ARCHITECTURE OF THE INTERNET AND THE CONSEQUENCES

Freedom. An ideal and right that many cherish in the United States, yet within the context of the Internet, it is easily overlooked. Freedom is a fundamental principle that is implicitly built into the current architecture of the Internet. Any message, any data a user may wish, can be transmitted from one location to another cheaply, quickly, and unhindered. Bits are routed without knowing what the content is, who the sender is, or who the recipient is (unless you define an IP address to be a person)—much of the freedom hinges on anonymity. This anonymity extends beyond who, it includes where. “The Net enables transactions between people who do not know, and in many cases cannot know, the physical location of the other party” [Joh ‘96]. The Internet is a virtual space with its own set of norms, namely that of freedom and anonymity, that also physically exists in reality. However, because of the lack of geographic connection to Internet traffic, it is unregulable with respect to geographic, namely political, boundaries.

So the freedom that is built into the Internet effectively prevents sovereign states from enforcing their laws on their own citizens’ behavior on the Internet. Conversely, servers in one country might provide content that is legal in the country of origin yet illegal in the country of destination. National lines and laws are simply transcended, but world has far too complex a political landscape to expect all nations to accept the inherent freedom of cyberspace. A 2000 CNN article highlights the problem: ““Laws grow out of historical context. What is appropriate in Germany may not be appropriate in the U.S.,” said Mark Weitzman, director of the Simon Wiesenthal Center task force on hate. "I don't think one society should be able to impose its values on another" [Internet]. There are many examples of such conflicts arising on the Internet; the dispute between
Yahoo!, Inc. and La Ligue Contre Le Racisme et L'Antisemitisme (LICRA) of France will serve as a model example and springboard for this paper [Yahoo!]. Yahoo, a popular Internet portal, provided a link from its <www.yahoo.fr> page on its French server to a page on its American server, www.yahoo.com, which contained an auction for Nazi and Third Reich propaganda. It is illegal for French citizens to view/purchase this type of merchandise, and since Yahoo! provided access to this content, the French court held Yahoo! liable, ordering the company to prevent French users access to the violating pages at a penalty of 100,000 Euros per day for failure to comply. Yahoo! argued that they could not technologically comply with this and filed with a California court, arguing their First Amendment Rights were being violated. The U.S. court sided with Yahoo, stating that France’s ruling has no authority outside France.

This court case makes the problem with the current architecture of the Internet manifest. Physically, Yahoo! is subject to U.S. sovereignty as that is where their servers are located. Yet those same servers provide illegal content to users that physically reside in another jurisdiction. Hypothetically, if Yahoo!’s French servers were to contain 40% Nazi propaganda related material, the only perfect solution would be to physically remove each page or shut down/block the entire server. These unappealing situations are not a consequence of the government or market conditions, but a direct result of the Internet’s architecture.

The current architecture, while responsible for norms of freedom and unregulability, does not have to remain permanent. As Lawrence Lessig argues in Code and Other Laws of Cyberspace:

“There is no single architecture for cyberspace; there is no given or necessary structure to its design. The first-generation Internet might well have breached
walls of control. But there is no reason to believe that architects of the second generation will do so, or not to expect a second generation to build in control…. We can already see the beginnings of this reconstruction. Already the architecture is being remade to re-regulate what real-space architecture before made regulable. Already the Net is changing from free to controlled” (167).

The architecture of cyberspace can be modified to accommodate national sovereignty. The Internet is not a constant; its architecture can be molded to allow regulation, and the re-shaping process can be guided, even initiated by governments themselves. There are those that would argue that cyberspace deserves its own separate, new type of sovereignty that disregards physical boundaries. (See Joh, ‘96 who argues, “…[law governing cyberspace] will not, could not, and should not be the same law as that applicable to physical, geographically-defined territories.”) However, for the purposes of this paper, we rely on the notion that sovereign nations would seize the opportunity to enforce their respective laws within their own physical borders if presented with ability to do so. The United States has shown this precedent with the passage of the now defunct Communications Decency Act of 1996 and the Child Online Protection Act of 1999 (currently awaiting , both of which attempted to regulate minors’ access to pornographic material [Web Censorship]. It is the goal of this paper to discuss the possible solutions to the problem and to provide a reasonable, feasible solution to this tussle in cyberspace so that web users will be accurately and consistently subject to the laws of the physical place in which the users surf. The political ramifications and the ethical implications regarding control, regulation, and censorship on the Internet is debatable, but not the concern of this paper, which is merely that a level of control that governments find acceptable is established.
Mores specifically, the singular problem of “speech”, the information and content of the Internet itself, and the different laws regulating speech across different sovereign boundaries is being addressed. For example, “What constitutes “political speech” in the United States (Nazi speech) is banned in Germany; what constitutes ‘obscene’ speech in Tennessee is permitted in Holland; what constitutes porn in Japan is child porn in the United States; what is ‘harmful to minors’ in Bavaria is Disney in New York” [Res 395].

