Semantic Web Services
Assignment 2

Stefan Haselwanter
Juliette Opdenplatz

April 10, 2018

1 SOA and SESA

Service-oriented architecture (SOA) groups different functionalities (services) around business processes which are built on low abstraction levels. Applications can access these services through a standardized communication protocol over a network in order to to exchange data with one another. SOA units can be combined (*orchestrated*) to built more complex and larger software applications on a higher abstraction level. Hence, SOA is less about modular programming of software applications, and more about how to compose an application by integrating distributed, separately-maintained and deployed software components. SOA aims to completely abstract away the underlying complexity of a service such that users can access these services without any knowledge of their internal implementation. Further, SOA provides loose coupling between services in order to independently use and reuse them.

Machine processable semantics need to be added to bring SOAs to their full potential and make them scalable by describing important service aspects, e.g., negotiation, discovery, adaptation, composition, invocation, monitoring, data and process mediation.

Formal contract Define functionality, interaction, data types, data models, policies, assertions.

Loose coupling Dependencies to components, resources, service consumers, etc.

Abstraction Hide as much of the underlying details (technology, functionality, logic, QoS) as possible.

Reusability Independent logic that is divided into various services and can be reused for future purposes.

Autonomy Control the encapsulated functionality at runtime and design-time.

Statelessness Do not retain states to achieve concurrent access scaling.

Discoverability Avoid creating bad, redundant, ... services. Properly find, identify, and communicate with services using metadata from anywhere.

Composability Combine different services to get new services instead of building from scratch.

2 People as a Service

Tasks that cannot be solved by machines by now or tasks for which machines still need help from humans, e.g., search, research, exploration, processing photos and videos, data cleaning or verification (yellow pages, product catalogues, restaurant details, etc.). People are able to work on tasks from all over the world and communicate and exchange data via different services (e.g., cloud).

For example, in *Amazon Mechanical Turk*, users are able to post jobs or tasks known as Human Intelligence Tasks (HITs) that are then browsed and completed by other users in exchange for money. It is also possible for so called requesters to submit jobs, retrieve completed work

and accept or reject completed work through an API, this is extremely useful for example to generate training data for your machine learning projects. Some other use cases are missing persons searches, social science experiments, information collection, editing and transcription of podcasts, translation, and matching search engine results.

A further example is *Google Answers* which was shut down by December 2006. Anyone could ask questions that were later answered by so-called Google Answers Researchers (GARs). The requester chose a bounty for his or her answer and let do the search by another person which got paid at the end. *Google Answers* provided answers to all kinds of questions like: How many tyrannosaurs in a gallon of gasoline? or why flies survive a good microwaving. Even though *Google Answers* shut down, there are still similar platforms around, like *Uclue* which is probably the most similar to *Google Answers* because it is owned and operated by former GARs.