically, the actual work does not have to be performed with the help of a computer and, therefore, may not only exploit the cognitive or intellectual capabilities, but also physical capabilities of the worker. For example, a worker may be asked to check the condition of a specific street lamp that had been reported to be defective. A cloud labor platform may deliver any type of work or action that humans are capable to perform. As the platform has only limited control over the remote workers, one obvious challenge is quality management. From the perspective of the business process, the cloud labor platform should deliver an automated service for which a well defined level of quality is guaranteed. It does not have to be visible to the business process that the service is actually provided by humans. Therefore, programmatically managed cloud labor provides additional flexibility when designing electronic business processes and they remove the need for a general distinction between automated and manual steps.

The scalability of cloud labor is enabled by the fact that there is a large group ("crowd") of workers available. An open call among the workers is used in order to ensure that new tasks are being promptly discovered by an adequate number of suitable workers and the requested responses are delivered in time.

Considering its service characteristic and its scalability, this book introduces the term *cloud labor services* for the services delivered by programmatically managed cloud labor. Cloud labor services are defined as "Web based services that deliver human intelligence, perception, or action to customers as massively scalable resources".

Because of the scalability characteristic, programmatically managed cloud labor requires a scalable implementation of the complete task invocation cycle comprising task submission, worker allocation, progress monitoring, quality control, payment and result integration. This cycle is indicated by Fig. 2.2. Scalable basically means that all the steps either need to be automated or recursively implemented as cloud labor services. Manual steps performed by employees in a traditional way would compromise the scalability of the whole approach, except if there is a sufficient number of employees available to handle the steps even in times of high workload.

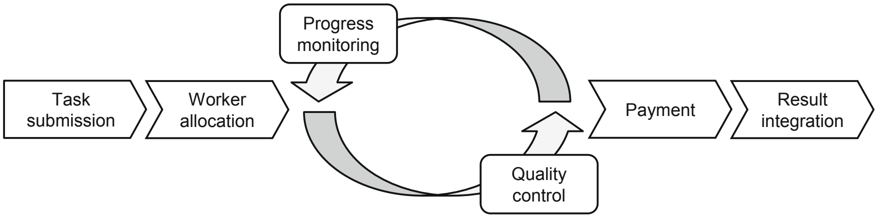

Fig. 2.2 Invocation cycle of a cloud labor service.

Obviously, there needs to be an incentive for the worker in order to spend time on cloud labor services. This incentive may be monetary as on existing cloud labor platforms like MTurk or non-monetary. An example for a non-monetary incentive are *games with a purpose* (von Ahn and Dabbish 2008), in which the tasks are designed as games and the workers (or *players*) perform work just for fun.

2.2 Related Concepts

This section describes, how the concepts of cloud labor and cloud labor services overlap with related concepts, namely crowdsourcing, human computation, social computing, open source, collective intelligence and also business process management (BPM), service oriented architecture (SOA) and cloud computing.

While there is a direct overlap with the scope of crowdsourcing, human computation, social computing and collective intelligence that is illustrated by Fig. 2.3, BPM and SOA complement the concept of cloud labor by providing technical concepts for their implementation. Cloud computing shares the basic objective of cloud labor to provide resources as scalable services but focuses on IT resources rather than on human workforce.

2.2.1 Crowdsourcing

Cloud labor represents a specific form of crowdsourcing, which, according to Sect. 2.1.1, is defined as the "act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call" (Howe 2006b). Apart from cloud labor, crowdsourcing.org (Crowdsourcing.org 2012a) segments crowdsourcing into the following concepts:

- *Crowd creativity*: "Tapping of creative talent pools to design and develop original art, media or content. Crowdsourcing is used to tap into online communities of thousands of creatives to develop original products and concepts, including

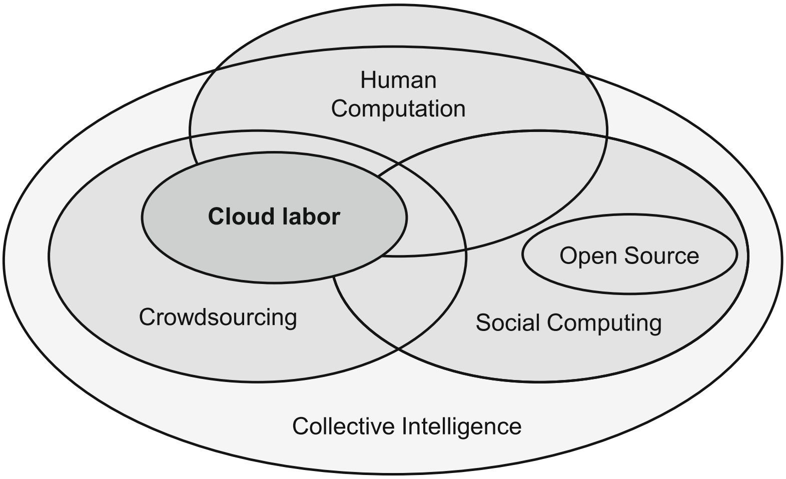

Fig. 2.3 Overlap of the cloud labor concept with similar concepts; based on Quinn and Bederson (2011).

photography, advertising, film, video production, graphic design, apparel, consumer goods, and branding concepts". Prominent examples include *99 designs*[3], a platform that hosts public design contests, and *Spreadshirt*[4], a platform for the crowdsourced design of personalized apparel.

- *Crowd knowledge*: "Development of knowledge assets or information resources from a distributed pool of contributors. Crowdsourcing is used to develop, aggregate, and share knowledge and information through open Q&A, user-generated knowledge systems, news, citizen journalism, and forecasting". This segment covers traditional crowdsourcing applications like Wikipedia[5] and also the concept of the "wisdom of crowds" introduced by Surowiecki (2004). According to him, when aggregating imperfect intuitive judgements of a group of people in the right way, its collective intelligence is often excellent. An example for commercial use of the concept is Ask500People[6], a website that gathers feedback from hundreds of independent voters in minutes.
- *Open innovation*: "Use of sources outside of the entity or group to generate, develop and implement ideas. In a world of widely distributed knowledge, where the boundaries between a firm and its environment have become more permeable, companies cannot afford to rely entirely on their own research and ideas to maintain a competitive advantage". The term open innovation was originally introduced by Chesbrough (2003). An example application is InnoCentive[7], a problem

---

[3] http://99designs.com/, last accessed on July 1, 2013.

[4] http://www.spreadshirt.com/, last accessed on July 1, 2013.

[5] http://www.wikipedia.org, last accessed on July 1, 2013.

[6] http://www.ask500people.com/, last accessed on July 1, 2013.

[7] http://www.innocentive.com, last accessed on July 1, 2013.

solving marketplace on which organizations can publish challenging problems. Any Internet user can propose solutions (Lakhani et al. 2007).

Crowd labor differs from the other segments regarding its scope and regarding the way the quality of the work results is being managed. Crowd creativity, crowd knowledge and open innovation are represented by targeted applications implemented as web-based communities which are led by shared goals. Members of the community are contributing assets like texts, photographs, designs or ideas which are then evaluated or gradually improved by peers. The quality of an individual contribution may be questionable, but because many community members are interacting, there is an implicit quality control which sorts out inferior contributions or gradually improves them. In contrast, cloud labor is represented by all-purpose platforms that can be utilized for any type of remote work. The quality of the work results needs to be managed explicitly depending on the type of the application and according to the requester's requirements. Crowd creativity, crowd knowledge and open innovation may be applications of cloud labor, but they do not necessarily have to be. For example, Wikipedia belongs to the segment of crowd knowledge but does not represent cloud labor, whereas 99designs could be seen as both, cloud creativity and cloud labor. As indicated by Fig. 2.3, cloud labor can be seen as a sub-concept of crowdsourcing. Like crowdsourcing, cloud labor outsources work to a crowd of people and utilizes an open call for allocating workers to tasks.

2.2.2 Paid Crowdsourcing

The term *paid crowdsourcing* shaped by Frei (2009) addresses those forms of crowdsourcing in which a monetary incentive is used. From that point of view it is orthogonal to the segments defined in the previous section. Frei defines paid crowdsourcing as "the act of outsourcing paid work of all kinds to a large, distributed group of workers using a technology intermediary that helps oversee the definition, submission, coordination, acceptance and payment for the work done". Note that this definition is actually broader than crowdsourcing because it does not necessarily require an open call. Therefore, it is rather inconsistent with the definition of crowdsourcing. As the term is only rarely used in the scientific literature, it is omitted from Fig. 2.3.

2.2.3 Human Computation

Quinn and Bederson (2011) argue that the modern usage of the term human computation was inspired by von Ahn's 2005 dissertation of the same name. Von Ahn defines human computation as "a paradigm for utilizing human processing power to solve problems that computers cannot yet solve" (von Ahn 2005). By taking into account definitions from a series of other papers, Quinn and Bederson (2011) come to the conclusion that there is a consensus about the following characteristics of human computation:

- "The problems fit the general paradigm of computation, and as such might someday be solvable by computers."
- "The human participation is directed by the computational system or process."

The authors explicitly point out that traditional crowdsourcing applications like Wikipedia are excluded because Wikipedia was designed as a "collaborative writing project" and not to replace a machine.