In the physical world, governments regulate access to banned speech; this is the type of regulation which will be discussed. The enforcement of other types of laws, such as intellectual property laws like copyright and trademark, is not within the scope of this paper.

**EXISTING SOLUTIONS**

Before discussing the possible ways to modify the architecture of cyberspace to allow for greater control, it is necessary to show that solutions within the current architecture are insufficient. As discussed earlier, there is a high degree of anonymity on the web, and when a user visits a web page, the server has no indication as to where the user is physically located. Within this architecture, the only option really available to countries is the establishment of a “blacklist.” There are two main ways in which blacklists can be effected, neither of which is a sufficient solution to the problem at hand.

First, a government can force/mandate the search engines with servers in their country to filter out pages on the government’s blacklist or results for blacklisted terms. For example, Google.de and Google.fr omit results on white supremacy and neo-Nazi information on its German and French servers respectively that do return when queried from Google.com [Big Search Engine]. While this limits Google’s liability within those
countries, it does not deny access to those pages if the user knows the actual web address. Search engine filtering is a pre-emptive strike that prevents citizens from stumbling across illegal content, but it is weak at best.

A more powerful method is for a country to set up a firewall or proxy server and permanently block blacklisted URL’s, entire domain names, or even all information from specific IP addresses. For example, China blocks access to many of the Google results from the search terms like: “Taiwan”, “equality”, “revolution”, “democracy China” (How to be). Countries that block sites may block an entire domain, even if only some of the web pages associated with the site are deemed inappropriate. In China again for example, all of Columbia University’s websites were blocked from the top on down, presumably because a campus pro-Democracy site is hosted on the server. It is infeasible for a government to rake the entire Internet looking for specific web pages to ban. Conversely, banning entire domains and IP addresses is a shotgun approach at best, often blocking permissible content along with the objectionable material. Manual blocking via firewalls/proxies is not a sufficiently robust or scalable solution.

POSSIBLE SOLUTIONS

For speech to be regulated in cyberspace, there are three pieces of information that must be known in order to effectively decide if a user is blocked: first, the jurisdiction in which the user resides, second, the type of content or speech that is requested from the server, and third the laws of the sovereign state in which the user resides regarding the type of content being requested. Under the current architecture of the Internet, little to no reliable information indicates the geographical location of a user. A user usually doesn’t know exactly what kind of speech is being sent until after it is
received, and servers don’t maintain any state regarding the classification of content. The laws of states are simply unknown and absent within a typical Internet transaction. Note that by this definition it is the user’s jurisdiction that regulates the content, not the server’s. (The converse system could be used where the sender’s jurisdiction regulates the content, but that will not be discussed in this paper.) An implicit consequence of this definition is that it is not necessary for the server’s jurisdiction to be known by the entity doing the regulating.

Before getting into a discussion of the architecture for obtaining this information, let us hypothetically assume the architecture a priori exists. In such a world, there exist two effective, yet different ways to regulate speech on the Internet. The point of difference lies in the actual object of regulation or blocking. In one method the pages (content) themselves would be filtered if they are illegal for the user to view. In the alternative method, the servers would be responsible for the “filtering” by blocking access to users who can’t prove that they have the right to view the content. Both solutions require a restructuring…as Lessig puts it: “Both solutions – placing the burden on the recipient or on the sender – require a new architecture for the Net, not at the level of the TCP/IP protocol, but in the application space” [Les 175]. While both methods accomplish the same thing, there are different prerequisites and outcomes. A full discussion of the two options is warranted to make an informed decision on which is the better solution.

**FILTERING: A CLOSER LOOK UNDER THE PICS MODEL**

An architecture for filtering already exists, yet has not been implemented in a way that would resolve the conflict at hand: this is the World Wide Consortium’s “Platform
for Internet Content Selection”, known as PICS. (Full details can be found at http://www.w3.org/PICS/). PICS in and of itself is not software for filtering pages based on regional laws, but it does provide the specifications to do so. Essentially, PICS provides a standardized platform for the labeling of content. In a PICS enabled web page, content-labeling tags based on an accepted vocabulary are written into the header of the HTML file. This labeling can be done by the content creator or by a third party. Software is then used to filter pages by comparing tags to an accepted rules list, which is already standardized by the PICSRules language [PICS FAQ]. As the creators of PICS envisioned their platform, end users would use their personal browsers to filter out unwanted pages based on personal preference. In the context of national sovereignty, this is inadequate, as individuals cannot be trusted to filter themselves correctly. The solution then is “upstream filtering:” “Filtering can occur at any level in the distributional chain—the user, the company through which the user gains access, the ISP, or even the jurisdiction within which the user lives” [Les 178]. A PICS based software solution can be implemented using a trusted third party, most likely the Internet Service Providers, which conceivably could be forced to filter out content based on government defined rules and laws. The burden for determining the necessary information for regulation is not placed on the content providers but the access providers. So by default all web pages are still viewable; the offending pages however are checked against a defined set of rules and filtered as necessary the third party.

