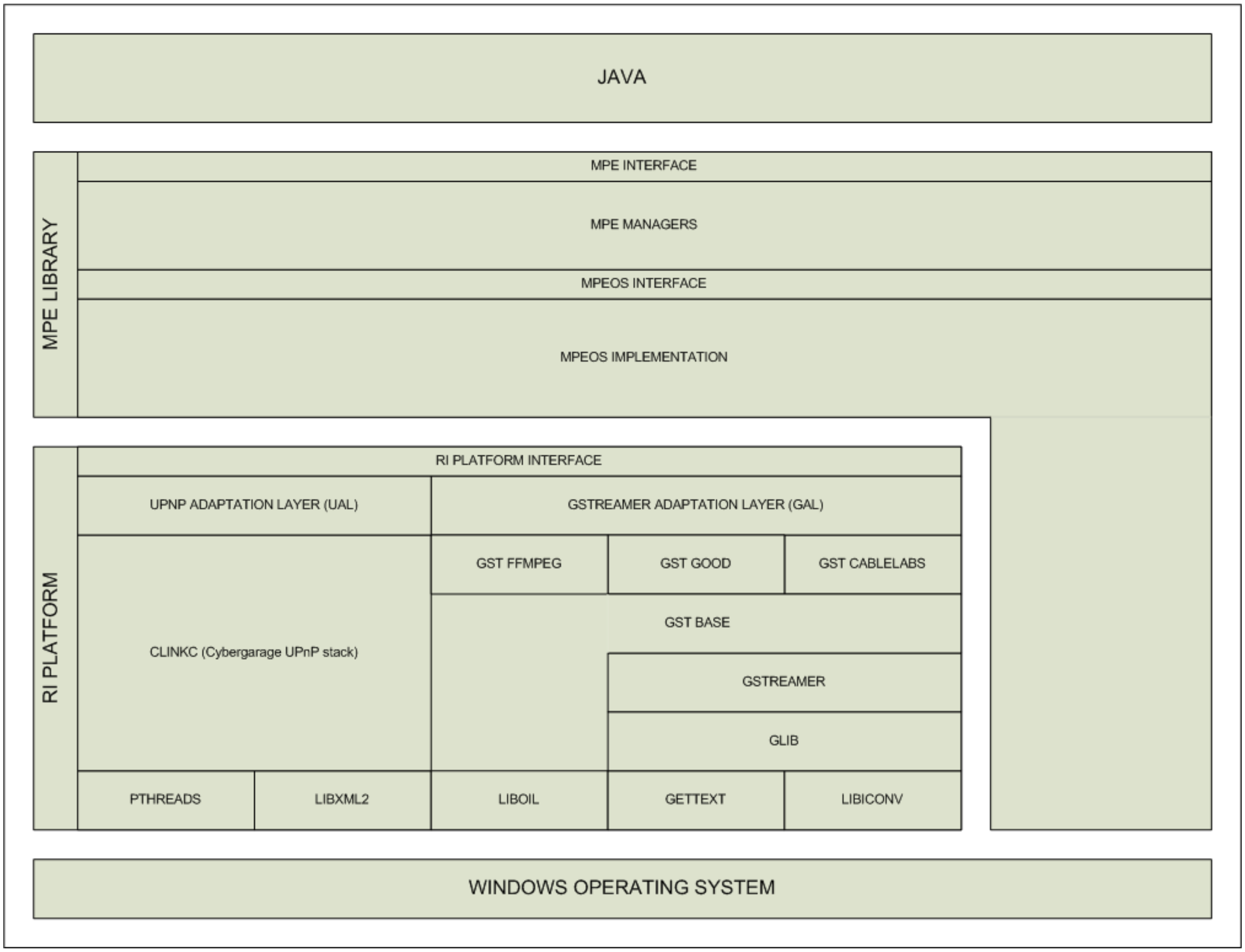
RI PC Platform Overview

*****NOTE: This is an early design document. Some (many) changes have occurred since it was written.*****

The following diagram shows how the RI PC Platform component fits into the overall Reference Implementation system:



The RI Platform module is a mix and match of various external libraries, available under favorable licensing terms that we can incorporate into the CableLabs source code base as well as some CableLabs developed code. Its main purpose is to provide a multimedia framework that will allow the RI OCAP stack through the RI MPEOS implementation to use and excercise all of the multimedia features. That includes emulation of tuning, demultiplexing, decoding and displaying decoded content.

