

Social Interactions Routing

White Paper

June 6, 2014

Version 0.3

Executive Summary

In this paper, we shall consider the integration of social media channels with universal routing systems allowing distribution interactions of social media to contact center agents. The social media service can listen to all social media channels including Facebook and Twitter. The service filters the interactions in accordance with sop preconfigured set of topics. The selected interactions are delivered to a/the routing service that distributes the interactions to appropriate agents. The system could be deployed as a cloud service and be accessible via REST and WebSocket interfaces.

Introduction

In the last few years, we have observed tremendous growth of social media usage. There is an explosion of social communications via blogs and various social networks such as Facebook, Twitter, and LinkedIn. This creates new opportunities for activities involving marketing, brand monitoring, market relations, customer service, sales, etc. However, to take advantage of these opportunities one needs special technical tools enabling listening, filtering and processing social interactions. Such technologies are commonly referred to as social media monitoring platforms.

We shall consider a solution aimed at listening to social media, filtering relevant interactions, and routing them to appropriate operators (aka agents), who could act in response to interactions content. The solution is based on the integration of a/the listening social media service developed by Master-Domino Company (Moscow, Russia) and the Interaction Processing system developed by Contact Technology Labs, Inc. (California, USA). Both are developed and deployed as cloud services with access via Internet.

The structure of the integrated system is depicted in Figure below. The system is comprised of two cloud services – a/the social media service and a/the routing service which are connected via HTTP link.

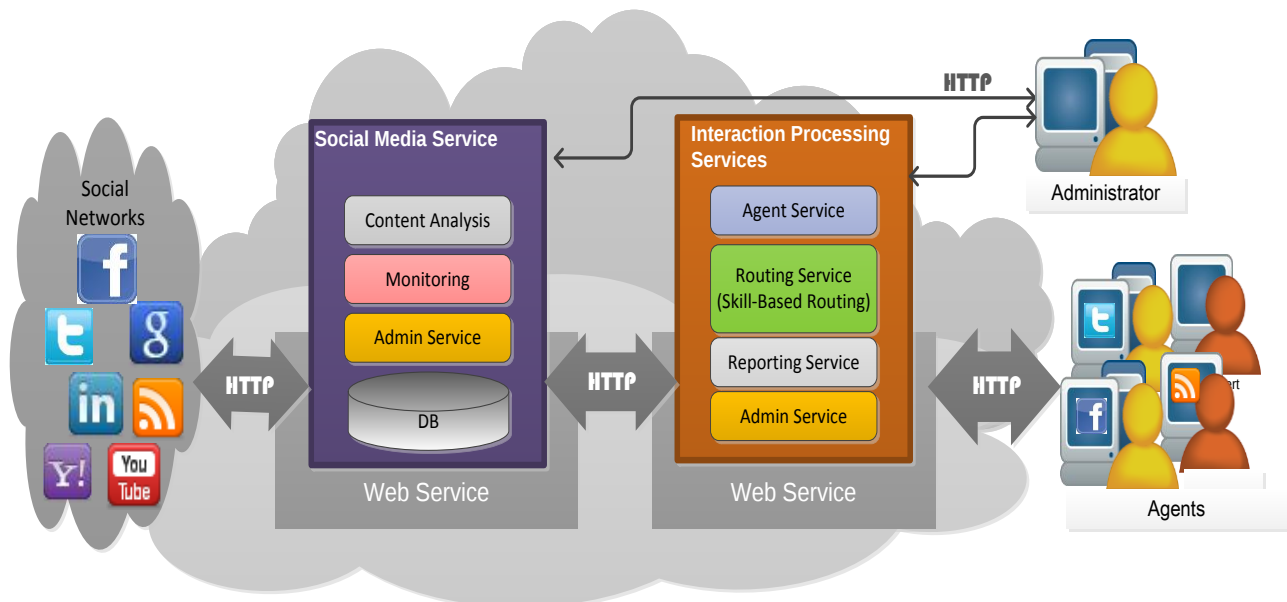


Figure 1: Contact Center in Cloud

Social Media Service (SMS) is a service that monitors various social networks on the Internet such as Facebook, Twitter, and LinkedIn. To accomplish this it employs proprietary APIs to access these networks and to listen to them. An administrator can identify topics of interest and the service filters all posts to select relevant ones. For example, the topic could be the name of an organization, resulting in information on what is being said about this organization. Apart from filtering, the service analyzes content of all interactions trying to identify useful information, such as sentiments (e.g. positive, negative, and neutral). The selected interactions are submitted to an/the interaction processing service that routes them to relevant agents.

