DRM as a Layered System

Pramod A. Jamkhedkar[†] Gregory L. Heileman[‡]

[†]Department of Computer Science
[‡]Department of Electrical & Computer Engineering
University of New Mexico
Albuquerque, NM

Introduction

- The fragmented nature of the DRM industry in 2004 is similar to that of the telecommunications industry in the late 1980's.
- Telecommunications industry in the 1980's:
  - numerous competing networking technologies;
  - limited interoperability.
- The OSI layered framework and the TCP/IP protocol suite were created to address this problem. These provided the scaffolding for building the Internet.
- Can something similar be done for DRM?
  - Does a layered architecture make sense?
  - If so, what is the "glue" that would hold the layers together?

Layered Systems

Partitioning the design of a system into layers, with well-defined interfaces between layers, and specific services provided within each layer.
Basic tenet: protocols and procedures at one layer should be independent of those at other layers.
Advantages:

- *Reduces complexity*–only need to worry about the services provided by a layer, and how to communication with the layer above and the layer below.
- *Promotes interoperability*–with appropriate handshaking protocols, vendors can independently focus on technologies that address different parts of the system.

Layered Systems

The issues involve determining:

1. What services should be provided by each layer, i.e., choosing the boundaries between functions.
2. Where standards should be defined.

For the Internet:

End-to-end arguments have been useful in resolving the first issue, and standards processes have been useful in resolving the second.

End-to-end Arguments

Functions should be implemented at the edge of the network rather than at the core when this is possible.

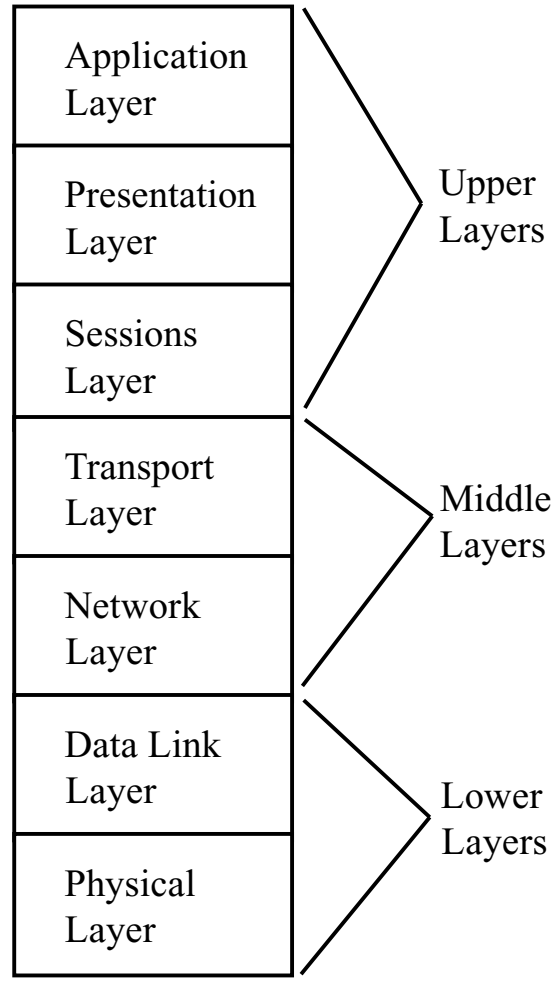
1. Providing too many choices to end users adds to the complexity of configuring a service.
2. Security and trust issues are causing some functionalities to be pushed more towards the core.

Standards Processes

Different stakeholders have differing interests that may conflict with one another – the "tussle". Should not design the architecture so as to dictate the outcome of the tussle:

1. Modularize design along tussle boundaries.
2. Design for choice – the protocols should allow parties involved in an interaction to express preferences. Choices drive innovation, product enhancements, and impose discipline on the marketplace.