At this time, the RI Platform is split into a number of sub-components. They are:

- **Support Libraries**
- **Gnome Library**
- **UPnP Stack**
- **GStreamer Framework**
- **UPnP Adaptation Layer**
- **GStreamer Adaptation Layer**

Support Libraries

The support libraries, which are located here are not required directly by the stack or any of the CableLabs produced code. Instead, they satisfy dependencies of the required GStreamer and UPnP components. At this time, the support libraries include:

1. A PThreads library implementation for Windows. This package is required by the UPnP stack. It is a Windows-only component and would not be required for Linux. More information available here.
2. An XML library implementation. This is also required by the UPnP stack. It might also be used at some point by the GStreamer library. More information available here.
3. OIL library. It is required by the GStreamer Base Plugins library and provides optimized versions of computation-intensive functionality for various CPUs. More information available here.

4. Gettext library. Provides support for internationalization and allows producing programs supporting multiple natural languages. Required by the Gnome Library (Glib). Windows-only package. More information available here.
5. iConv library. Provides support for conversion between different character encodings. It is required by Gettext/Glib. Also a Windows-only package. More information available here.

Gnome Library

The Gnome Library, or Glib, is required by GStreamer framework. It also contains a useful collection of utility functions such us queues, stacks, hashtables, etc. that the RI platform implementation will leverage when we produce the CableLabs code for UPnP and GStreamer layers. A reference manual is available here. Repository location here.

UPnP Stack

The UPnP stack is part of a larger HE4RI client component. It is used as a communication provider for control mechanisms with the HE4RI server component over a network. The RI Platform uses the "pure" C version of the Cybergarage UPnP stack implementation. More information available here. Repository location here.

GStreamer Framework

A separate page that describes GStreamer has been created here. In addition to the external libraries, a significant effort will be contributed towards development of GStreamer plugins, that are required by the RI Platform and do not exist at this time in any of the publicly available plugin collections. Examples of those include section filtering elements, DVR indexing elemenets for record/timeshift, demux extensions, decoder extensions, etc. Repository of CableLabs plugins is located here.

UPnP Adaptation Layer (UAL)