Each agent logging in to the system can see a list of topics generated and selects ones she will work with. These topics play role of agent's skills and interactions will be routed to appropriate agents. One agent can work with several interactions simultaneously. When work with some interaction is completed, the agent closes the interaction and it leaves the system.

Interactions could be distributed to agent's desktop by two mechanisms. The first one is called "pull mode", which implies the claiming of interactions by agents. An agent can see all unassigned interactions in the common queue and selects one to be assigned to her and distributed to her desktop. The second mechanism is called "push mode" where interactions are directly assigned and routed to agents.

The next Figure demonstrates a desktop of an administrator of a/the social media service. The desktop shows all interactions of configured topics received by the service.

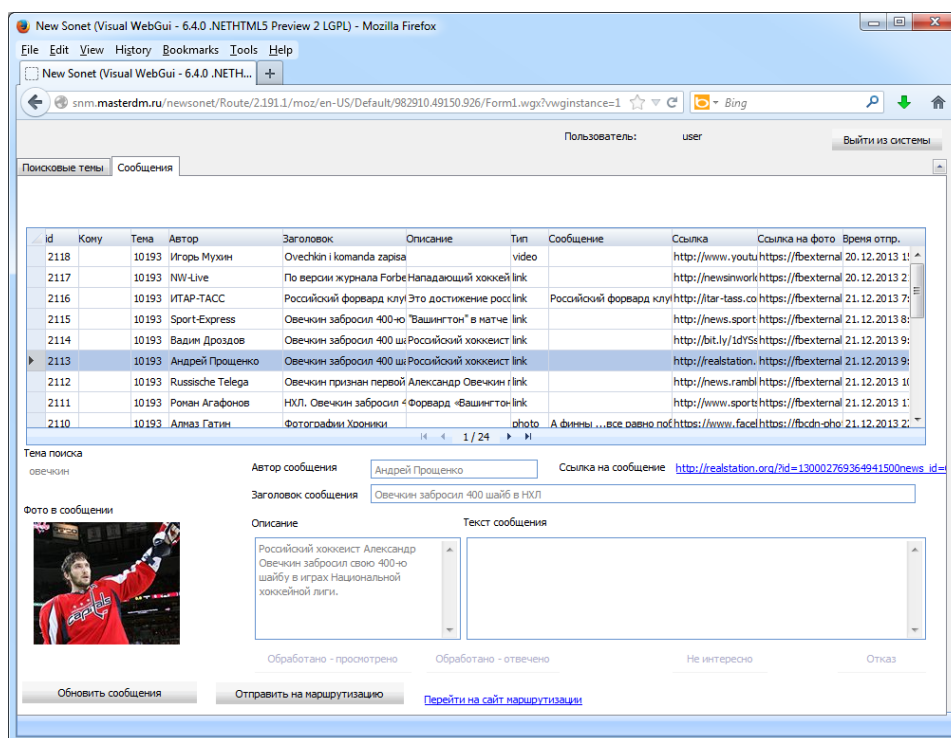


Figure 2: Snapshot of social media service desktop

The next Figure shows a snapshot of the agent's desktop. The desktop contains all active interactions assigned to the agent. He or she may open each interaction and see all interaction data. When processing of some interaction is completed the agent closes it, and the interaction is removed from the system. The agent can also return an interaction back to the system without processing. In this case the interaction is placed back in common queue and again becomes visible to all appropriate agents.

In this example an agent is logged in with a skill "ovechkin" and can receive all interactions with the same topic. In general, an agent may log in with several skills allowing her to receive interactions of different topics simultaneously.

