

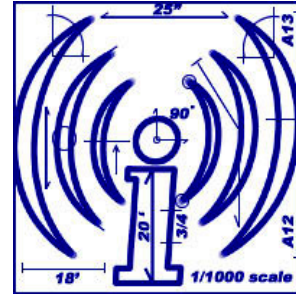
WORKING GROUP: imc-cms

TOPIC: Technical Guidelines for Proposed Proof-of-Concept

AUTHOR: Ryan (SF-IMC, Linefeed, Techmeet, SF Community Colo)

AUTHOR CONTACT: ryan@linefeed.org

DATE: 6 July 2008



Contents

PURPOSE OF THIS DOCUMENT.....	1
BRIEF BACKGROUND.....	1
WHERE WE ARE TODAY.....	2
A TECHNOLOGY PROPOSAL.....	2
THE MALANDRO TECHNOLOGY CONCEPT.....	2
WEBSERVER TIER.....	3
MIDDLE TIER.....	4
DATABASE TIER.....	4
UNRESOLVED ISSUES.....	4
CONCLUSION.....	4
HOW TO GET INVOLVED.....	5

PURPOSE OF THIS DOCUMENT

The purpose of this document is to provide an update on the imc-cms project and articulate a proposed Proof-of-Concept for the Indymedia imc-cms working group. This is not intended to be a complete technical specification but rather an overview of the proposed architecture that has been worked on by the “dev-dot-bunke” sub-group, so named because the efforts have been coordinated at <http://dev.bunke.indymedia.org/>. This document is intended to be used as one point of reference for an upcoming IRC meeting that will prepare participants for a renewed effort to achieve the aims of the imc-cms group as well as Techmeet 2008, where face-to-face discussions about this and many other topics will take place.

BRIEF BACKGROUND

In July 2006, many of the core developers of Mir and sf-active (two of the most-used content management software projects used for Indymedia sites) met face-to-face with many other Indymedia programmers, site administrators and content producers. This opportunity was used to discuss three serious problems facing Indymedia as an organization: 1) the problem that only a handful of people were active on a daily basis to support, develop and maintain these two Indymedia CMS applications was creating bottlenecks and burn-out, 2) Mir and sf-active as well as the various other applications being used to run Indymedia sites (Drupal, dada, etc) were painfully behind the innovations being developed by corporate websites which focus on user-generated content and 3) the Indymedia network is vulnerable to significant downtime caused by server seizures by law enforcement, server hardware failures and sudden withdrawals of volunteered technology resources.

These points were ironically proven shortly after Techmeet 2006 by two crises: 1) the withdrawal of the *ahimsa* server, which hosted a number of IMC websites and 2) the closure of the California Community Colocation Project, which is where the Linefeed organization had hosted a number of Indymedia sites for over five years. These crises further burdened the already over-worked human resources available to maintain the Indymedia websites and

PROPOSAL : imc-cms : WHITE PAPER

the emergency situation delayed progress on the plans made at Techmeet 2006 to rectify these problems.

The decision made at Techmeet 2006 was that the development teams for Mir and sf-active would combine their forces and organize an open call to the Indymedia network-at-large to assist us in solving these problems. We decided that Indymedia was not a big enough organization to sustain a completely custom-built software application and that we would create a list of requirements, conduct an audit and review of existing open source software and decide on an existing project to join and customize per the unique needs of Indymedia.

The response to our open call was good. Many people in the network had already been thinking the same thing and, in some cases, had even initiated their own projects to address these problems (for example, the imc-alternatives working group).

Since then, a lot of progress has been made towards this goal – but not enough progress and not fast enough. We began by forming the imc-cms working group, we sent out surveys to popular open source CMS projects and evaluated their responses, we discussed in-depth the possibilities and challenges we faced, we balanced the day-to-day requirements of our time against the long-term plans for the future, we networked with other IMC volunteers around the world who were interested in this project and we held Techmeet 2007 to solidify that event as a forum for collaboration amongst IMC tech volunteers.

WHERE WE ARE TODAY

Two years later, where are we?