It can be concluded that one major difference between human computation and cloud labor is that human computation is about dealing with *computational problems* while crowdsourcing and specifically cloud labor is dealing with *work* in general. Out of the three human capabilities covered by the definition of programmatically managed cloud labor in Sect. 2.1.2 only "intelligence" matches with the idea of human computation while "perception" and "action" do not because they do not represent computational problems. There are several types of applications which satisfy the definition of programmatically managed cloud labor but not the one of human computation: Filling in a survey or a poll can be regarded as work, but it does not represent a computational problem. Another example is any form of physical activity like asking a worker to take a picture of a specific building and send it back to the requester. Therefore human computation is narrower than cloud labor, but at the same time it is broader: While cloud labor is limited to the field of crowdsourcing, human computation comprises any scenario in which one or more humans are working on a computational problem and are directed by a computational system or process. There is neither a crowd required nor an open call. Figure 2.3 indicates the overlap between cloud labor and human computation.

Games with a Purpose

For programmatic cloud labor there is another difference that concerns the incentives being used to attract the workers. While programmatically managed cloud labor always relies on a financial incentive, human computation does not necessarily do so. It explicitly comprises the concept of *human computation games* or *games with a purpose* (GWAP) (von Ahn and Dabbish 2008) that "constructively channel human brainpower using computer games" (von Ahn 2005). In this concept, people are playing a game just for fun, but implicitly they are actually performing work. Indeed, the early works on human computation have been in the area of games. The most noted one is surely the ESP game developed by von Ahn and Dabbish (2004). When Luis von Ahn originally shaped the term human computation, it was entirely focusing on unpaid contributors (von Ahn 2005).

In order to illustrate the idea of human computation games, the ESP game is described here in more detail, following von Ahn and Dabbish's (2004) original paper. It is played by two players over the Internet, who are randomly paired by being picked from a pool of available players. The same random picture is shown simultaneously to both players and they are asked to guess what words come into the mind of the partner player and type them in. As there is no way to communicate, the only commonality between the players which they are aware of is that they are looking at the same picture. So naturally, they type in words that describe what they see on the picture. For each match they both earn a certain number of points. The actual objective of the game is to attach meaningful labels to images, e.g. “cat”, “black”, “sitting”. ESP stands for *extra sensory perception* and reflects the idea that the players have to “think like each other” (von Ahn and Dabbish 2004).

The concept had been licensed by Google and has been extremely successful as the *Google Image Labeler* which was available in Google Labs for many years. As of July 2008, more than 200,000 images had been labeled by some 50,000 players (von Ahn and Dabbish 2008). In 2011 the service was discontinued when Google decided to close down the test bed Google Labs (Google Inc. 2011).

2.2.4 Social Computing

Social computing applications comprise “blogs, wikis, social bookmarking, peer-to-peer networks, open source communities, photo and video sharing communities, and online business networks” and can be characterized as “applications and services that facilitate collective action and social interaction online with rich exchange of multimedia information and evolution of aggregate knowledge” (Parameswaran and Whinston 2007).

According to Quinn and Bederson (2011), the “key distinction between human computation and social computing is that social computing facilitates relatively natural human behavior that happens to be mediated by technology, whereas participation in a human computation is directed primarily by the human computation system”. The same distinction can be also made for cloud labor services because they are actively directed by a platform rather than by natural human behavior. The idea of social computing can be combined with the concept of cloud labor by building social networks of workers that share knowledge and jointly work on tasks in order to build synergies, gain better results and have more fun than when working alone.

2.2.5 Open Source

Open source software development is usually driven “by a large voluntary community” and represents a “revolutionary process of producing software based on unconstrained access to source code as opposed to the traditional closed and property-based approach of the commercial world” (Bonaccorsi and Rossi 2003). Open source can be seen as a sub-concept of social computing (Parameswaran and Whinston 2007), but it is usually not considered to be crowdsourcing. One fundamental difference between the concepts lies in the motivation of the contributors. While contributions to open source projects are usually motivated by the intrinsic reward of successful problem solving and gaining recognition within the team, there may also be monetary incentives for contributing to crowdsourcing activities (Brabham 2008a). For the same reason, there is no overlap between cloud labor and open source in Fig. 2.3. As indicated, open source also differs from human computation. While the participation in open source projects is community driven (Lakhani and Von Hippel 2003), human computation applications are directed by a computer.

2.2.6 Collective Intelligence

While Lévy (2001) describes collective intelligence as "intelligent communities, as open-minded, cognitive subjects capable of initiative, imagination and rapid response", a recent definition by Malone et al. (2009) very broadly defines it as "groups of individuals doing things collectively that seem intelligent." As illustrated by Fig. 2.3, collective intelligence represents the basic principle and "breeding ground" behind all three, crowdsourcing, social computing and human computation. The latter one is only included as far as a group of humans is involved, which is always the case for social computing and crowdsourcing. By being a specific form of crowdsourcing, cloud labor also represents a subset of collective intelligence.

2.2.7 Human Tasks in Business Process Management and SOA

Scheer and Brabänder (2010) define BPM as "a structured approach employing methods, policies, metrics, management practices, and software tools to coordinate and continuously optimize an organization's activities and processes". From a technical perspective, BPM is closely related to the concept of SOA that is defined as "a paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains" (OASIS 2006).

Within the fields of BPM and SOA, there are a series of IT concepts like *Web Service human task* (WS-HumanTask) (Agrawal et al. 2007a) and *business process execution language for people* (BPEL4People) (Agrawal et al. 2007b) that support the orchestration of human activities as components of service oriented architectures or electronic business processes. Like cloud labor services, these concepts allow for the provisioning of work through electronic interfaces. By focusing on technical interfaces, language design and protocols, they complement the idea of cloud labor services as they offer a technological framework for the implementation of human tasks.

The biggest difference lies in the variety of tasks, in the way they are assigned to a workforce and in the type of the workforce. Cloud labor service tasks are made available to a rather large and undefined crowd of people who choose from a large variety of tasks based on their capabilities and desires. In contrast, WS-HumanTask and BPEL4People use a role based assignment of tasks to employees in an organizational environment. Depending on well defined responsibilities, tasks are assigned to specific employees or groups of employees who complete them as part of their daily business. There is typically only a small set of task types that may be processed by a specific person, i.e. there is a higher degree of specialization than for cloud labor services. Because there is more control over the workforce and there are fewer degrees of freedom, aspects like quality management, motivation and incentives play a less important role than for cloud labor services. Nevertheless, human tasks in BPM and SOA can likely profit from the considerations made in this book while cloud labor services can benefit from the technological framework of human tasks in BPM and SOA.

Human provided services (HPS) (Schall 2011) go one step further than WS-HumanTask and BPEL4People by providing a framework for modeling human expertise and supporting its discovery and provisioning in service-oriented environments. HPS can be seen as a technical framework for implementing cloud labor services. Besides the definition of protocols and interfaces, HPS also provides an interaction model for structuring and delegating work.

2.2.8 Cloud Computing

Cloud computing can be defined as "a model for enabling ubiquitous, convenient, on-demand network access to a shared pool of configurable computing resources (e.g., networks, servers, storage, applications, and services) that can be rapidly provisioned and released with minimal management effort or service provider interaction" (Mell and Grance 2011).

According to Mell and Grance, the fundamental characteristics of cloud computing are:

- It provides "on demand self-service", i.e. there is no human interaction required with the service provider.
- The service can be used through "broad network access" using standard computing devices like laptops or mobile devices.
- A "resource pooling" mechanism serves multiple clients by dynamically assigning resources to them depending on the demand.
- Capabilities are provided with "rapid elasticity", i.e. the resources are provisioned and de-provisioned in a scalable way.
- It represents a "measured service", i.e. a metering capability monitors, controls and reports the resource usage per client.

There is a fundamental symmetry between cloud labor services and cloud computing. Basically, cloud labor services are for human workforce what cloud computing is for IT services. According to Sect. 2.1.2, all the characteristics of cloud computing apply to cloud labor services as well: Cloud labor services are issued through a programmatic interface and delivered by a pool of workers who are dynamically allocated to the appropriate tasks. This results in a highly scalable service. Efforts are tracked and billed per requester. The overlap between the two concepts will be further discussed in Sect. 2.5.2 when considering service models for cloud labor services.

2.3 Challenges

The concept of cloud labor poses a series of challenges that concern a wide variety of disciplines. In this chapter, the key challenges are identified and general research questions are deduced. Because of the overlap with the fields of crowdsourcing and human computation, the challenges can be derived from those research fields.

In order to structure requirements for crowdsourcing and human computation, Alonso (2011) proposes a three role perspective similar to the roles of the basic cloud labor service concept: The "experimenter view" represents the perspective of the requester who is designing the tasks, the "engine view" represents the technical platform and the "human view" represents the perspective of the workers. In this book, more generic terms are used that better match the terminology of the cloud labor services concept: Application view, platform view and workforce view.

For each of the perspectives, the subsequent Sects. 2.3.1–2.3.3 describe the key challenges along with corresponding research questions. Table 2.2 serves as a chapter overview by summarizing the challenges clustered by the three perspectives.

Later in Chap. 2, Sects. 2.4–2.6 will again take up the three perspectives and describe to what extent the corresponding challenges have already been addressed by the state-of-the-art.

2.3.1 Application Challenges

The application view addresses the identification of relevant applications and tasks, the decomposition into manageable work units and the recomposition of the work results as well as privacy, copyright and compliance considerations.

Identification of Relevant Tasks

For the requester, one of the initial challenges is to identify relevant tasks to which the cloud labor concept can be applied and how the cloud labor services can be integrated into existing business processes or combined with automated tasks. Often, at least a portion of the task can be automated so the question is how to define the manual portions and their interplay with the automated portion (Shahaf and Amir 2007). Corresponding research questions are:

Table 2.2 Challenges of the cloud labor service concept structured by the perspective of the application, the platform and the workforce.