Upstream filtering using PICS creates several problems, and ultimately we must discard the filtering of content in our search for a viable solution for several reasons. First, while a specification for labeling may exist, the actual process of labeling web
pages on the Internet is a massive undertaking. Whether done by the government itself or a hired 3rd party, accurate rating or labeling would be costly, lengthy, and assuredly not one hundred percent accurate. Moreover, different sovereignties would likely argue over the labeling as social norms vary from culture to culture. This means each sovereign government partaking in the process would need to label pages independently. There are also social implications in that URL filtering can be invisible. “If you cannot see the content, you cannot know what is being blocked. In principle at least, content could be filtered by a PICS filter somewhere upstream and you would not necessarily know this was happening” [Les 179]. In cultures that espouse freedom whilst eschewing censorship, there is a vested interest of the citizens to know exactly what is being blocked. Only if the denial of access is explicit can citizens make an informed response to the filtering, pro or con. The invisibility of filtering also lends itself to abuse. If citizens do not see what is being filtered, the government can easily over-filter content, blocking its citizens’ access to legal, if controversial, material. Based on the scale of labeling, its inherent subjectivity, and the consequences of upstream filtering, we conclude that a solution that blocks web pages based on content is an inappropriate way to respect sovereignty on the Internet.

**AUTHENTICATION: CERTIFICATES FOR EVERYONE**

As mentioned, the other method for regulating access to content on the Internet is by placing the burden of responsibility on the content providers. Before serving a web page, the server’s task is three-fold: (1) determine, with high confidence, the jurisdiction of the user and (2) determine the nature of the content and whether it is regulable, and (3) whether that jurisdiction allows the requested content to be served. To simplify the
required architecture for such a system, each user could be assigned an encrypted digital
certificate signed by the government or local jurisdiction. When a regulated web page is
visited, the certificate is automatically presented. The server then consults a local
database that maintains a list of rights defined by the participating sovereignties and
authenticates or denies accordingly. By default then, denial is the rule, and access is the
exception contingent upon a valid certificate.

This architecture of authentication avoids some of the pitfalls noted with a PICS
based filtering solution. Creation of certificates is easy and cheap—a government could
readily create a few dedicated servers for certificate creation and distribution. Also,
there is no invisible filtering. If a user is denied access to a given web page it is made
explicitly clear. A certificate based “zoning” solution is also less subject to abuse than
upstream filtering: “A zoning regime is not only less transformative but less enabling (of
other regulation)—it requires the smallest change to the existing architecture of the Net
and does not easily generalize to a far more significant regulation” [Les 181]. This is
important to countries with a norm of freedom, like the U.S, as many freedom-loving
citizens would be naturally circumspect of any widespread censorship. A certificate
based solution that places the burden of regulation on the content servers, while not
perfect, is the most attractive model for enabling sovereignties to enforce their laws on
their on their citizens’ Internet behavior.

A PROPOSED SOLUTION IN DETAIL

The rest of the paper is a thorough, technical discussion of a solution devised by
the writers of this paper. This will discussion will include all the key players involved:
the government, private citizens, ISP’s, and the content providers. The makeup and
issues surrounding digital certificates and the technical implementation required by
servers will also be explained. No system is without flaw, so the shortcomings and issues
regarding feasibility will also be acknowledged.

A running model of the proposed system is included with this paper and
demonstrates the ease and viability of the system. For more information see the
programming documentation included with this paper.

WHAT’S THIS ABOUT A DIGITAL CERTIFICATE?

A digital certificate provides a “means of verifying identity on the Internet” and is
“also a way of creating trust between groups conducting transactions that may involve
personal information” [Digital Certificate FAQ]. Most commonly public/private key
(PPK) encryption is the algorithm used for establishing this trust. A detailed description
of PPK is outside the scope of this paper, but can be found in Applied Cryptography:

In the proposed solution, digital certificates would be created and assigned to each
machine that accesses the Internet. There is no need to assign certificates to people
individually; one per machine is sufficient. Unlike personal digital ID’s that contain
relevant personal information for e-commerce, these certificates would only contain a
string describing the jurisdiction to which the machine/user is subject, the IP address of
the machine, and the dates or time indicating the length of validity.

The choice to exclude personal or unique identifying information of individuals is
purposeful. First of all, this prevents the architecture from being abused. Because there
is no personal information, tracking or analyzing the habits of users cannot be monitored
by observing the use of the certificates. Regulation cannot be extended to a deeper or
more personal level. Also, this minimizes the risk of a lost or compromised certificate. If a malicious person obtains another person’s certificate or if a private key becomes public, there is no loss to the certificate owner. The issue is privacy, which is maintained, and must be at the risk of losing support or creating a public outcry. As a side note, during the developmental stages of this model, a unified database that mapped IP address to jurisdiction was briefly considered over certificates. While a 1:1 mapped database provides O(1) lookup and validation, it was dismissed based on the premise of privacy. Most likely countries would not want a list of all their IP addresses freely available on the Internet.