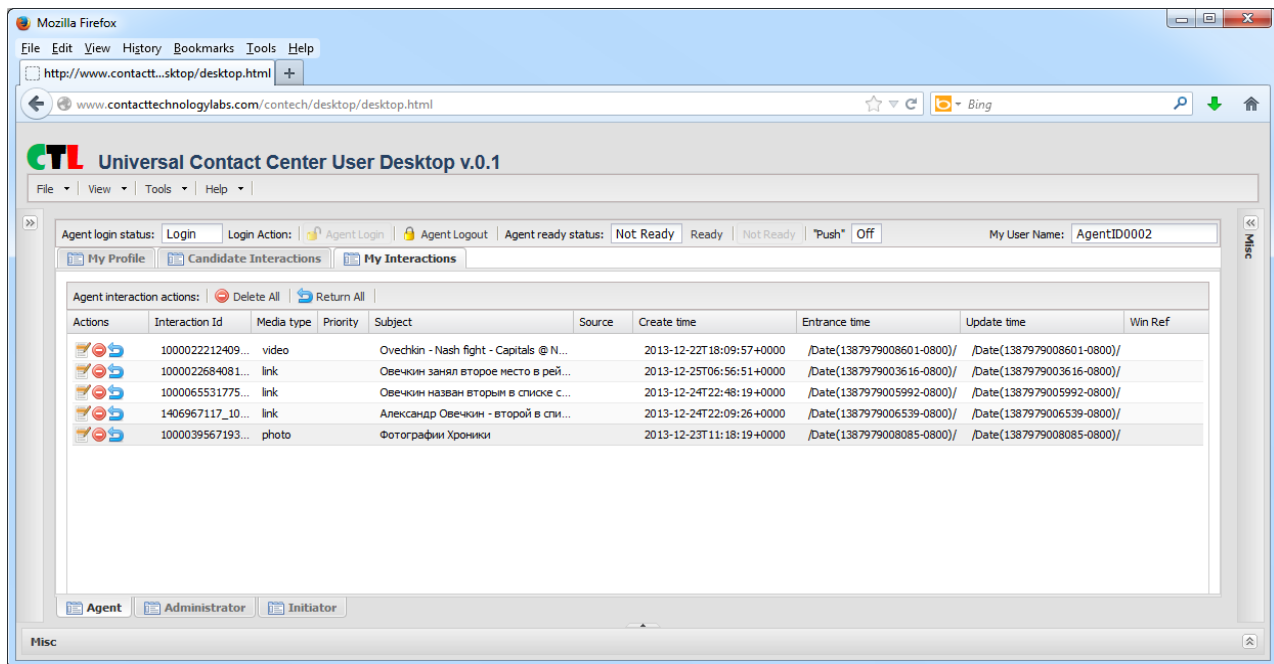


Figure 3: Snapshot of agent's desktop

The following snapshot shows an interaction window the agent opened to work with. The window contains all interaction data including subject and body of the interaction, as well as interaction metadata.

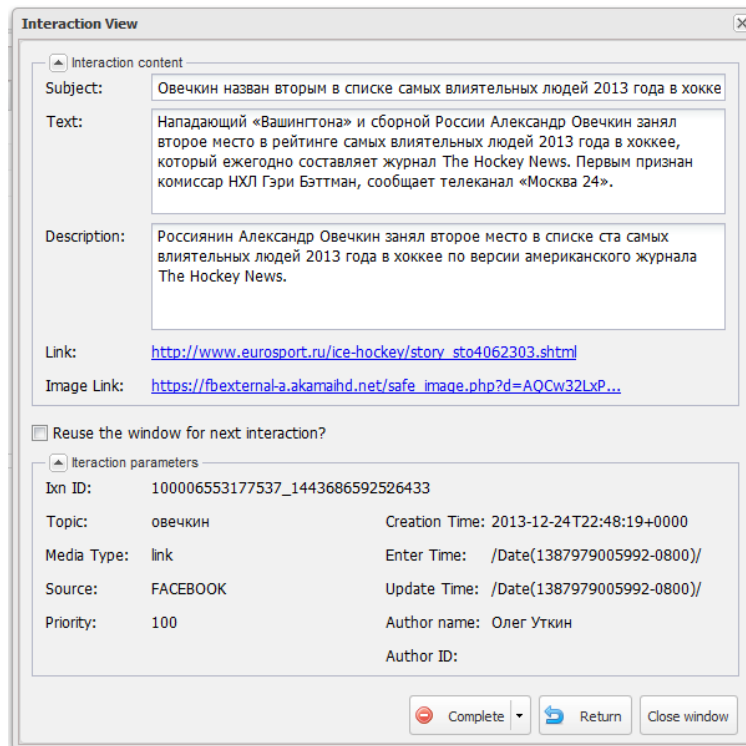


Figure 4: Snapshot of social media interaction window