Perspective	Challenge	Source (selection)
Application	Identification of relevant tasks	(Shahaf and Amir 2007)
	Workflows & task granularity	(Kittur et al. 2011)
	Privacy, copyright & compliance	(Felstiner 2011)
Platform	Deployment & service models	
	Worker to task matching	(Ambati et al. 2011; Yuen et al. 2011)
	Crowd management	(Schall 2011)
	Quality management	(Snow et al. 2008; Kittur et al. 2013)
	Technical infrastructure	(Alonso 2011; Schall 2011)
	Interfaces & usability	
Workforce	Motivation & incentives	(Kaufmann et al. 2011)
	Education & feedback	(Mason and Watts 2012)
	Work model	(Felstiner 2011; Silberman et al. 2010)

1. What are suitable types of tasks "that are easy for humans and hard for computers" (Shahaf and Amir 2007), i.e. what tasks can be performed by humans in a better or cheaper way than by computers (Alonso 2011)?
2. How can automatic tasks and manual tasks be combined in order to gain the optimal synergies? (Alonso 2011)
3. How can cloud labor services extend existing business processes? (Vukovic et al. 2010)
4. What are suitable business cases for applying the cloud labor concept? How can it be applied to the enterprise scenarios (Alonso 2011; Vukovic et al. 2010)?

Workflows and Task Granularity

Once a task has been identified, the second challenge is to adjust the granularity of the tasks to a level that can be handled by individual workers. Therefore, workflows need to be defined which decompose complex tasks into smaller ones and recompose the responses returned from the workers (Kittur et al. 2011). Relevant research questions are:

1. How can a complex task be decomposed into smaller entities that can be handled by a human in an acceptable amount of time?
2. How can the results of the individual sub-tasks be reassembled in order to gain a consolidated result?
3. What are the decision criteria for deciding whether to decompose a task or not?

Privacy, Copyright and Compliance

Another important challenge is to address potential privacy, copyright and compliance needs. In general, all the tasks and responses exchanged on a cloud labor service platform represent information that may be subject to data privacy protection or may represent business secrets. Even if the requesters add appropriate privacy policies to their participation agreements "it would be naive for firms to count on those agreements, or on vendors in general, to protect intellectual property" (Felstiner 2011). Also, data protection acts may prohibit the disclosure of specific types of information either in general or restrict the use of the data, for example with regard to the countries it may be transferred to. A third challenge is that information generated or researched by workers within the scope of a cloud labor service (e.g. in text authoring or image retrieval applications) may be subject to copyright. Corresponding research questions are:

1. How can the privacy of the requesters and their customers be guaranteed even when dealing with confidential information?
2. How can the privacy of workers be guaranteed given that the cloud labor service provider or requester might gather a lot of information about the individual workers over time?
3. How can the platform or the requesters respect the copyright of texts and multimedia data that workers research from the Internet?

2.3.2 Platform Challenges

The platform view concerns the technical and economical challenges concerning the cloud labor platform including deployment models, quality management mechanisms, the design of the requester and worker interfaces, the technical architecture of the platform as well as the underlying protocols and standards. Another aspect is the matching and allocation of workers to tasks and tasks to workers.

Deployment and Service Models

Similar to cloud computing, various deployment and service models can be conceived for cloud labor platforms with respect to the level of service they deliver and the type of workforce they rely on. The platform may be operated by the requester or by an external company. The definition and tailoring of the tasks and the management of the result quality may be performed by the requester, by the platform owner or again by a third party service provider. The workforce may be provided by the requester, by an external organization or it may consist of freelancers or Internet users who want to earn some extra money in their spare time. There may also be a combination of platforms and workforces that are interconnected in some way. Corresponding research questions are:

1. What are viable deployment models for task market places and how do they relate to the corresponding models of cloud computing?
2. What are viable service models of cloud labor service platforms with respect to the workforce they rely on?
3. How can multiple cloud labor platforms and workforces be combined in an effective way?

Worker to Task Matching and Allocation

One of the key benefits of cloud labor is that it provides access to a huge pool of skills, experiences, cultures, beliefs, opinions, habits and desires that makes it applicable to almost any kind of human work one can think of, assuming that it can be performed or at least initiated electronically. However, an effective worker to task matching and allocation is needed in order to ensure that requesters find the right workers for their tasks and workers find appropriate tasks to work on. Relevant research questions are:

1. How can tasks be effectively matched to workers while taking into account their individual preferences, skills, experience, strengths and weaknesses? (Law and Ahn 2011, p. 6)
2. How can task properties and worker characteristics be formally modeled in a comprehensive way that supports matching of tasks to workers?
3. How can a taxonomy of skills be designed that captures relationships and proximity between skills?

Crowd Management

Especially if an organization relies on its own workforce for delivering cloud labor services, it has to be carefully managed in order to ensure that the employees are fully utilized but not overloaded, that their expertise is leveraged and that they are happy with the work conditions. But also for marketplaces that rely on Internet users, a proper crowd management is an important instrument for establishing and maintaining a powerful and loyal workforce. Relevant research questions are:

1. What mechanisms allow for an effective workload and skill management on task platforms? (Alonso 2011)
2. What tools can facilitate an effective communication with the crowd workers?

Quality Management

Quality management can be regarded as one of the key challenges of cloud labor services. Kittur et al. (2013) report that it has “arguably received the most attention so far”. Because of the anonymity of the workforce, there is only little control over the quality of the work results delivered by an individual worker (Suri et al. 2011). This becomes particularly obvious when dealing with a workforce of Internet users. Relevant research questions are:

1. How can a crowd of remote workers be coordinated in a way that satisfying work results can be achieved with minimal effort?
2. What factors are affecting the quality of work results and what are suitable quality metrics?
3. How can reputation or trust be utilized in order to predict the quality of work results delivered by a worker?
4. How can fraud be discovered or prohibited?

Technical Infrastructure, Protocols and Standards

The concept of cloud labor is enabled by information technology and most of the characteristics of the cloud labor platform are basically represented by IT. This applies to the architecture and design of the cloud labor platform, the computing, storage and network infrastructure as well as the underlying protocols and standards. Other important aspects are the network bandwidth and the computing devices that can be expected on the worker side. Related research questions are:

1. How can a reliable and highly scalable cloud labor platform be designed?
2. How can existing protocols and standards be enhanced in order to address the specific needs of cloud labor applications? What protocols and standards allow for integration with other enterprise systems (Alonso 2011)?
3. How can skill profiles and work descriptions be represented in a standardized way?

Interfaces and Usability

The acceptance of cloud labor marketplaces will not at least depend on their interfaces and usability. This is not limited to a beautiful design of the screens but primarily concerns the requester and worker experience as a whole. Just because cloud labor is IT based, it does not mean that all activities have to be performed in front of a computer. Maybe other devices can be introduced or workers can organize into teams who even cooperate physically and just submit their results on a computer in the end. Relevant research questions are:

1. How should the interfaces be designed in order to make the interaction with the cloud labor service platform as easy, unambiguous, encouraging and efficient as possible for all involved parties? (Alonso 2011)

2. What interfaces could support the virtual or physical cooperation of workers? (Law and Ahn 2011, p. 6)

2.3.3 Workforce Challenges

The workforce view comprises the motivation of the workers by means of appropriate incentives (e.g. payment), the importance of education and feedback, the overall acceptance of the work model as well as contracting and fiscal considerations.

Motivation and Incentives

One of the basic principles that cloud labor services inherit from crowdsourcing is that work requests are published in an open call. It is up to the worker to decide what projects or tasks to work on. Thus, appropriate incentives need to be in place that attract suitable workers. Corresponding research questions are:

1. What are suitable incentives and incentive schemes that motivate workers to contribute continuously and with high quality? (Law and Ahn 2011, p. 6)
2. What combinations of incentives are feasible and can leverage additional advantages, e.g. how can monetary incentives be mixed with non-monetary ones in order to save costs?
3. What incentive schemes can be applied to a private workforce of employees that anyway receive a more or less fixed salary? (Vukovic et al. 2010)

Education and Feedback

The capabilities of the workers are obviously one of the most important prerequisites for a successful application of the cloud labor concept. This involves fundamental skills in relevant disciplines but may also comprise specific skills about the project. Relevant research questions include:

1. How can education mechanisms be designed that seamlessly integrate into the concept of cloud labor?
2. How can feedback mechanisms be designed that support a continuous improvement of the worker skills?
3. What effect do feedback mechanisms have on the quality of the work results and on the worker satisfaction?

Work Model

The concept of cloud labor differs significantly from traditional employment models. On the one hand, it may lead to advantages as it may reduce the barriers to find a job and increase the flexibility regarding the work schedule and work location; on the other hand, there may also be the risk of jeopardizing traditional work models with regard to lay-off protection, minimum wages and pension funds (Felstiner 2011). Furthermore, legal contracts and taxes are subject to national laws and regulations. In contrast, as it is based on the worldwide Internet, the concept of cloud labor is not limited to specific countries. That implies a need for research on the consolidation of national employment and fiscal laws into a global perspective. Relevant research questions include:

1. Under what conditions can the cloud labor concept be a viable work model that successfully complements traditional work models?
2. How can the social acceptance of the cloud labor concept be fostered? (Kittur et al. 2013)
3. How may the existing national laws and tax models be adapted to the needs of the globalized perspective of the cloud labor concept?

2.4 Application Perspective

This section describes the state of the art of cloud labor services from the perspective of the usage scenario. After providing an overview on the existing types of applications in Sect. 2.4.1, Sect. 2.4.2 explains how relevant applications may be identified. Section 2.4.3 then covers concepts and tools for defining task workflows as well as for decomposing tasks into smaller portions of work and recomposing the results. Finally, Sect. 2.4.4 addresses some considerations regarding privacy, copyright and compliance.