The decision to include IP address is also deliberate and necessary. By including the IP address, the receiver of the certificate has a way of verifying that certificate truly belongs to the sender. When a connection is established between end user and server, the IP address of the other party is known. By comparing the IP address of the connection at run-time with the IP address included in the certificate, the server can be trust that the machine is the one it purports to be.

The certificates’ date stamp or expiration date would be decided by the respective sovereignty. The certificates should probably have a limited lifespan to prevent users from taking their machine, and certificate, to a neighboring country with different laws and using the old certificate. This would most likely be thwarted anyway as this shift in physical region would be accompanied by a change in IP address. In fact, a new certificate might be obtained for every new session the machine makes connecting to the Internet.

WHO’S CERTIFICATES ARE THEY REALLY?
Having a model for the certificate, who will do the creating and the distribution? It is proposed that the Internet Service Providers will distribute the certificates to machines. This makes sense because most machines connect through a provider in their local jurisdiction, so the ISP knows this information, along with its subscribers IP addresses, by default. Each time a dial-up connection is made, which use transient IP addresses anyway, a new certificate would be sent to the machine and stored in a predefined place on the user’s hard drive. For always-on broadband connections, the certificates might last longer, and new ones might be “pushed” to the user less often, perhaps every other day.

The creation of the certificate is a different issue. Who does the creation depends on the government’s level of trust in its national infrastructure, and who will bear the economic burden? The government itself could be responsible for the creation of certificates. In such a model, when a machine connects to an ISP, the ISP would relay its jurisdiction and the machine’s IP address to the governmental server. The governmental server will create the certificate and encrypt it with its private key, add an unencrypted string indicating jurisdiction again (reason to be discussed later) and return this information to the ISP, which in turn sends the certificate to the client machine. The government is economically responsible for the certificates and is the source of trust for the certificate, as only the government’s published public key can decrypt the certificate.

Alternatively, a trusted third party appointed by the government could perform the same function. The government would have to trust such a company to a large extent as the enforcement of national sovereignty is at stake. In a more direct method, which again places a large amount of trust in a private party, the ISPs themselves would create the
certificates. This would reduce the extra lag in waiting for the certificates to be generated and returned to the ISP. If the ISP’s can be regulated to the necessary extent by the government, then this would be the most efficient way to automatically create and send the certificate when a user “signs on.” Note that this is the way the accompanying program works.

**JURISDICTION DEFINED**

Thus far it has been implied that the jurisdiction applicable to the certificates is that of the national or federal governments. While this is most certainly an option, the certificates need not be confined to one level in a hierarchical state. Jurisdictions can be subdivided into any number of groups. For example, at Duke University one might obtain a certificate defining the jurisdiction to be “NORTH_CAROLINA_USA”. This certificate would first be subject to the rights of North Carolinians, while also inheriting the rights of general United States residents. Special certificates could even be created for specific types of people, e.g. “CHINA_PRIVELAGED_MEMBER” which might entitle the holder to more rights than an average citizen would enjoy. The system for defining jurisdiction is open, and hierarchical when necessary. The implemented program demonstrates how this might be put into effect.

**CLIENT-SERVER PROTOCOL**

At this point the user’s machine has the previously discussed digital certificate stored on its hard disk as a small file. Ideally, the user has not had to wait more than a few seconds for the certificate, thus the system so far is neither intrusive nor cumbersome. Now the user visits a web page that has chosen to comply with the certificate system. If the server’s content is illegal in at least one jurisdiction, then before
the content can be sent to the machine, the server must validate that the user has the right to access its content. (If the content is completely unregulated, then there is no need for the extra Internet traffic.) The following is a description of this protocol.

First, the server requests that the certificate be sent. This request is received by the client’s machine and the certificate is sent automatically, without the user being burdened. This is significant: the burden of presenting a valid identification is negligible to users of the Internet. If users were burdened in even a slight way, e.g. with a noticeable time delay or the requirement to present a password, then more than being unattractive, the system would be unconstitutional in the United States [Les 177].

So the user’s computer has sent the certificate to the web page. This connection need not be secure, due to the previously detailed fact that there is no loss, economic or privacy-wise, in the event of malicious 3rd party snooping of the certificate. The added benefit is that the validation procedure is not slowed by a further level of encryption. This is not to say that a secure connection between user machine and server couldn’t be established. If that extra level of security is desired, it may be performed.