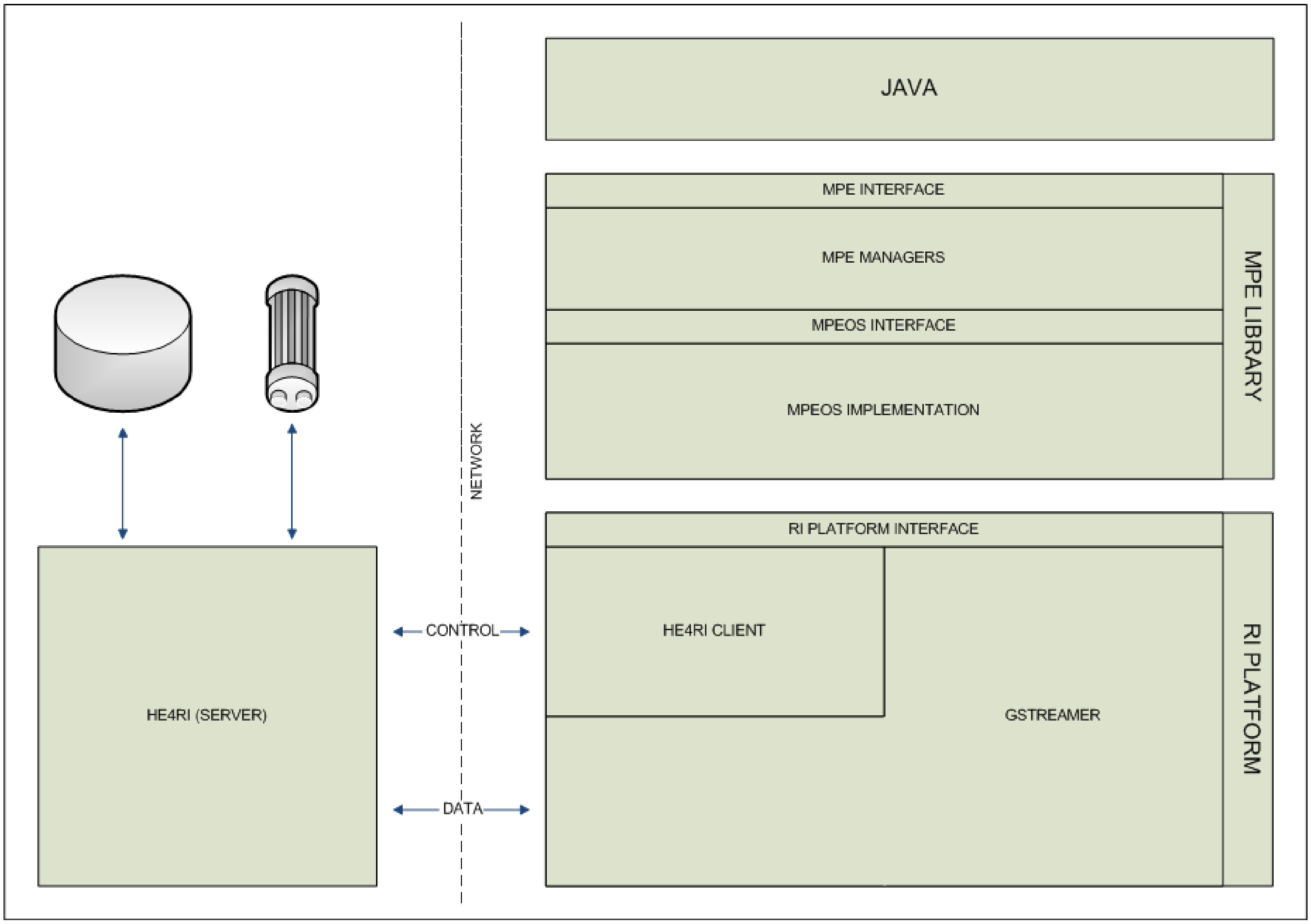
A separate page that describes this layer has been created here.

GStreamer Adaptation Layer (GAL)

This layer will be essentially a GStreamer multimedia application. The application will have a public API, exposed via the Platform Interface to allow for manipulation and control of its elements, bins and pipelines.

Functional description

Another view of RI Platform architecture would be the functional view.



At this time, RI can be split into three distinct subcomponents:

- GStreamer
- HE4RI (Head-End For RI) Client
- Platform Interface

As development moves forward, it is expected that at least two other major componets will be added:

- Graphical User Interface
- Storage Facility (for stored content - needed by DVR and HN extensions)

GStreamer

The GStreamer component is expected to supply the following functionality:

- Supply the hardware emulation of all major components of a set-top box decoding and presentation pipeline.
- Simplify loading, management and instantiation of GStreamer components.
- Simplify runtime manipulation and configuration of GStreamer components.
- Specialize the use of GStreamer elements.

HE4RI Client

Don, Steve - can you put something here?

Platform Interface

The main goals of this interface would be:

- Keep it simple.
- Remove dependencies on any technology specific terminology (GObject model, UPnP/XML).
- Hide as much as possible any technology specific particularities - try to model the porting layer/MPEOS concepts as closely as possible:
  - Do not expose GStreamer-related concepts (element, bin, set property, etc.) - use set-top box concepts instead (tuner, demux, decoder, display, set frequency).
  - Do not expose UPnP-related concepts (RPC, XML) - use set-top box concepts instead (tuner control, POD control, OOB section filtering).
  - Limit GUI implementation exposure to minimum.
- Support multiple instantiation of each exposed object/API.

- Design APIs to be reentrant and thread-safe.
    - If possible, do not design APIs that would disallow use (or would make it hard to use) by multiple clients.

Development Environment

The RI PC Platform code will be developed in C.

It will utlize GLIB which a Software Utility Platform without GUI components which makes interfaces to the Operating System platform independent.

The build process will utilize Make. It will be targeted to run on Windows XP and Linux. Building the RI PC Platform in a DOS Window or from Eclipse is not supported but should work.

The use of IDEs is at the discretion of individual developers.