2.4.1 Existing Applications

This section classifies existing cloud labor service applications in two ways. It examines what applications public cloud labor platforms are actually used for as well as what applications have been investigated in the scientific literature.

Applications on Public Cloud Labor Platforms

In order to develop an understanding of the types of applications being implemented on public cloud labor platforms a classification has been created within the scope of this work based on a manual analysis of all available tasks on the Amazon Mechanical Turk (MTurk) platform[8] in August 2009. The tasks have been categorized by the

[8] See Sect. 2.5.1 for a comprehensive description of the platform.

action the worker is asked to perform. Table 2.3 presents the resulting classification. The classification consists of five main categories, with each one comprising two to five application types. The category *Create content* comprises any applications in which workers are asked to create textual or multimedia content or provide creative ideas. Category *Revise content* addresses all applications in which workers are asked to classify, refine, summarize or transform content. This includes the identification of pornographic or other undesired content. It also applies to the recognition of objects on images or the transcription of recorded speech. In *Research information* applications, workers are asked to locate and potentially retrieve information or multimedia data from the Internet. In the applications of category *Rate and sense*, users are asked to provide subjective feedback, for example by responding to surveys or polls or by telling about their individual experiences with products or websites. The last category covers all types of applications that do not fall into any of the other categories. At the time of the analysis, the major portion of tasks in that category applied to translation, software and Web application testing. The category also covers location specific applications, in which the worker has to be at a specific location in order to provide the service, for example by taking a picture of a specific event. A remarkable application in this category is *Traffic augmentation*, in which workers are asked to generate traffic or to provide positive feedback in the Web 2.0 environment (e.g. rating a YouTube[9] video, promote people on Facebook[10], increase search rank of websites, etc.). The class *Other activity / other* covers any applications that are not covered by the previous categories or classes and for which no separate class was defined because there had only been a small number of instances.

The analysis indicates, that creating, revising, researching, rating and sensing information are the key categories of applications being performed on cloud labor platforms. This is actually confirmed by the way the platform providers are advertising their services. For example, MTurk focuses on the four categories "clean your data", "categorize items", "get feedback" and "create or moderate content" (Amazon Inc. 2013a).

Interestingly, a number of companies has emerged whose business model would not work without having access to the scalable workforce of cloud labor platforms. One of them is the transcription service CastingWords[11], which will be discussed in detail in Sect. 3.3.5.

Applications Discussed in the Scientific Literature

Naturally, cloud labor services are primarily used for tasks that cannot be automated. These tasks represent problems that cannot be satisfactorily solved by *artificial intelligence* (AI), "the science and engineering of making intelligent machines" McCarthy (2007). The vast majority of applications discussed in the scientific literature can be

[9] http://www.youtube.com/, last accessed on July 1, 2013.

[10] http://www.facebook.com/, last accessed on July 1, 2013.

[11] http://castingwords.com/, last accessed on July 1, 2013.

Table 2.3 Classification of cloud labor service applications by the activity to be performed by the worker.

Activity	Type of application
Create content	Create texts
	Create media
	Creativity or idea generation
Revise content	Check or classify content
	Tag content
	Audio or video transcription
	Summarize content
	Refine content
Research information	Data or information research
	Media research
Rate or sense information	User opinion
	User knowledge
Other activity	Language services or translation
	Programming
	Software or Web application testing
	Traffic augmentation
	Location–specific services
	Other

mapped to such problems. Table 2.4 clusters a selection of research papers according to areas of artificial intelligence. In addition to extending the reach of AI, cloud labor services are also considered an important tool for conducting behavioral research. According to Mason and Suri (2011), surveys performed on task platforms have been observed to deliver results comparable to online surveys. However, they provide access to a more diverse population than traditional subject pools do. Also the wages tend to be lower.

2.4.2 Identification of Relevant Tasks

Although the classifications of existing usage scenarios provided in the previous section can be used as a starting point when searching for additional use cases, they do not necessarily provide a complete picture. Especially within organizations there might be a large number of promising use cases which are specific to certain industries or methodologies.

From the concept of cloud labor services described in Sect. 2.1.2, a number of basic requirements can be deduced that help identifying relevant applications. First, the task must be issued (but not necessarily performed) through an electronic interface. Second, the entire task invocation cycle is automated. In particular, there is no personal interaction required with the requester. Additional considerations can be made from an economic perspective: Because of the one time effort for setting

Table 2.4 Examples of cloud labor service applications clustered by typical problems of artificial intelligence.

Area	Application
Natural language processing	Language translation, evaluation of machine translation quality (Callison-Burch 2009), Word sense disambiguation (Parent and Eskenazi 2010), Textual entailment (Negri and Mehdad 2010), Text creation based on abstract, modification of tense (Little et al. 2009a), Creation of question-answer sentence pairs (Kaisser and Lowe 2008), Ranking of computer generated questions about provided texts (Heilman and Smith 2010), Rating Wikipedia articles (Kittur et al. 2008)
Relevance assessment	Relevance of search results (Grady and Lease 2010), Relevance evaluation (Alonso et al. 2008)
Sentiment analysis	Classifying sentiment in political blog snippets (Hsueh et al. 2009), Obtain polarity scores for customer comments (Mellebeek et al. 2010), Capturing the amount of action indicated by a sentence (Madnani et al. 2010), Web site reviews and marketing surveys (Barr and Cabrera 2006)
Visual perception	Validate hierarchy of images generated by an algorithm (Deng et al. 2009), Describe a given image in one sentence (Rashtchian et al. 2010), Geometric reasoning experiment: Fold protein structures in a three-dimensional space (Corney et al. 2010), Identify and locate objects in images (Yang et al. 2008), Validate identity of a person (Gentry 2009), Medical image segmentation (Raykar et al. 2009)
Audio processing	Transcription of voice recordings (Marge et al. 2010; Novotney and Callison-Burch 2010a; Liem et al. 2011), Elicit narrations of Wikipedia articles as audio recordings (Novotney and Callison-Burch 2010b), Various applications (Parent and Eskenazi 2011)
Knowledge provisioning	Ontologies creation and matching (Eckert et al. 2010), Evaluate common sense knowledge from news or Wikipedia (Gordon et al. 2010)

up the task and developing the worker interfaces, there should be a large number of equivalent tasks. In order to profit the most from the cloud labor service (compared to using dedicated employees), this number should ideally vary over time. And finally, cloud labor services obviously primarily make sense if automating the task is not an option.

Altogether as a rule of thumb, cloud labor should be considered if the task

1. can be mapped to an electronic interface (Web, mobile device, etc.).
2. does not require personal interaction with the requester.
3. is subject to a large and ideally varying demand.
4. can either not be automated at all, or not well enough or automating it would be too expensive or too time consuming.

Often, a task can be automated, but the results are actually not good enough. In such situations, cloud labor can be used to accomplish the delta. For example, the recognition of handwritten texts still cannot be perfectly automated[12]. If a high level of quality is needed, such cases that are difficult to decide can be passed to a cloud labor platform in order to increase the overall result quality.

Cloud labor services can also help to reduce the time to market because they can often be set up in a few hours or even faster while automating the task may take much longer. So, even if automation is possible, cloud labor may sometimes be the preferred choice. This especially applies to the ramp-up phase of new services. Once the service is up and running, the focus may be shifted towards automating the service in order to reduce the overall cost. For temporary efforts, it may not be worth considering automation at all, because it may be more expensive than to apply cloud labor services.

2.4.3 Workflows and Task Granularity

There are many situations in which a sequence of multiple cloud labor tasks needs to be performed or a cloud labor task needs to be combined with automated tasks. Such sequences are called workflows. Workflows may also be used to reduce the task granularity by decomposing complex tasks into smaller, manageable units. Another specific objective of workflows can be to improve the quality of work results by combining activities of multiple workers, e.g. a response generated by a first user is validated by a second one.

There are situations in which it may be useful to pursue both objectives in parallel in order to optimize the overall system. For example, by having one worker compose responses from other workers into a final result, that worker could implicitly validate the responses of the other workers. As the focus of this book is quality management, workflows are discussed in two separate chapters: The overall concepts of workflows and their use for reducing the granularity is discussed here, while specific workflows for quality management will be discussed in Sect. 3.3. For example, *TurKontrol* (Dai et al. 2010), which is often mentioned in the context of human computation workflows, is actually rather a quality management concept and is therefore covered later.

Table 2.5 provides a list of the workflow concepts that have been proposed in the context of cloud labor services in the last couple of years.

TurKit (Little et al. 2009b, 2010b) is implemented as a Java program that can execute Javascript files and provides an API to them through which they can communicate with the MTurk platform in order to use "MTurk workers as subroutines". Additional functions are available for writing information into a database and for writing trace messages that can be used for tracking the execution flow. A recent crash-and-rerun feature ensures that long running processes can be seamlessly continued after a crash. TurKit can be accessed using a public Web GUI by entering the MTurk credentials. TurKit has been used as a basis for a series of other research contributions. One example is *Soylent* (Bernstein et al. 2010), "a word processor with

[12] Refer to Sect. 8.1.

Table 2.5 Workflow models and tools for cloud labor services.

