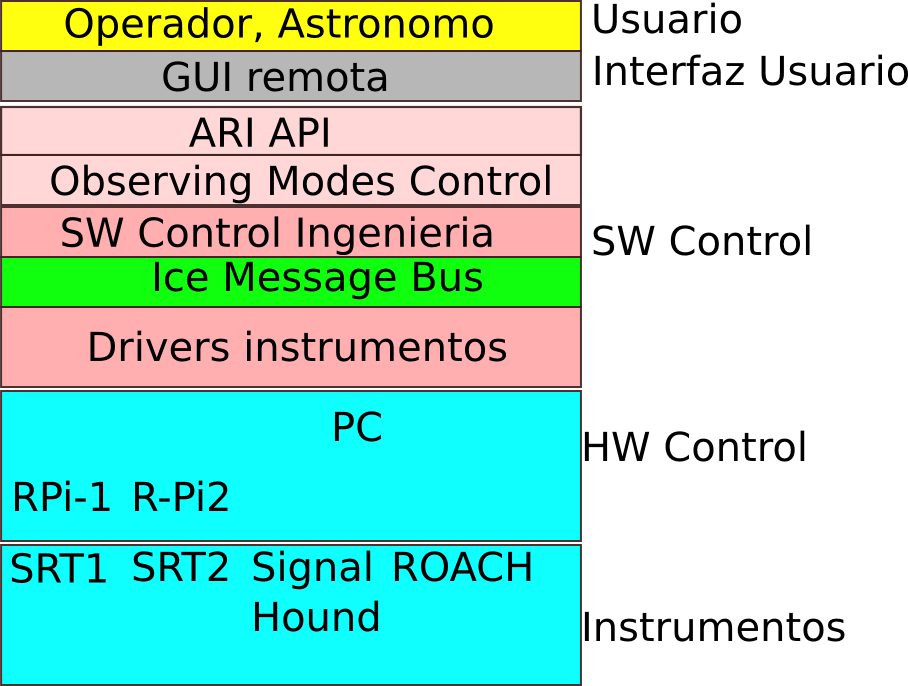
**Estructura Software control ARI**

El siguiente diagrama muestra las distintas capas que conforman el sistema de control de ARI.

La principal funcion de la API de ARI es la de proporcionar funcionalidades de alto nivel para la interfaz de usuario que permiten ejecutar acciones de control sobre los observing modes, de modo de poder ejecutar actividades de observacion cientifica con el telescopio ARI.



**Manual ARI API**

**Para hacer uso de la API es necesario instalar Ice de ZeroC y descargar desde el proyecto acs-ari en google code :** [**https://code.google.com/p/acs-ari/source/browse/**](https://code.google.com/p/acs-ari/source/browse/) **el folder API desde trunk/ARI/TCS/rpiSRT/src**

**Acceso y uso de la API**

En computador Remoto:

3.- Copiar carpeta Observing modes, verificar que Ice y sus dependencias estan instaladas.

4.- En Python importar la API de ARI

from ARI\_API import \*

5.- Hacer una instancia de la API

g = API()

6.- Connectarse a los servicios de la API

g.connect(‘192.168.3.100 –p 10015’)

Notar que esta direccion IP es la q tiene localmente el PC de control SRT, la forma en como se ruteara en forma remota desde fuera de Santa Martina esta fuera del alcance de este manual.

En caso de presentarse falla de conexion, ejecute compilacion de los archivos \*.ice utilizando slice2py, i.e.: slice2py GUI.ice

Notar que son 2 archivos \*.ice en el directorio.

**Metodos disponibles en la API**

Hasta este paso se ha establecido las conexiones logicas y de datos para conectar un codigo remoto a los metodos ofrecidos por el servidor. A continuacion se presentan los metodo expuestos actualmente, esta lista de metodos ira cambiando a medida que el desarrollo del control se vaya completando. Estos primeros metodos tienen por objetivo prototipar la interfaz de control de mas alto nivel. En este momento el servidor esta desconectado de la interfaz de control de la antena hasta que su operacion sea segura, en una etapa posterior la interfaz de control se conectara a esta capa y permitira el acceso al hw del SRT y ARI.

Comandos de Control para una observacion:

Una vez establecida la comunicacion los siguientes metodos se encuentran disponibles, se recomienda se ejecuten en el orden en que aparecen presentados en este manual

* **SetObservingMode(Config, HW)**

Elige modo de Observacion.

Config es la configuracion del telescopio que puede tener los valores:

SD: SRT Single Dish

ARI: Academy Radio Interferometer

HW detalla los equipamientos involucrados en el observing mode y sus valores posibles son:

SRT1 : Antenna SRT1

SRT2: Antenna SRT2

Double: Las dos antenas SRT1 y SRT2

SH : Signal Hound

ROACH: ROACH board

Solo las siguientes combinaciones de Config y HW se encuentran permitidas:

SD, SRT1

SD, SRT2,

SD, Double (Doble SRT single dish mode)

ARI, SH

ARI, ROACH

Cualquier otra combinacion se considera illegal y generara un error.

Ejemplo:

g.SetObservingMode( ‘SD’, ‘Double’)

g.SetObservingMode(‘ARI’, ‘SH’)

* **HWConnect()**

Este comando es necesario para inicializar las comunicaciones hacia los controladores de bajo nivel

Ejemplo:

g.HWConnect()

* **Initialize()**

Este comando configura e inicializa el Hw de SRT o ARI (envia antena a Stow, inicializa encoders, carga parametros, etc).

Ejemplo: g.Initialize()

* **SetTarget(source, frequency, receiver\_mode)**

Este comando configura y prepara el telescopio para una observacion,

**Source** es el nombre de la fuente a observer (en esta version el codigo no verifica la validez del nombre de la fuente o su visibilidad, sin embargo esta funcionalidad sera incorporada a medida que el desarrollo prosiga), por ahora el codigo solo verifica que se reciba un string

**Frequency** es un numero float y define la frecuencia central de observacion

**Receiver\_mode** es un numero que define el modo de funcionamiento del receiver de SRT en configuracion Single Dish, en ARI esta parametro no se utilizara.

Ejemplo:

g.SetTarget( ‘SgrA’, 1420.0, 2)

* **StartTracking()**

Este comando inicia la observacion enviando la antena(s) hacia las coordenadas de la fuente definida por **SetTarget** y luego hara tracking de la fuente

Ejemplo

g.StartTracking()

* **StopTracking()**

Este comando detiene la observacion iniciada con **StartTracking()**

Ejemplo:

g.StopTracking()

A medida que el control de observaciones se vaya completando esta API agregara funciones para el control de alto nivel.

**Ejemplo de session**

In [1]: from ARI\_API import \*

In [2]: g = API()

In [3]: g.connect('192.168.3.100 -p 10015')

Connecting to ARIAPI

connected to client

In [4]: g.SetObservingMode('SD', 'Double')

Setting observing mode

Double Single Dish Mode chosen

In [5]: g.HWConnect()

Connecting

In [6]: Observing Mode Created

In [6]: g.Initialize()

Initializing Telescope

Telescope Initialized

In [7]: g.SetTarget('SgrA', 1420.1, 2)

Setting observation

Receiver and Target Set

In [8]: g.StartTracking()

Starting observation

Tracking Source

In [9]: g.StopTracking()

Stopping observation

Track source stopped