If you recall, the certificate also contains an unencrypted string which indicates jurisdiction, which, while not inherently necessary, provides the server with an ostensible first place for lookup. The server then uses the public key associated with this jurisdiction (whether government, ISP, or 3rd party based) to decrypt the certificate. If the certificate is successfully decrypted, then the server can be sure that the certificate was indeed created by the correct authority. For this to be efficient, servers would need to maintain a small database (or any sustained data structure) mapping all the jurisdictions to the public keys used to create the certificates in those areas. The database
would use negligible space. In the event that the user altered the unencrypted part of the certificate, the decryption would run through the different public keys until the correct one was used. In the event the certificate can’t be decrypted, it is declared invalid and the user is automatically denied access. If the certificate is successfully decrypted, first the date/time stamp is checked to confirm that the certificate has not yet expired. Next, the IP address contained in the certificate is compared to the actual IP address of the connected machine which is verified by a quick challenge-response protocol as the program model demonstrates: the server sends the client a random sequence of bits, which the client must return. If the IP addresses do not match, then the server denies access; otherwise it proceeds to the next step.

At this point the server has reliably validated the jurisdiction in which the machine resides. Now the server must determine whether or not the government allows the requested content. For this to process efficiently, the server maintains an internal database that maps all the jurisdictions to all the possible access rights and whether or not that jurisdiction grants each right, this information being binary, 0/1 as true/false. By default, all rights would be set to true for each country. However, governments would have the responsibility of making their access control lists easily available to servers for download and periodic updating (a weekly update, conducted at a random time, should be sufficiently frequent while not generating too much extra or concentrated traffic). Because the governments define these lists, they can be tailored to each sovereignty’s laws. The United States might choose to enforce a denial of its citizens’ access to child pornography, while Germany might add pro-Nazi speech to this list. An agreement between different nations to standardize the descriptions of rights would be beneficial for
simplicity’s sake, but is not necessary for the system’s functionality. There’s no reason a server’s database couldn’t contain both “porn_right” and a “pornography_right”. At this point, the server knows the location of machine and the rights of machine in that location and can actively approve or deny the user for access to the server’s content.

The entire process is invisible to the end user, but they will obviously know in the end whether or not they can gain access. The process is sufficiently fast that it shouldn’t be an inconvenience to general Internet traffic. There is a further point for expedition though. Instead of comparing at run time whether a user from a given country has the right to view a given page every time, the server can simply map approval or denial to its own index of web pages for each jurisdiction. For example, the server might simply store the information that page “index.html” should not be accessed by any user in Germany, and so forth.

The only code that does actual validating and comparing in our model resides in the application space of content servers. This is a front-end solution and is not inherently hard to implement. Ideally, code could be automatically generated and inserted into web pages themselves in order to comply with the certificate verification model. The following is the server protocol for access verification succinctly described by pseudo-code:
Request received from a user “client” for a webpage “content” on “server”;

bool valid = false;

if (server.content.Is_Regulable() )
{
    Certificate cert = get_certificate(client);
    while (cert.decrypt(cert.associated_public_key) != true)
    {
        cert.associated_public_key = next_public_key_from_database;
    }
    if (cert.Is_Decrypted() && cert.Is_Not_Expired() )
    {
        if (client_IP == cert_IP)
        {
            if (cert.jurisdiction.HasRight(content) )
            {
                server.serve(content);
            }
        }
    }
}
else
{
    server.serve(content);
}

JUDGING CONTENT

Thus far, all of the above code has been explained except for one line: content.Is_Regulable(). How does the server know what type of content a web page contains, and what is the process for classifying the content. The proposed solution maintains that it is the server’s responsibility to be aware of the type of content it is serving and block accordingly. The content does not have to be explicitly labeled for the approval of third parties like in a PICS-based filtering model. However, servers need to internally maintain which rights from the sovereignties’ access list are relevant to each regulable web page. The programming model developed does not actively maintain an
awareness of the type of content the server stores. However the entire protocol, including
the judgment of content, could be implemented using a digital rights management
language based license placed as a separate file on the server, or even used to embody the
digital certificate itself. The next part of the paper provides a background on digital
rights management languages and offers such an implementation.

**DRM LANGUAGES: DISCUSSION AND IMPLEMENTATION**

“Solutions to DRM challenges will enable untold amounts of new content to be
made available in safe, open, and trusted environments.” [Ianella 5]. There are many
uses for digital rights languages, however this paper will discuss how these languages can
be used for the purposes of respecting sovereignty on the Internet.

Any digital rights language must have the following four features: First, the
language must be comprehensive [The Need 4]. This means that the language must be
able to express rights expressions that can be simple or complex. Whether an owner of
some piece of content wants to make the content accessible to everyone, or whether it
should be made accessible only to people in the United States who pay five dollars to the
owner per viewing, both such expressions should be possible with the language. Second,
the language must be generic [The Need 4]. That means that the language should be able
to describe rights for various types of content or services. The language should work
whether describing the rights for a physical book, a web page, or a service. In being
generic, the digital rights language should be extensible, so that the digital rights
language is not tied to a specific set of technologies. For example, although public key
encryption is discussed as a security measure in the implementation proposal, there is no
guarantee that such technology will always be the best or most widely used. Third, the
language must be precise. [The Need 4] There should be no ambiguity in the system. Given a person’s credentials (jurisdiction) and conditions (subject to local law), one should be able to algorithmically determine whether the person has the right that they are requesting, and if the person does not have that right, what is needed to get those rights, assuming that something can be done. Finally, the language must be system interoperable [The Need 4]. The architecture of the system that you are using should not matter in terms of being able to use the rights language. You also want to be able to ensure confidentiality of the language document, and you want the ability to authenticate and validate the entities that are part of the language document, in the case at hand by effectively handling the certificates.