Name	Key features	Source
TurKit	Supports workflows created in JavaScript and provides a tracing and a "crash-and-rerun" feature.	(Little et al. 2009b)
CrowdLang	Supports predefined workflows as well as dynamic creation of workflows at runtime.	(Minder and Bernstein 2011)
CrowdWeaver	Allows for visually creating and managing workflows with real time monitoring and notification.	(Kittur et al. 2012)
Crowdforge	Implements MapReduce framework.	(Kittur et al. 2011)
Turkomatic	Allows for recursively decomposing work into smaller portions.	(Kulkarni et al. 2011)
Clowder	Uses decision theory for deciding on task flow, customizing interfaces and controlling quality.	(Weld et al. 2011)

a crowd inside", that allows for crowdsourcing text editing tasks to MTurk workers out of a Microsoft Word™ document.

Rather than extending a state of the art programming language, *Crowdlang* (Minder and Bernstein 2011) represents a proprietary executable and model-based programming language and framework that is specifically designed for creating workflows of human tasks. Workflows can either be predefined or created by workers at runtime. The tool comprises ready to use operators for managing task granularity, the actual control flow and the aggregation of responses.

CrowdWeaver (Kittur et al. 2012) does not use a written programming language but is designed as a visual system. Workflows of human tasks and machine tasks are represented as a visual model and are created using a visual interface. At runtime, the workflow graph is being updated in real-time with status information like the number of assignments being processed of each task, the quality and the costs. Templates allow for reuse of workflows. A notification feature keeps the user informed about important events. The tool is implemented by a JQuery-based Web visual and a Ruby controller and internally uses a MySQL database. In contrast to the tools discussed previously, CrowdWeaver is not based on MTurk but on the CrowdFlower[13] platform. That way, it has access to a wider range of Cloud labor platforms, including MTurk. CrowdFlower functionality is used to create and modify the underlying human tasks.

CrowdForge is a concept and prototypical system for managing task granularity. The basic idea is to crowdsource not only the actual task execution but also the breakdown of complex tasks into manageable units and the recombination of the individual results. Inspired by Google's MapReduce algorithm, the entire process consists of three steps: A "partitioning task", a "map task" and a "reduce task" are each being defined as a separate human sub-task. A first set of workers recursively breaks down

[13] The CrowdFlower platform will be described in Sect. 2.5.1.

the work into smaller units, which are then executed by other workers. Again other users are finally recombining the work results on several stages until the overall result has been generated. In addition to task de- and result re-composition, CrowdForge (Kittur et al. 2011) also supports a variety of quality management mechanisms. The tool is implemented as a web-based prototype.

Turkomatic (Kulkarni et al. 2011, 2012) suggests a recursive approach similar to the one used by CrowdForge. A collaborative approach is proposed for managing the quality of the decomposition and recomposition tasks.

Weld et al. (2011) argue that "artificial intelligence methods can greatly simplify the process of creating and managing complex crowdsourced workflows". Their *Clowder* framework represents a vision of an all purpose cloud labor system that uses decision theory for deciding on alternative workflows, for personalizing the appearance of interfaces and for controlling the overall workflow. They expect that "optimized workflows are significantly more economical (and return higher quality output) than those generated by humans". One of the features they suggest is the personalization of interfaces based on implicit "A-B" testing performed by workers. They also recommend the use of a hierarchical task-network (HTN) for modelling task decomposition.

2.4.4 Privacy, Copyright and Compliance

The cloud labor concept is based on the exchange of information. Information is being passed to the workers through the cloud labor services platform as input for the task. Based on this information, the worker processes the task, potentially by utilizing external information from the Web or, especially in case of surveys, by using information about personal preferences, desires etc. While working on the task, the worker may generate new intellectual property. For example, a text authored by the worker will be subject to copyright. There may be four major types of critical information involved in the cloud labor service scenario, which are illustrated by Fig. 2.4:

1. Sensitive information of the requester's organization. This could be business secrets or any information that is subject to data protection acts or other rules and regulations. An example is sensitive information about customers, employees or business partners.
2. Personal sensitive information of the workers, e.g. details about their financial situation or their personal desires.
3. Intellectual property generated by the workers either during task execution or earlier.
4. Copyrighted information used by the workers during task execution, for example a text or picture being retrieved from a website.

The potential exposure of sensitive information or use of protected intellectual property causes a series of risks that need to be addressed accordingly: The most

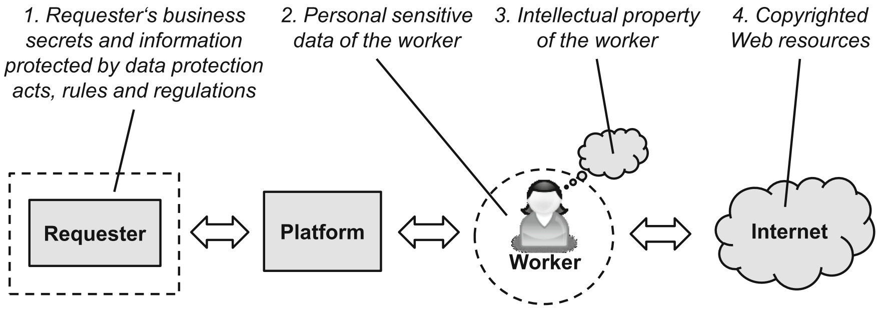

Fig. 2.4 Critical information potentially being involved in the cloud labor service scenario.

obvious risk is probably the disclosure of sensitive information by the requester. A common mitigation strategy is the anonymization of the data by removing or replacing all irrelevant information like names and phone numbers, or by splitting tasks into small units. For example, the transcription service CastingWords[14] breaks speech recordings down into small pieces before they are transcribed into text. Afterwards, the pieces are recomposed into a complete transcript. As most workers see just small portions of the transcript, the risk of privacy issues is reduced[15]. However, anonymization may not fully avoid the risk of disclosing sensitive information. As Felstiner (2011) points out, a "crowd worker may still be able to glean knowledge of a valuable piece of intellectual property by completing even a small task". This risk increases with the number of closely related tasks a worker performs.

If the information cannot be anonymized, a private or hybrid people cloud can be used. The most conservative approach would be to use a workforce from the requester's organization. For less critical privacy issues it might be sufficient to oblige service workers to maintain confidentiality about the processed information by adding an appropriate clause to the terms and conditions of the task or platform.

From the worker perspective, there is a risk of potential misuse of personal data. Even though the workers' identity is usually not disclosed to the requester, a unique ID is being assigned to each worker which is attached to any responses returned by that worker. Over time, the requester may gather a large amount of information about the worker based on that ID. As long as the worker does not explicitly trust the requester, a basic mitigation strategy is not to disclose any personal information. Platform providers should not disclose the identity of the workers.

If the task involves the creation of intellectual property by the worker, the requester and / or the platform provider should ask the workers to grant their IP rights to the requester. In this context, an important risk is the misuse of copyrighted information, e.g. due to plagiarism. This applies specifically to content generation applications. Instead of creating new content, workers may search for appropriate content in the Web and represent it as their own original work. If the requester then uses it publicly, he might get sued by the actual originator. Even though penalties could be enforced

[14] CastingWords will be discussed in detail in Sect. 3.3.5.

[15] http://castingwords.com/support/transcription-faq.html, last accessed on March 11, 2012.

against the worker for such an offense, this would likely not remedy the damage, at least not in public people cloud scenarios in which there is only a lose contract with the worker. Therefore, the requester should check for plagiarism by using services like Plagiarism-Detect[16] or Copyscape[17] or by asking other workers to manually perform a check.

In information research applications, licensing issues are another possible threat. For example in an image research application a requester might ask the workers to find pictures of products that he wants to use for an online product catalog. The requester needs to ensure that the licensing rules of those images allow for commercial use. Therefore, he should always ask the worker to return the link for the image instead of the image itself. This way, the licensing rules can be validated, for example by asking other workers to do so.

2.5 Platform Perspective

This section describes the state of the art of cloud labor services from the perspective of the coordination platform. After providing an overview of the existing commercial cloud labor platforms in Sect. 2.5.1, Sect. 2.5.2 introduces considerations about their service models and technical infrastructure. Section 2.5.3 then addresses the matching of workers to tasks and tasks to workers while Sect. 2.5.4 finally refers to quality management for cloud labor services.

2.5.1 Existing Platforms

A large number of cloud labor platforms have emerged in the last few years. The industry website Crowdsourcing.org (2012b) lists 164 entries in the category of cloud labor[18]. Compared to that, Frei (2009) had listed only 46 sites in 2009. As most of the platforms do not disclose any information about their task throughput, it is difficult to estimate the actual relevance and ranking of the sites. In order to identify the key players, Table 2.6 contains those of them that have been sponsors or exhibitors on the 2010 CrowdConf (Crowdflower 2010), which can be considered the primary commercial crowdsourcing event of the year. As the only exception, the table also covers the MTurk platform which is used as the workforce provider for several of the other platforms. The table only captures platforms for programmatic cloud labor and does not cover platforms for manually managed cloud labor like ODesk. The last two columns indicate the founding year and the number of workers available to

[16] http://plagiarism-detect.com/, last accessed on July 1, 2013.

[17] http://www.copyscape.com/, last accessed on July 1, 2013.

[18] The directory does not just comprise crowd labor platforms but also other sites that build there business model on the concept of cloud labor.

Table 2.6 Selection of commercial cloud labor service platforms.

Platform	Focus	Country	Year	Workers
Amazon Mechanical Turk	All purpose platform with self service only and limited quality control; used as a workforce provider by many other platforms	USA	2005	500 k (own)
Clickworker	Processing of unstructured data, such as text, photographs, and videos	Germany	2005	300 k (own)
CrowdFlower	All purpose platform with specific focus on quality management	USA	2007	3 M (several)
Microtask	Document processing and data entry	Finland	2009	unknown
Microworkers	Traffic augmentation	USA	2009	unknown
CloudFactory	All purpose platform with focus on ongoing processes as well as one-time tasks	Nepal	2011	500 k (Mturk)
Crowdsource	All purpose platform	USA	2010	500 k (Mturk)
Serv.io	Content creation, content management, translation services	USA	2009	150 k (own)
UTest	Application testing (Web, mobile and desktop)	USA	2007	70 k (own)

the platform followed by the type of workforce. "Own" means that the workforce is managed by the platform itself, "MTurk" means that the workforce of MTurk is used. "Several" indicates that the platform relies on the workforces of several other platforms.