The server crisis we faced in late 2006 was handled by a group of imc-tech volunteers who found space for homeless IMC's on existing servers and the San Francisco Community Colo was founded to provide low-cost server colocation to non-profits as well as a “protected space for free speech and privacy” via relationships with EFF.org and other digital rights attorneys. Unfortunately, these solutions took time away from the long-term solution, although progress was made on that front, as explained in the previous section.

In the first half of 2008, the Techmeet organizers have begun meeting and making plans for Techmeet 2008. We've reviewed almost every open source content management system and come to the conclusion that many (if not all) of them aren't really appropriate for the specific needs of Indymedia. However, we are firm in our decision to not “start from scratch” and whatever solution we decide on must leverage an established, vibrant open source community with many participants who are not a part of Indymedia. In addition, we are committed to re-claiming Indymedia's past position as *the* innovator in online, participatory media. Too many IMC sites are content being “just another blog” out of literally millions of blogs or an online forum for an inbred and insulated group of left-oriented social scenesters.

Towards that goal, the imc-cms group is committed to breathing new momentum into our project that began two years ago.

A TECHNOLOGY PROPOSAL

That said, various work that has been done here and there by various volunteers has the potential to be organized into a proof-of-concept that would satisfy all of the technological requirements set by the Techmeet and imc-cms groups. This proposal comes out of the work of the “dev-dot-bunke” group, where an online wiki and project management system is available for further coordination of this project. This proof-of-concept has been code-named *Malandro*, a cultural export from Brazil that is difficult to exactly translate into English. Afro-Brazilian samba musician Noel Rosa brought popularity to this term. Ironically, a study of the US and Brazilian Wikipedia entries for “malandro” demonstrates the cultural mis-understanding that exists in the US. In Brazil, the malandro is resourceful, cunning, skillful, always planning and working an angle, gets by in life through clever plans against all odds. But on the US Wikipedia page, the malandro is described as “lazy, sluggish” and someone who thinks they are being clever but actually their plans don't work out as expected. This US mis-interpretation is, in fact, a perfect example of how Brazil's folk hero “hedonist” is terribly mis-understood by “hedonists” in the United States who more closely fit the bill of their misinterpretation.

THE MALANDRO TECHNOLOGY CONCEPT

PROPOSAL : imc-cms : WHITE PAPER

The *Malandro* technology approach is a distributed and highly-redundant three-tiered software design that leverages a number of open source projects with large development communities. The design's three tiers are: 1) web server tier, 2) middleware tier and 3) database tier. Each tier would contain as many machines as we could find to add in. Even someone with a static IP on their DSL line could contribute a server to one of the tiers. The goal is to create light-weight servers on each tier that can intersect with each other in any number of ways via the middleware tier. In doing so, there is no single server which can be a single point of failure. If a server is seized, it is merely taken out of the rotation and the system continues to function as normal. This diagram simply shows how the three tiers are organized, with "clouds" in between them to represent the ability of any particular server in any given tier to talk with any particular server in another tier.



In an arrangement like this, the current paradigm of a given Indymedia site being hosted on "stray" or "ahimsa" would be over. Each IMC site would be hosted on a grid of servers.

Through the use of round-robin DNS, a user who attempts to surf to argentina.indymedia.org will get one of many webservers which host a number (or all!) of the Indymedia sites. If the page being served requires data that isn't held on the webserver, the webserver will make a request to the middle tier (which also functions as an object store). If the middle tier doesn't have the data the webserver needs, it will handle the interaction with the database tier and send it back to the webserver.

Similarly, if a user is publishing a story, it will send the request to the middle tier which will handle the database INSERT.

There are specific proposals for the technologies to be used on each tier, which we will look at now.