The two languages that will be discussed in this paper are XRML and ODRL. Both follow the principles of the <indecs> model. [Iannella 2] In this model there are three core entities: Users, Contents, and Rights. Users own rights over content, and users can create and use the content. Any metadata, or rights language, must include a mechanism for linking these three concepts together [Ianella 3]. In order to specify the rights for objects (in this case the web sites themselves), the objects must be uniquely identifiable. For the purposes of this paper, the URL of the web page can serve as a unique identifier.

**XRML**

There are several reasons why XrML or eXtensible rights Markup Language, is a good choice for a digital rights language. XrML while maintained by one organization, is intended to be open, and it does take input from industry members. XrML is useful for any business model. XrML is extremely interoperable. XrML is extensible [The Need
5]. Also, XrML currently has a very good Language SDK (Software Development Kit), which allows for the easy creation and reading of XrML documents. This SDK can be used in applications that want to implement access controls based on XrML documents. In addition to these abilities, the XrML SDK also includes the ability for signature and trust verification. This makes it easier to actually concentrate on the access, and less on the cryptography issues. Also, because the XrML standard is maintained by one organization, the SDK will always be in strict adherence to the current version of the XrML language. All documents being in strict adherence to the language standard helps to increase operability: any two programs written using this standard SDK will be compatible.

The core concepts of XrML are licenses, grants, principals, resources, and conditions [XrML spec 9]. Licenses are containers of grants [XrML spec 10]. The license consists of a set of grants that convey to the principals the rights to resources under the specified conditions. In addition, the license contains the issuers of the license, represented as principals. Finally, the license can include additional information. An example of a license file, which allows anyone to view the resource, is as follows. Notice that the license file also contains the information about the sender, encrypted so that it can be verified:

```xml
<license licenseId="Example1">
  <grant>
    <mx:print />
  </grant>
  <digitalResource>
    <nonSecureIndirect URI="http://www.contentguard.com/sampleBook.spd" />
  </digitalResource>
  <issuer>
    <dsig:Signature>
      <dsig:SignedInfo>
        
      </dsig:SignedInfo>
    </dsig:Signature>
  </issuer>
</license>
```
As can be seen here, the format can easily handle the verification of certificates. A license like this could even serve as the template for the certificates used in the proposed solution. These licenses then would be given to the user every time the user logs on to the ISP, and would be resent when the license file has expired.

Grants are elements that are subparts of the licenses [XrML spec 11]. The grant lets the principal have authorization of a resource for a specific right (or action). In addition, the grant can contain conditions that must be met in order for the authorization to occur. In our program model, we use the do_grant column of the rights table to express the grants. See the programming documentation for more details.
A principal is the identify of someone to whom rights are granted. Each principal identifies one party. In addition, principals support the following identification technologies: “A principal that must present multiple credentials, all of which must be simultaneously valid, to be authenticated…A keyholder, meaning someone identified as possessing a secret key such as the private key of a public / private key pair. Keyholders are represented using the KeyInfo technology from XML DSIG.” [XrML spec 12]

The right is the verb, or action, that a principal can be given rights to under certain conditions [XrML spec 13]. Rights will often specify actions that a principal can do to a certain resource. For example, if a user can read a file, but not modify the file, then this set of rights would be expressed here. This is where the list of actions is implemented, as different rights.

The resources are the objects that are granted rights to principals [XrML spec14]. For our purposes, each web page is viewed as a different resource, so each web page will need to be identified uniquely by its URL.

“A condition specifies the terms, conditions, and obligations under which rights can be exercised. A simple condition is a time interval within which a right can be exercised. A slightly complicated condition is to require the existence of a valid, prerequisite right that has been issued to some principal. Using this mechanism, the eligibility to exercise one right can become dependent on the eligibility to exercise other rights. Moreover, a list of conditions can be put in conjunction to form a condition requiring that the conditions all be true simultaneously.” [XrML spec 15] For our purposes, the condition would be that if a page contained a certain type of content, then
he must be from a pre-defined list of countries. Moreover, the condition that the certificate has not expired must also be met.

**ODRL**

Open Digital Rights Language, or ODRL, is another digital rights language. Unlike XrML, it is less controlled by one organization, and is therefore more open. However, there is not one SDK that nearly everyone would use, so this has the potential to hurt operability. “The ODRL provides the semantics for DRM expressions in open and trusted environments whilst being agnostic to mechanism to achieve the secure architectures.” [ODRL spec 3] This makes ODRL extremely flexible, even more so than XrML.