One of the most popular cloud labor platforms is MTurk which has been started in late 2005. On MTurk, service requesters can publish open calls for *human intelligence tasks* (HITs). Any Internet user that meets certain skill criteria may act as a service worker and work on tasks to earn some money. Amazon keeps a fee of 10 percent of the task price with a minimum of $0.005 per task[19]. MTurk provides a basic Web interface for interactively defining tasks and designing the corresponding user interfaces. In addition, a command line API as well as SOAP and REST based APIs support several programming languages including Java, Ruby, Perl and .NET[20]. While workers from the US and from India can transfer their earnings to a bank account, workers from all other countries can claim their earnings only in form of Amazon gift certificates (Amazon Inc. 2013b). According to Ipeirotis (2010a),

[19] See FAQ section on the help page of (Amazon Inc. 2013a).

[20] See developer section of the MTurk requester website (Amazon Inc. 2013a).

MTurk is a "heavy-tailed market" with regard to the requester activity. A minority of one percent of the requesters account for half of the overall dollar amount being paid to the workers. Furthermore, there is a focus on small tasks. For 90 % of the HITs a reward of $0.10 or less is being paid.

Another early player is the German platform *Clickworker* who first went live in 2006 under the name *Humangrid*, which is still the name of the company that operates the platform. Clickworker has a workforce of about 300.000 own workers and mainly focuses on processing unstructured information, such as text, photographs, and videos (Humangrid GmbH 2013).

The *Crowdflower* platform founded in 2007 positions itself as the market leader among the platforms for cloud labor services. It is a Meta platform that leverages the workforce of several other platforms and external workforce providers including MTurk. According to an e-mail of the founder Lukas Biewald, they process about one million tasks on an average day, corresponding to an effort of 4 person years (Biewald 2012).

Microtask from Finland primarily concentrates on document processing and data entry with a specific focus on digitizing handwritten forms. In cooperation with the National Library of Finland, volunteer workers are leveraged to correct OCR errors in a historical newspaper archive (Microtask Oy n.d.).

The *Microworkers* platform launched in mid 2009 mainly focuses on traffic augmentation applications like search engine optimization. Compared to MTurk, two additional important differences concern the way payments are being made to the workers as well as the structure in which the tasks are being organized (Hirth et al. 2011). While MTurk requires a US bank account for monetary transactions, Microworkers supports online payment services like PayPal[21], which are better suited for international workers. Rather than putting all HITs into a single list, Microworkers provides predefined job categories with individual minimum payments starting from $0.10.

CloudFactory was founded in 2011 in Nepal in order to develop job opportunities for the developing countries. The platform *Crowdsource* was acquired by Scalable-Workforce LLC in 2011 and merged with their own platform into crowdsource.com (CrowdSource n.d.). *Serv.io* claims to be the market leader for e-commerce content services[22]. Through their ClowdCrowd platform they have access to over 250 k workers[23].

UTest founded in 2007 is not a typical task platform but an application testing service for Web, mobile and desktop applications that uses crowdsourcing in order to distribute testing work to a pool of some 70 k testing professionals around the world[24]. Because of the formalization of those tasks and the limited need for personal interaction with the requester, these tasks are also considered to match the definition of cloud labor services.

21 https://www.paypal.com/, last accessed on July 1, 2013.

22 http://www.serv.io/, last accessed on April 2, 2013.

23 http://www.cloudcrowd.com/company/about, last accessed on April 2, 2013.

24 http://www.utest.com/about, last accessed on April 2, 2013.

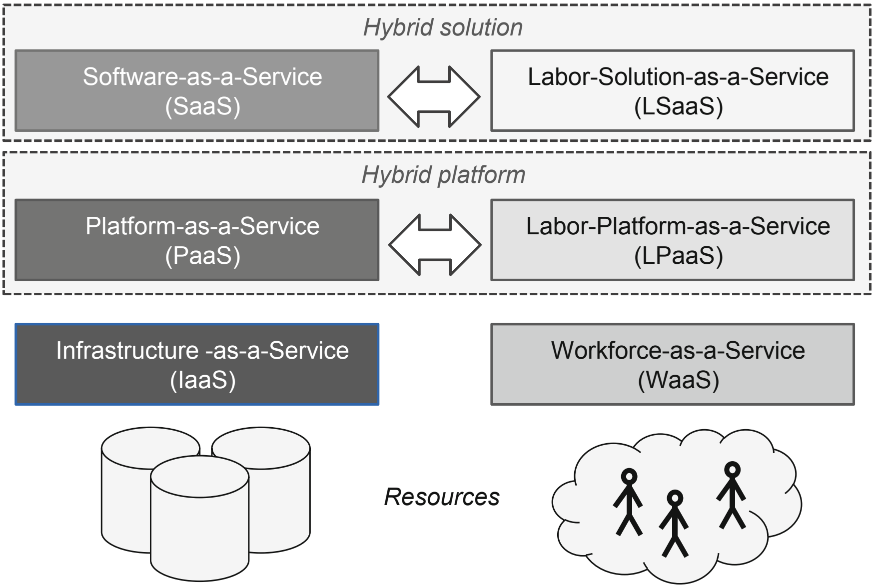

Fig. 2.5 Concept of a cloud labor service (cloud labor) stack in comparison to the cloud computing stack.

2.5.2 *Technical Infrastructure*

Because of the analogies between cloud computing and cloud labor it can be assumed that similar service models can be applied to both concepts. This section aims to provide a sketch of a cloud labor stack by deriving a set of service models from those of cloud computing, which are defined by Mell and Grance (2011) as:

- *Infrastructure as a Service* (IaaS) provides the actual IT infrastructure including computing power, storage and networks on which the user can deploy operating systems or other arbitrary software. The user only manages the operating system or other software but has no control over the underlying infrastructure.
- *Platform as a Service* (PaaS) allows for deploying own applications or third party applications on top of the cloud infrastructure by leveraging programming models supported by the provider. The user only controls the "application-hosting environment" but has no control over the underlying operating system or hardware.
- *Software as a Service* (SaaS) provides remote access to specific applications that are running on a cloud infrastructure. The user only controls "user-specific application configuration settings" but has no control over the underlying hosting environment, operating system or hardware.

Inspired by the definitions of Mell and Grance (2011), a similar stack could be described for cloud labor which is illustrated by Fig. 2.5:

- *Workforce as a Service* (WaaS) delivers the workforce of a crowd of workers to a requester as a scalable resource. The requester can define arbitrary tasks which are completed by the workers remotely, typically through a Web application or mobile application, and for which responses are delivered by the workers back to the requester. The requester does not manage or control the availability of the individual workers but can decide, which workers are allowed to work on specific types of tasks.
- *Labor Platform as a Service* (LPaaS) provides the ability for requesters to deploy cloud labor solutions that have been created by using workflow languages and task definition capabilities provided by the platform. The consumer does not manage the assignment of workers tasks but has control over the quality of work results and the skill profiles of the workers by utilizing facilities provided by the platform.
- Labor Solution as a Service (LSaaS) provides access to the cloud labor solutions running on a labor platform. The cloud labor solutions are accessible from various client devices such as a Web browser or API. The requester does neither manage or control availability, assignment or the skill of the workers, nor defines the quality of the work results, with the possible exception of solution specific configuration settings.

Of course, there are many applications that require both cloud labor and computing resources. Therefore, there should not be a sharp borderline between SaaS and LSaaS. Instead, LSaaS should be merged with SaaS into a hybrid layer in order to be able to effectively combine the capabilities of both worlds. The resulting hybrid solutions may again be running on a hybrid platform which itself is a merger of PaaS and LPaaS. This layer would have access to both IT infrastructure (IaaS) and human workforce (WaaS).

The above considerations have been developed based on a discussion with Stephen Dill from the IBM Almaden Research Center in California, USA in March 2012. A similar stack was proposed independently by Panos Ipeirotis in July 2012 (Ipeirotis 2012) which mainly differs in two aspects: First, from a terminology perspective, the term *Labor Infrastructure as a service* (LIaaS) is used instead of WaaS. In this book, WaaS is preferred in order to emphasize the fact that the actual service being provided to the requester is the workforce of the crowd. Second, LSaaS is defined by Labor Applications/Software as a service and represents basically what is described as hybrid solution above. In order to clearly distinguish from and illustrate the symmetry with the cloud labor stack, LSaaS is defined independently of the cloud labor stack here.

The existing commercial cloud labor services platforms follow different service models. MTurk can mainly be seen on the level of WaaS, but also cooperates with partners in order to deliver hybrid solutions, e.g. for media tracking. MTurk actually names such solutions “Apps”.[25] Crowdflower could be seen as LPaas because it leverages a multitude of WaaS providers including MTurk. Crowdflower also offers a series of LSaaS or hybrid solutions, e.g. for content moderation, lead

[25] See apps and case studies on (Amazon Inc. 2013a).

data enhancement and sentiment analysis.[26] However, as Turian (2012) points out, Crowdflower does not provide native support for workflows. He sees MobileWorks close to the idea of LPaaS but mentions that they offer only immature self-service capabilities. On the LSaaS level, he sees CastingWords, UTest, Microtask and others, but he underlines that due to the lack of a robust LPaaS provider, all labor solutions "must be vertically integrated with the platform layer or implement their own application-specific platform".