The OSI Layers



The OSI Layers

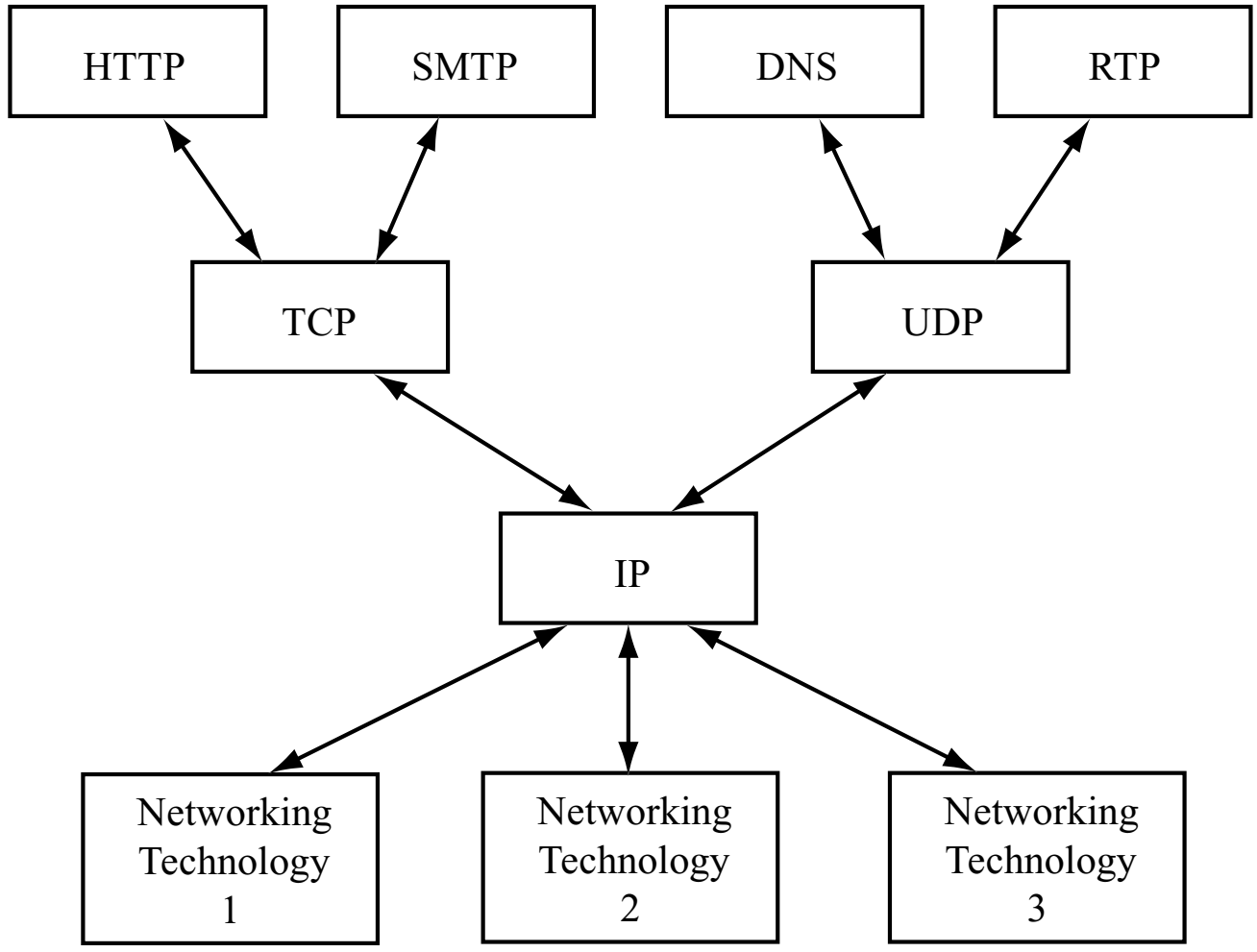
- *Upper Layers*– the interface to the user on the client machine.
- *Lower Layers*–where the physical process of communicating bits between various types of telecommunications equipment takes place.
- *Middle Layers*– largely concerned with providing services, e.g., QoS, best effort delivery, etc.

What is used to link application-level functionality with the low-level transmission of bits? IP

The Role of IP

- IP plays a critical role in making these layers work together.
- It is the minimal service (least common denominator) in the protocol stack.
- Different types of communication equipment know how to break messages up into IP packets, and different delivery services can be built on top of these packets.