As is the case with the <indec> model, the ODRL model includes three core entities: assets, rights, and parties. [ODRL spec 6] Assets are used to refer to the actual content. The assets must be uniquely identified, and the assets can consist of different subparts which can be in many different formats [ODRL spec 6]

Rights include permissions, which can contain constraints, requirements, and conditions. Permissions are the activities that are allowed on the assets, for example, playing or viewing a file. Constraints limit the permissions. [ODRL spec 7]. There are four main groups of permissions: usage, reusage, transfer rights, and asset management. [ODRL spec 10]. For example, a permission tag would look like:

```xml
<permission>
  <display/>
  <print>
    <constraint> … </constraint>
  </print>
  <annotate/>
</permission>
```
Requirements specify that a certain action must be taken in order to actually view the content. There are three groups of requirements: fee, interactions, and usage. An example of a requirements tag would be:

```
<play>
  <requirement>
    <peruse>
      <payment>
        <amount currency="AUD">20.00</amount>
      </payment>
    </peruse>
  </requirement>
</play>
```

This example would mean that for each use, the user must pay the server twenty dollars in order to access the resource in question. For our purposes, the requirement for a resource would be that the user is from a specified list of countries, and the certificate has not yet expired.

Parties include all users, including end users and the rights holders. End users are usually those who will be receiving the content. The rights holders are often those who created or produced the content, although that doesn’t have to be the case. [ODRL spec 7] For our purposes, however, the rights holders would be the servers. The end users would be whoever wants to access the page.

ODRL supports revoking offers, agreements, and other rights expressions. [ODRL spec 24]. For example,

```
<rights>
  <revoke>
    <context>
      <uid>…</uid>
      <date>…</date>
      <remark>Error in agreement</remark>
    </context>
  </revoke>
</rights>
```
This rights element could be used to revoke an agreement where the uid (unique identifier) matches the uid of some agreement tag. This method of revoking agreements could be used in our sovereignty problem. This mechanism could be used to have the governments tell the web sites about what rights of the citizens has changed over a period of time. Having a list of added and invoked rights, included with the time of addition and subtraction, would allow for an easy incremental rights update mechanism.

Inheritance is another important feature of ODRL relevant for our solution: “The ODRL expression language also supports the ability to specify the inheritance of expressions between assets. That is, to allow parent/child relationships to be defined...The child asset element can include the ‘inherit’ element which will indicate from which other parent asset(s) to inherit the rights from. All the parent asset rights will be merged with the child asset rights.” [ODRL spec 33] This inheritance is ideal for specifying various subgroups of a country. For example, a nation with a sovereignty that is divided first into states and then further into cities could have individual rights lists for the nation, the states, and the respective cities. This inheritance model would allow for easy specification of that user’s rights, first by defining the rights of the nation member, then for the state member, and then finally for the cities. Adding a new city simply involves specifying the parent state, and the rules that differ from the parent state and the city.

ODRL already supports security through the use of encryption. “ODRL supports both secure rights expressions with digital signatures and specifying the encryption of assets. The security model uses a profile of the W3C XML Signature and W3C XML
Encryption specifications to support enveloped signing of the entire rights expression (when using an XML binding) and including encryption information about assets.” [ODRL spec 26]

Having looked at two different languages, ODRL and XrML, it was decided that a system that corresponds to a subset of ODRL best implements the features that we require, namely: assets, right, parties, and permissions. The actual labeling of the content of the web page to be viewed by the user is stored in a separate XML file. This document simply contains a root element, and then all of the actions that viewing this file entails. An example would be for a page that contains both Nazi propaganda and pornography:

```xml
<root>
  <nazi_propaganda/>
  <pornography/>
</root>
```

In order to view the page, one must have the permissions for both the actions of nazi_propaganda and pornography. Admittedly, this is quite a disturbing web page.

**IMPLEMENTATION ISSUES AND FLAWS**

In the development of the proposed solution, it became evident that there is no simple solution. For every problem addressed, a new one seemed to arise. The conflict of different physical sovereignties in a currently sovereign-less Internet is an inherently difficult problem. When dealing with regulation and censorship, one is unlikely to polar opposites equivalently. The government of the USA (theoretically) would want to uphold the freedoms guaranteed by the First Amendment at all costs, while the government of a regime like China would want to regulate any and all cyber-speech
critical of the government. When walking on the middle ground, one is just as likely to please no one at all.

Given a certain social unwillingness to start regulating content on the Internet, there are also technical flaws related to the proposed solution. First of all, this solution assumes a certain level of public key infrastructure (PKI). For the system to work, the governments of all sovereign states on the Internet would need to have their own private/public key pair and all of these public keys would need to be freely available to all servers on the Internet. In practice the number of keys should be manageable, from a few hundred to a few thousand, but the infrastructure needs to exist in the first place. The solution was crafted in a way to prevent the necessity of every single machine on the Internet from having its own PPK pair. The management of such a large number of keys would be much less feasible, thus lending more credibility to the proposed solution.