These observations indicate that there is indeed an overlap between the concepts of cloud computing and cloud labor. However, the option for deploying own solutions on cloud labor platforms in the sense of PaaS or hybrid platforms is still limited to explicit cooperations with the platform providers. The functionality is not yet available as a self service according to the original idea of cloud computing.

From an architectural and implementation perspective, the considerations about BPM and SOA mentioned in Sect. 2.2.7 are highly relevant for cloud labor services as they provide fundamental standards and tools for orchestrating human tasks.

2.5.3 Worker-to-Task Matching and Allocation

In the context of cloud labor platforms, *matching* is the process to identify suitable tasks to be performed by a worker or to identify suitable workers for a task. *Allocation* is the actual assignment of a task to a worker. The following paragraphs describe how the two concepts are being addressed.

Allocation

According to the definition of crowdsourcing, the allocation is always performed as an open call, i.e. the worker browses tasks and decides what tasks to work on. In practice, this open call is often not fully open, but the pool of workers who are eligible to work on a task is restricted. This can be due to language, location or skill requirements or because a so called *certification test* needs to be successfully completed before being allowed to work on a task. It can be argued, "how open" the call needs to be in order to satisfy the definition of crowdsourcing. In any case, a directive assignment of a task to a worker would not be considered crowdsourcing.

The actual assignment of a task could also be the result of an auction. Satzger et al. (2011a) describe a concept in which a closed-bid auction mechanism assigns tasks to one of multiple workers who have bid for it. Even though the actual assignment is being performed by the platform, there is still an open call which in this case is a call for bids.

Apart from the possible use of auction mechanisms, the open call assumption leads to the fact that the allocation of workers is rather straightforward. The actual

[26] See solution overview on http://crowdflower.com/products, last accessed on February 6, 2013.

challenge is the matching step, which comes down to presenting the right options to the worker to choose from.

Matching

There are three factors that constitute the list of tasks that is presented to the worker:

1. A search term and a sort order specified by the worker.
2. A qualification requirement or other restriction that prevents the worker from working on the task.
3. A recommender system that proposes suitable tasks to workers.

On current task platforms like MTurk, only options 1 and 2 are implemented. Workers typically browse for the most recently posted tasks or the ones that have the largest number of available assignments and limit their scope to the first pages of the task list (Chilton et al. 2010). For the requesters, this behavior limits the chance for getting access to workers with a specific expertise. It also reduces the average result quality as many workers are performing tasks for which they do not have the skills (Ambati et al. 2011).

For the workers, there can be an advantage to focus on tasks with qualification requirements as this will likely reduce the competition of workers. DiPalantino et al. (2011) have observed that the revenue tends to be smaller if there are many competitors. Better matching mechanisms also have the potential to reduce the costs and the resolution time. Chilton et al. (2010) have observed that tasks which are easy to find for a worker are "completed 30 times faster and for less money" than others. Therefore, recommending suitable tasks to workers (option 3 mentioned above) can be assumed to be an important ingredient for cloud labor platforms. However, recommender systems have only rarely been described in the context of crowdsourcing and specifically cloud labor.

For Wikipedia, Cosley et al. (2007) have developed a *Suggestbot* that proposes articles to be revised by editors. In a preprocessing step, the software creates a local mirror of Wikipedia that contains the texts of all articles and identifies all editing needs based on tags provided within the articles. At the same time, it models the interests of the Wikipedia editors by identifying all editing activities they have previously performed and all the links they have created between articles. By analyzing the text similarity to other articles that need work, recommendations are made to the editors.

For cloud labor platforms, only a small number of papers have proposed initial ideas about the use of recommender systems: Ambati et al. (2011) have sketched a recommendation engine for suggesting tasks to users based on implicit modeling of skills and interests. The system creates a preference model of the worker using a bag-of-words approach. The model is based on three types of information:

- A profile created by the worker covering information about his location, education, skills and experience,
- explicit feedback about the worker's rating of the tasks and the payment, and
- implicit feedback using search terms, tasks being selected, task description, reward, number of assignments and the requester's feedback in form of rejection, bonuses or comments.

Yuen et al. (2011) have proposed a system that assumes a platform with well defined task categories. By keeping performance data per category like selection preference, task acceptance rate and reward and time allotted, the system aims to rank the available tasks according to the order of best matching.

Matching of workers may also be influenced by their locations. For example, Reddy et al. (2010) have described a "recruitment framework for participatory sensing data collection" that utilizes the ability of modern mobile phones to identify their location and to be able to perform in-situ data collection by taking pictures, audio or video recordings.

2.5.4 Quality Management

As mentioned before, quality management has received most attention of all research foci in the context of cloud labor services (Kittur et al. 2013) and is regarded as the by far most pressing challenge in this space (Crowdsourcing LLC 2012; Turian 2012). Because of its importance and because it represents the focus of this book, a separate chapter is devoted to this topic. Chapter 3 will solely concentrate on quality management considerations for cloud labor services.

2.6 Workforce Perspective

The workforce perspective covers the aspects and concepts that are mainly influenced by the workers. After illustrating the demographics of the workforce in Sect. 2.6.1, Sect. 2.6.2 describes the primary motivators for people to perform work on cloud labor platforms. Sections 2.6.3 and 2.6.4 then address the importance of worker education and feedback mechanisms as well as considerations about the task design. Finally, Sect. 2.6.5 discusses to what extent cloud labor can be considered a desirable work model.

2.6.1 Worker Demographics

By far most studies on worker demographics have been performed on the MTurk platform, which is therefore used as an example here. Over the years, the demographics have shifted towards a more international population on that platform. In late 2008, the population had still been US centric. Compared to 83 % US workers there had been only a minority of 5 % of workers from India and 12 % from other countries (Ipeirotis 2008). At the end of 2009, the portion of Indian workers had already increased to 36 %, while there had been 56 % workers from the US and 8 % from other countries; at the same time, "the number of lower-income workers has increased, along with the number of young workers, male workers, and Indian workers" (Ross et al. 2010).

According to Ipeirotis (2010b), the demographics differ considerably between the US and India. In the US, the average age of the workers is some 34 years compared to 27.5 years in India. While 35 % of the US workers have at least a bachelor degree, that is the case for 52 % of the Indian workers. In the US, 75 % of the workers have an annual household income of at least $25,000 and only 13 % have less than $10,000, whereas the picture is almost the opposite in India: Only 16 % of the workers have an annual household income of more than $25,000 and more than 55 % declared to have less than $10,000. In both the US and India, the majority of workers are singles and do not have children. Most of the workers spend only a day or less per week on cloud labor and earn often less than $20 during that time. A portion of 12 % of the US and 24 % of the Indian workers state that Mturk is their primary source of income, whereas 59 % of the US workers and 70 % of the Indian workers consider MTurk as a "fruitful way to spend free time and get some cash (e.g., instead of watching TV)" (Ipeirotis 2010b).

One reason why most of the workers are from the US and from India may be that on MTurk, cash payment is only supported in US dollars and Indian Rupees (Mason and Suri 2011). And indeed, on the Microworkers platform, which allows for international cash payments through payment services like Paypal, there are much more workers from other countries. According to an analysis performed by Hirth et al. (2011) based on anonymized data received from the platform provider, most workers are from Indonesia (18 %), Bangladesh (17 %), India (14 %), United States (11 %).

2.6.2 Motivation and Incentives

The factors motivating people to perform work on cloud labor platforms can be divided into two groups, intrinsic and extrinsic motivation (Kaufmann et al. 2011; Rogstadius et al. 2011).[27] Kaufmann et al. (2011) state that "intrinsic motivation exists if an individual is activated because of its seeking for the fulfillment generated by the activity (e.g. acting just for fun)" while "in the case of extrinsic motivation the activity is just an instrument for achieving a certain desired outcome (e.g. acting for money or to avoid sanctions)".

According to Kaufmann et al.'s (2011) model for worker motivation in crowdsourcing presented in Table 2.7, there is a set of motivating factors that can be

[27] This distinction was originally proposed in the self-determination theory by Deci and Ryan (1985).

Table 2.7 Model for worker motivation in crowdsourcing with empiric scores for motivational constructs; derived from (Kaufmann et al. 2011).

Type	Category	Construct	Score
Intrinsic	Enjoyment based	Skill variety	2.4
		Task identity	2.3
		Task autonomy	2.4
		Direct feedback from the job	2.0
		Pastime	2.1
	Community based	Community identification	2.0
		Social contact	1.3
Extrinsic	Immediate payoffs	Payment	3.0
	Delayed payoffs	Signaling	1.9
		Human capital advancement	2.2
	Social motivation	Action significance by external values	1.7
		Action significance by external obligations & norms	1.0
		Indirect feedback from the job	1.7

classified to be extrinsic or intrinsic. Each of the categories may be measured by a number of motivational constructs. The score value provided by the last column was determined by a survey performed on the MTurk platform. A higher score means that the construct had been considered more relevant by the workers. The minimum possible value is −0.78, the maximum is 3.99.

With a score of 3.0, the extrinsic construct *payment* sticks out from all other constructs. That reveals that payment is the primary motivation for people to do work on MTurk. But also task related factors seem to be important. Those factors include the variety of skills needed for performing the task, the autonomy to decide how to perform the work, the direct feedback that workers may get by validating the quality of their work themselves[28], and also the opportunity to train their skills or gather new skills by working on a task (*human capital advancement*).