The Role of IP



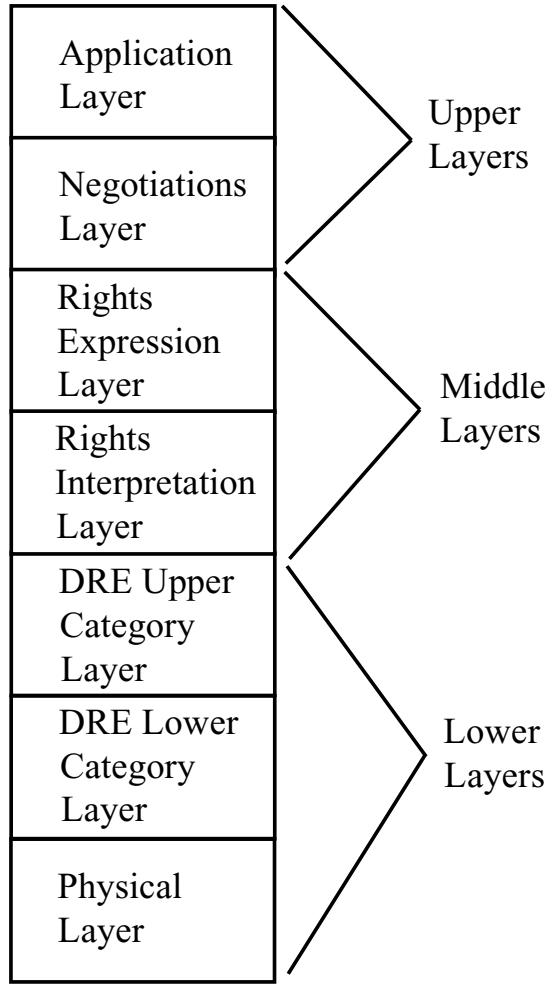
DRM Layers

Similar to the OSI layers, involves a pair of protocol stacks (in this case one for the content distributor and one for the content consumer):

- *Upper Layers*–creates the high-level services that are frequently used by applications that involve DRM.
- *Lower Layers*–concerned with the enforcement of rights.
- *Middle Layers*–provide the minimal set of DRM services necessary to "connect" the upper and lower layers.

What technology can be used to link DRM application-level functionality with low-level rights enforcement technology?

DRM Layers


Application Layer
Negotiations Layer
Rights Expression Layer
Rights Interpretation Layer
DRE Upper Category Layer
DRE Lower Category Layer
Physical Layer
Upper Layers
Middle Layers
Lower Layers

DRM Layers – Upper

- *Application Layer*–use of content in a DRM setting, rendering of content.
- *Negotiations Layer*–responsible for communicating with client-side layers in order to determine the level of rights enforcement available:
  - Influences the type/value of content delivered by the content distributor.
  - Would likely involve certification of client-side technologies.

DRM Layers – Lower

- *Physical Layer*– memory, file systems.
- *DRE Lower Category Layer*– device drivers, secure containers.
- *DRE Upper Category Layer*– encryption, watermarking, digital certificates.