While it is not a large presumptuous to believe that governments would be trustworthy in the proposed architecture, there is a security hole that cunning end users could exploit. The verifying that a user has the correct certificate is based on IP address. Technically savvy users could trick the system first by obtaining both another user’s certificate and IP address (likely from a friend in another country or some freedom-activist bent on circumventing the system). If the malicious user “spoofs” his/her IP address (directly or by routing through the cohort’s machine) to be that of the one contained in the certificate, then the user would obtain access to all rights granted with that certificate. It is hoped that this type of attack would fail during the challenge/response part of the protocol. When the random string challenge is delivered to the “spoofed” IP address, the imposter would have to “sniff” out those packets and return
them in a “spoofed” manner again [Source Address]. In theory this is doable, but both the collaboration and technical know-how necessary to circumvent the system would place steep limitations on such devious behavior.

An obvious issue that must be acknowledged is the sheer scale and coordination required for the proposed infrastructure to be accepted. Many independent governments would need to simultaneously start issuing, or regulating the issuance of, digital certificates in such a way that the certificates are all similar enough such that web servers can easily obtain the information contained in any certificate, regardless of physical place of origin. This would take a grand level of international cooperation and coordination.

**WHY COMPLY?**

This paper has discussed the infrastructure and protocol of the proposed system in detail. Any remaining feasibility problems aside, there is one issue that still needs to be addressed: with the responsibility for blocking resting solely on the servers, what is their incentive to comply with the protocol if in doing so they are limiting their own user base? And how is a level of trust to be maintained that a server is blocking potential web-traffic appropriately? There are several responses, all of which deserve explication.

First there are those providers that would willingly want to comply. Lessig and Resnick argue: “if a jurisdiction database or a credential-rich Internet were in place, we might expect voluntary uses of that infrastructure to proliferate….For example, recording companies might refuse access to their web sites from countries where pirated copies of intellectual property were rampant” [Res 423]. The proposed infrastructure provides a way for servers to deny access to citizens of a specific government that it has a vested interest in denying, for one reason or another.
How to persuade regulable websites at large to comply with the standard? In one method governments would create a situation in which servers voluntarily complied. For example, the government might place a country-wide firewall that blocks an entire domain or any packets originating from a non-trusted IP address. Presumably, every single site/page on the server does not merit blocking by the government. If the server were to implement the system described in a way that the government finds acceptable, then the ban on that server would be lifted, and many sites and pages would get through to that jurisdiction. To use the example introduced by this paper, Germany might block all of yahoo.com until Yahoo blocked German citizens from accessing information related to pro-Nazi propaganda. Under the proposed infrastructure, Yahoo could no longer claim an inability to comply, rather the company could satisfy Germany’s request within hours. The threat of total blockage is a power that sovereignties have to coerce web servers into a trustworthy compliance.

Finally, and most radically, once the architecture is reliably provided, countries could require servers within its own borders to comply with the blocking protocol. Not only is this a method for guaranteeing that citizens don’t access illegal material within the country’s own borders, but also from servers in other countries. In one scenario, “States would enter a compact whereby they, as home jurisdictions, agree to require senders…within their own jurisdictions to respect the rules of other jurisdictions, in exchange for senders…in other jurisdictions reciprocating” [Res 423]. The creation of such a network of trust between countries would be the final step in molding the Internet into a regulable space in which the content laws of independent sovereignties are respected.
CONCLUSION

This paper argued that under the current architecture of the Internet, there is no effective way to regulate a user’s access to content based on the laws of their physical jurisdiction. This is not a hopeless situation, as a modification of architecture coupled with government interaction could create an Internet in which regional laws and policies are obeyed. This paper has outlined a technical implementation of such an architecture. The proposal relies on the mass distribution of encrypted, digital certificates that tie a user’s machine to the jurisdiction in which it resides. Servers bear the onus of deciding if a user has the right to access the content contained on the server. The server-side implementation can be effectively coded using a digital rights management language. Given the limited development time of this paper, a more perfect solution could not be formulated. There are acknowledged faults with the system and issues of feasibility. However, the proposed solution provides a reasonable starting point for a further inquiry into the problem of regulating content access on the Internet on a jurisdictional basis.
Works Cited


http://www.cyfi.com/cyfi/dcfaq.htm


http://www.odrl.net/1.1/ODRL-11.pdf

http://www.w3.org/2000/03/PICS-FAQ/

http://www.w3.org/PICS/


http://www.networkmagazine.com/article/NMG20000517S0167

http://www.xrml.org/reference/TheNeedForARightsLanguage.pdf
http://www.prettys.co.uk/article/articleview/164/1/84/

http://www.beat-uofa-cafe.com/yahoo_summary.htm