Interestingly, even killing time (*pastime*) can be a reasonable motivation for using the platform, a phenomenon already identified by Brabham (2008b) and by Ipeirotis (2010b). In a survey conducted by Ipeirotis (2010b), 32 % of the US workers and 5 % of the Indian workers have reported that they participate on MTurk to kill time. However, Kaufmann et al. (2011) have determined that those workers are not using the MTurk platform frequently.

The authors also point out the "high potential for community induced intrinsic motivation" and in fact, even though the MTurk platform does not provide any capabilities for community building, a number of external communities like Turkernation[29] or the MTurk forum[30] and tools like Turkopticon (Silberman et al. 2010)

[28] For example, in a programming task, direct feedback could be received if the workers test the code that they have developed.

[29] http://turkernation.com, last accessed on July 1, 2013.

[30] http://mturkforum.com, last accessed on July 1, 2013.

have emerged that provide a forum for workers to discuss experiences and comment on requesters.

Social motivation like altruism (*action significance by external values*) or seeking for commendation (*indirect feedback from the job*) seem to be less relevant for the workers. This may be a specific characteristic of tasks platforms on which the workers are usually acting anonymously. External obligations usually do not exist on Mturk because the workers are freely choosing the tasks they are working on rather than being forced by an employer to work on specific tasks. That can be assumed to be different in an enterprise crowdsourcing scenario.

2.6.3 Education and Feedback

Education and feedback obviously play an important role in the cloud labor service scenario because the availability of adequately skilled workers is the primary requirement for generating high quality results. For certain tasks types, a specific training may be required in order to prepare the worker for successfully completing it. Such a training is usually integrated into a qualification test that the worker has to complete before being allowed to work on the tasks. On most platforms workers can contact the requester via e-mail if they have questions regarding the task. Ambiguous task instructions may also be discussed in worker communities like TurkNation. However, in order to prevent fraud, only the instructions themselves and not the responses to individual tasks may be discussed.[31]

Providing regular feedback to the workers can be seen as another important ingredient for continuously improving their skills. Hattie and Timperley (2007) differentiate between four general types of feedback, two of which can be assumed to be relevant for cloud labor services: *Feedback about the task* and *feedback about the processing of the task*.

The most common form of feedback is feedback about the task which "includes feedback about how well a task is being accomplished or performed, such as distinguishing correct from incorrect answers" (Hattie and Timperley 2007). A major drawback is that feedback about the task often cannot be easily generalized to other tasks and, therefore, does not encourage the receiver to develop strategies to attain a goal. Feedback about the processing of the task addresses this limitation by enabling the receiver to detect potential errors independently and to adjust the strategy accordingly, possibly by seeking help.

Platforms like MTurk support feedback about the task by allowing the requester to accept or reject responses along with a comment about the rationale for the decision[32]. Feedback about the processing of the task is partially supported by some quality management mechanisms. For example, Le et al. (2010) are using training tasks

[31] http://turkernation.com/faq.php?faq=vb3_board_faq, last accessed on February 17, 2013.

[32] http://aws.amazon.com/documentation/mturk/, last accessed on February 17, 2013.

with predefined explanations of potential errors. This will be explained in more detail in Sect. 3.3.2.

Not just the type of the feedback matters but also its timing. In a study with high school students, immediate feedback has been observed to be more effective for simple tasks while delayed feedback is still effective for complex tasks (Clariana et al. 2000). This suggests that difficult tasks require more feedback regarding the processing of the task, which can be better provided in form of delayed feedback.

2.6.4 Task Design

This section summarizes a number of observations that may be used as a guideline when designing cloud labor services and when deciding on the type and amount of the payment.

In an empirical study on the MTurk platform, Schulze et al. (2011) found out that in general, workers prefer clear instructions, genuine requests and a task design which does not require them to leave the Mturk site for completing the task: Everything they need should be integrated into the task's user interface. One reason the authors mention for that is that many workers have suffered from malicious websites.

The study further suggests that workers can be categorized into three groups which are outlined in the following:

- *Quick profit jobbers* prefer rather simple tasks with a short description that can be completed in a short time and result in a relatively high reward per hour.
- *Informed workers* prefer well described tasks that have been submitted from well-respected requesters who are willing to respond to questions if needed. The tasks should be underpinned with comprehensive examples for correct and incorrect responses so the workers will understand under what conditions their work will be rejected.
- *Challenge seekers* prefer challenging tasks that sound interesting and enjoyable and for which background information is provided. There should be multiple tasks of the same type available and bonuses should be payed for good performance.

Interestingly, the level of education does not seem to have an impact on the worker's preferences. Indian workers seem to be keen on bonuses being paid for good work while US workers and also full-time workers from any region appreciate to work for well-respected requesters (Schulze et al. 2011).

From these observations it can be deduced that the granularity of simple tasks should rather be low, which, as will be discussed in more detail in Sect. 3.1.2, can also have a positive impact on the result quality. For more complex tasks, background information and comprehensive examples should be provided, complemented by the ability to contact the requester. A good reputation of the requester matters for complex tasks and will for several reasons also have a positive impact on the execution performance: It attracts full-time workers and workers from different regions and, as further discussed in Sect. 2.6.5 also reduces the risk from being put on a black list by worker communities which would drastically slow down task execution. An important factor of the requester's reputation is the payment in terms of the average hourly rate determined by the workers. As described in Sect. 3.1.2, the payment also directly affects the overall execution performance but not the accuracy of the result. However, no payment at all may lead to a higher accuracy than a low payment.

In order to help requesters determining the actual payment, Horton and Chilton (2010) have proposed a model that predicts how workers supply work to a cloud labor platform depending on the available tasks and their expected hourly wage. The model is based on the assumption that workers would not start working on any task as long as the expected wage is lower than a certain minimum wage which the authors define as *reservation wage*. Although their tests have confirmed their model only to a certain extent, the authors are convinced that it still provides a useful approximation. A possible explanation they offer for the mismatch is that a significant number of workers may act as *target earners* who try to work towards a certain self-imposed target wage rather than responding to the actual market, i.e. "when wages are high, a target earner works less".

2.6.5 Work Model

From the perspective of the workers, the concept of cloud labor provides exceptional opportunities but also severe risks. The concept offers an unprecedented freedom to decide when, where, how much to work and what to work on. There is no need of commuting to an office during rush hour but at the same time, working remotely does not necessarily mean that people have to work in an isolated manner from home. Along with the success of cloud labor platforms, a new type of flexible office space is emerging that can be booked on an hourly basis. Liquidspace[33] connects "nomadic workers and property owners who have underutilized space" (Kolodny 2012). Office space may be booked and paid in seconds using a smartphone application. The booking confirmation displayed on the screen represents the admission ticket for the office. Interestingly, the service is not only used by crowd workers but also by large companies who want to offer flexible offices to their remote workers. Available space includes underutilized desks of large corporations, conference rooms in hotels and even free space in public libraries.

On the downside, employees are afraid that their jobs may be turned into freelance cloud labor engagements with no job security or benefits plan. Felstiner (2011) mentions that "many creative professionals and providers of skilled services (such as software development) fear that their industries will go the way of stock photography". In addition, he sees a risk of continuously low wages because "treating individual workers as sources of cheap labor has hardly proven to be a recipe for failure in the past. If some are willing to work for substandard wages and benefits without legal protection, by necessity or choice, there is no reason to believe that

[33] http://liquidspace.com, last accessed on February 16, 2013.

their willingness will evaporate in cyberspace". Felstiner points out that existing protective statutes[34] do not adequately cover the concept of cloud labor because "in the past, a worker could not physically perform a unit of piece-rate labor in under a second. Parties did not make employment contracts from the other side of the planet with the click of a mouse. Employees were unlikely to have twenty-five separate employers in the course of a single workday." Therefore, he believes that legal practice "must revise the definition of 'employer' and 'employee' to recognize the economic realities of online and virtual work".

IG Metall, which is the largest labor union in Germany[35], is concerned about how to represent the crowd workers in a global cloud labor scenario given that they are today organized by industry and location (IG Metall 2013). Specific challenges the labor union foresees include the definition of agreed wages along with standardized labor conditions and also the question, who would be the bargaining partners. Would it be the platform providers or rather large enterprise requester companies? IG Metall also sees a chance that self-organization of the communities may strengthen the position of the crowd workers with regard to the requesters.

And in fact, as mentioned in Sect. 2.6.2 with Turknation and Turkopticon, first shapes of such communities are already emerging. Therefore, Felstiner (2011) summarized that "in the end, crowdsourcing relies on the crowd for its very existence. Legal intervention can only buttress and protect the organized efforts of crowd workers; it cannot replace those efforts".

Chapter 2 has shown that cloud labor services are still an evolving field of research. It poses a variety of challenges that require additional research in a several disciplines. As just discussed, this concerns a critical reflection of the effect cloud labor services will have on the way we will perform work in the future. However, this also concerns their technical and conceptual feasibility. The remainder of this book contributes to the latter by focusing on quality management of cloud labor services.

[34] In his paper, he specifically focuses on the US and discusses to what extent the *Fair Labor Standards Act* (FLSA) and the *National Labor Relations Act* (NLRA) are applicable to crowdsourcing.

[35] http://www.dgb.de/uber-uns/dgb-heute/mitgliederzahlen/2010, last accessed on April 3, 2013.

http://www.springer.com/978-3-319-09775-6

Dynamic Quality Management for Cloud Labor Services
Methods and Applications for Gaining Reliable Work
Results with an On-Demand Workforce
Kern, R.
2014, XVI, 215 p. 49 illus., Softcover
ISBN: 978-3-319-09775-6