

# Metasearch System Identification

## Introduction

This is a very specific topic addressing a single area, but one of great interest to both the content providers and the metasearch service providers.

The problem is that generally a metasearch engine (“metasearcher”) accesses content provider target search services (“targets”) in exactly the same manner as do real people sitting in front of a browser and individually interacting with the targets. Because the metasearchers are emulating the end users, the targets have no way of distinguishing a metasearcher from a ‘real’ user – that is, if the metasearchers are doing their job properly!

This means the target has to authenticate the “user” and generate an http/html session framework (consisting of web pages complete with graphics, text, explanations, even adverts) even though the metasearcher will, immediately on receipt of these pages, destroy them and break them down to the essential data elements from which the target built them. This is, obviously, extra work for both the target and metasearcher. And it is both compute intensive work and subject to rapid variation.

Since the pages are designed for human consumption, they will contain style elements and layouts which designers and marketing people and others will wish changed for aesthetic or commercial reasons, rather than for any functional reason. Every such change means work for the maintainers of both the target and metasearcher systems, and a time lag during which they will be out of synch during which searching is not possible.

All this extra work combined with rising numbers of databases, content providers, metasearch engines, and users means the problem is getting larger at an alarming rate. System overload and resultant degraded performance and eventual denial of service will be one result. Another will be increased user costs as maintenance costs rise.

## Focus

We can do nothing to stop the increase in the number of players in this game, and hence the multiplier for the rate of increase.

However it is possible we can do something to reduce the basic amount of work being done.

The topic of this workshop looks at solutions based on the following train of logic:

“A target can interact with a metasearcher much more efficiently than with an end user.”

Efficiency is measured here purely in terms of system resources used – it has absolutely nothing to do with the actual search results returned, they are presumed to be identical in both the end user and metasearcher cases. An efficiency by-product would also be the reduction in the human resource costs of maintaining the increasing amount of connection information and code. This would be reduced as the number of variants would be reduced and the frequency of change would (presumably) be reduced as well.

“A metasearcher can interact with a target much more efficiently if it is not pretending to be an end user”

“If a metasearcher can identify itself as such to a target they can both use the more efficient method”

So we are looking at possible ways that a metasearcher can identify itself to a target so that the target can identify it.

## **Possible Solutions**

These break down into three basic types:

Special addresses

Special attributes

Authentication

Special addresses presume an IP address or port (or both) which is dedicated to metasearch access. This would support the ‘efficient method’ of interacting, whatever the protocol and language involved. This is a sort of “tradesmens entrance” with none of the fancy décor of the front of house, just an efficient way in and out. It is also a mechanism which could support a different quality of service level, or different commercial considerations.

Special attributes assumes the connection to the target is (initially) through the same address as for an end user, but an attribute/value pair (or equivalent in the reigning protocol) is sent in the initial stages of the transaction and the target reacts accordingly by

modifying its method of interaction to the 'efficient method'. In reality the data passed is not limited to a single attribute/value pair and could include information about the metasearcher, and other 'service modifying' information.

Authentication is a modification of the Special attributes method whereby the normal process of authentication of the target recognizes this user as a metasearcher and modifies the protocol accordingly.

Note that in all cases this is a method to identify the metasearcher as such, it is not a method to replace 'normal' authentication and authorization procedures. This would typically (except for the Authentication solution) be handled as the next step in the transaction process.

## **Issues with the Solutions**

Note that the list of possible solutions is not exhaustive, but includes what seem to be the most promising.

All will achieve the desired effect, it is the costs which vary.

### **Special Addresses**

By definition these are reserved addresses and if they are under-utilised this is a waste of resources.

Scaling of this solution may require sophisticated load balancing systems.

This is the easiest solution to implement.

### **Special Attributes**

This requires changes to the initial message parser, and the ability to re-route the metasearcher session (or transaction) to the handler for the 'efficient method'.

Or, it requires a handler which can operate in both the end user and metasearcher methods – a complex beast.

## **Authentication**

This requires the software complexity of the Special Attributes, except that the switching is done as part of authentication, which may be on a dedicated machine or server and so much of the problem is alleviated.

A special class of users (metasearchers) needs to be set up and the authentication modified to recognize them and respond accordingly.

This authentication action is orthogonal to the normal authentication which determines the class of service provided, thus it is not merely a matter of a different ID, but a modification which can be applied (or not) to any ID or other authentication method. This is a serious drawback for IP authentication, but would be easily handled by a token or certificate based system with embedded rights within the certificate.