WEBSERVER TIER

This proposal recommends the use of CakePHP as the front-end application for the imc-cms software. There are a number of reasons for this:

1. CakePHP is a rapid application development framework for PHP5, modeled after Ruby on Rails. It enables extraordinarily fast development of applications while maintaining a Model-View-Controller discipline.
2. CakePHP is widely used (the Firefox Extensions website uses Cake, for instance). It has an enormous and helpful developer community who would probably be overjoyed to hear that Indymedia has chosen to use it and may even promote Indymedia in the process.
3. A number of extensions and helpers already exist for CakePHP, which will make it easy for us to add Web 2.0 functionality to the sites. From standard features like pagination through to Javascript carousel routines are all available and can be incorporated into our front end.
4. Unlike Ruby on Rails, Cake is built on one of the most popular and widely-used programming languages on the web today. PHP is very easy to learn and CakePHP's structure makes sure that beginner-level programmers

PROPOSAL : imc-cms : WHITE PAPER

won't mess things up too bad :)

5. CakePHP was recently adopted by the popular CMS, Mambo, as their choice for a complete re-write. This means that a large number of CMS-related reusable code will be made available to us.

All that said, while imc-cms should have an "official" front-end website, given the architecture described above, Indymedia websites would be able to choose any software of their choice as long as they are willing to stick to our data model and follow the RPC services that our middle tier requires.

More information about CakePHP can be found at <http://www.cakephp.org/>

MIDDLE TIER

The middle tier for this architecture uses the Internet Communications Engine, found at <http://www.zeroc.com/ice.html> ... this powerful middleware was built and open sourced by a gaming company who required remote procedure calls by also needed them to be extremely fast with a simple-to-use interface. Those are the same requirements that we have. The imc-cms team (thanks to Occam and Toya) already have an ICE installation going with some Python scripts that publish articles. There are a number of reasons for using ICE:

1. ICE is easy to use, thread-safe with an exception-safe API.
2. ICE allows our web tier to dump off instructions to another tier which can quickly handle the heavy lifting in whatever language we want. ICE supports C++, Python, PHP, Ruby and even C#. ICE will do the object translations for us.
3. ICE has built-in load balancing. We set up an ICE Grid and it figures out for us where the request should go.
4. ICE is a fully-featured middleware: it supports broadcast messaging, automatic patch upgrades, SSL, persistent objects and support for mobile and embedded devices.

By using a distributed middle tier, we can further decentralize the Indymedia server network. And, ICE is an open source platform with a developer community so all we have to do is set up and configuration.

DATABASE TIER

This proposal recommends MySQL as the database backend of choice. Yahoo successfully used geographically-distributed MySQL servers for years and we have a number of options available to us as far as configuration goes. We can run a cluster, or we can do standard MySQL replication, or we can look at this project which we found out about this weekend from Acracia - http://giss.tv/wiki/index.php/Distributed_Multimedia_Database_System

UNRESOLVED ISSUES

The main issue that remains unresolved with this approach is how to handle uploaded media. A suggested solution is that whichever webserver received the uploaded file notifies ICE that it has the file and ICE lets other webserver know this until the media file is able to be propagated out to other servers.

CONCLUSION

This approach solves all of our problems and meets all of our requirements. In addition, the workload can be split up into smaller teams who can tackle the individual problems and then we unite them all together. As an added bonus, this is an innovative solution which could help IMC get back some of the tech respect that we deserve.

PROPOSAL : imc-cms : WHITE PAPER

HOW TO GET INVOLVED

We do need all the help we can get to save Indymedia! Are you content with having Indymedia being a blog that hasn't changed at all in nearly a decade? Are you happy that Indymedia can no longer impact the ideas of people beyond the leftist playpen? If so, get involved with our project! Here's how:

1. Become part of the imc-cms group - <http://lists.indymedia.org/imc-cms>
2. Get involved with Techmeet - <http://www.techmeet.org>
3. Come to the IRC meetings, announced on the imc-cms mailing list
4. Come get to know us on IRC - if you don't know how to use IRC, you can use <http://chat.indymedia.org/> ... the official imc-cms chat is in the #cms channel and the accompanying social channel is called #sfkids
5. Donate money! We need money for Techmeet, for server hosting, for the time it takes to code and organize this thing, etc. Get in touch with us to find out how.