The Rights Expression Layer

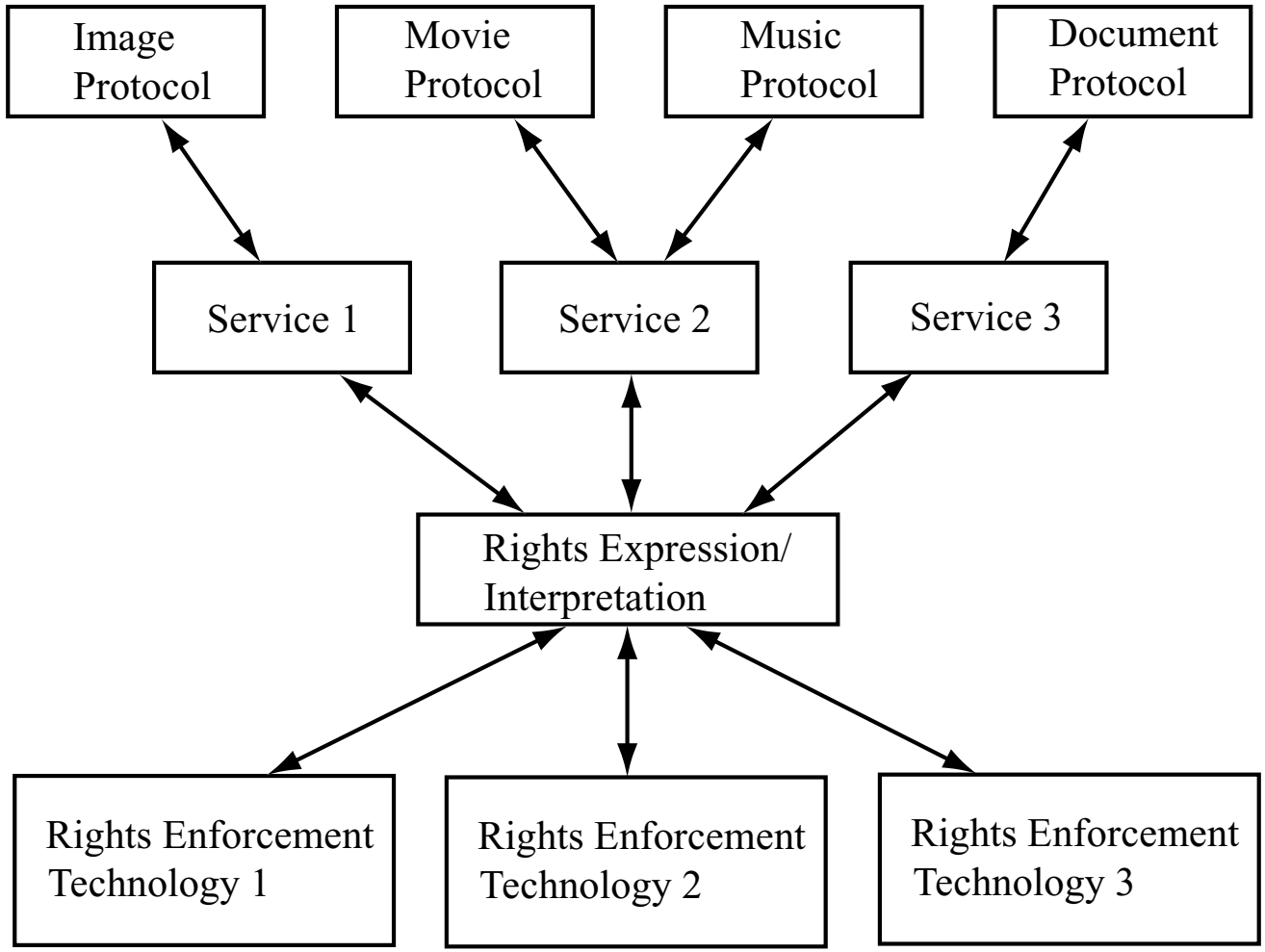
- Collects information from upper layers regarding the rights that need to be supported in order to provide DRM functionality to a particular application.
- Shields the service layers above from the intricacies of rights enforcement layers below.
- The services provided by this layer are the minimal ones required by any real DRM application. I.e., there must be rights in a DRM application, and this is the layer responsible for expressing them.
- Rights Expression Languages (RELs) appear to be the preferred implementation approach.

The Rights Interpretation Layer

Different rights can have different interpretations depending upon the environment, e.g.,

- A “read-only” right has one interpretation in a general-purpose computing environment (where printing or copying is readily supported), and another in a special-purpose environment (where they are not).

Rights Expression/Interpretation Role



Conclusions

- Adoption of a layered system framework may facilitate the development of the DRM industry:
  - allows for separate and independent development of DRM technologies by vendors,
  - specific DRM functionalities can then be created by piecing together these technologies.
- Success will require the systematic development of standards and protocols within the various layers.
- A standard REL appears to be a key piece for DRM architectures, as it provides the minimal service required in a DRM application.