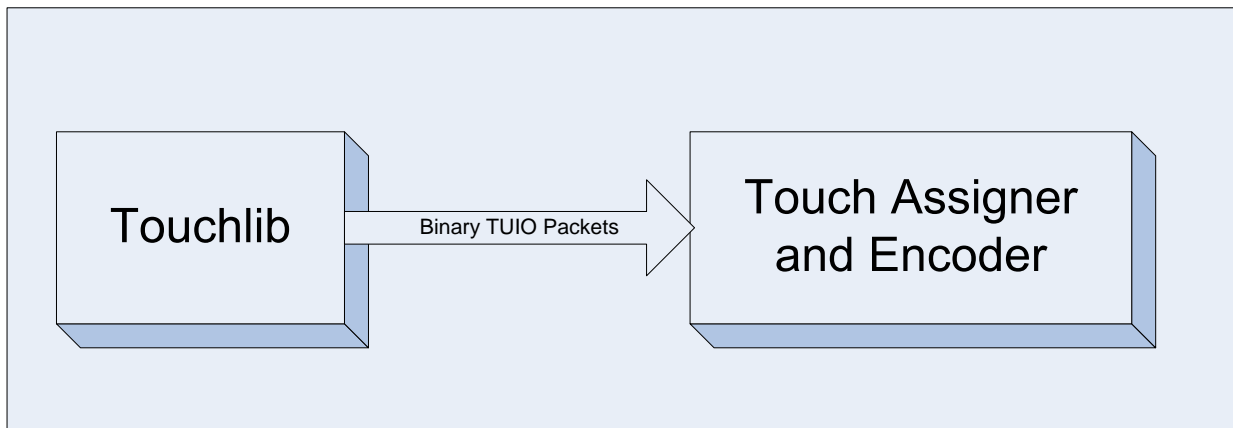


Here, we see the relationship of Touchlib and the Touch Assigner/Encoder.

Currently, Assigner/Encoder will likely be in C++ and link in Touchlib as a library.

This relationship may later be abstracted such that any provider of TUIO events can feed into the Assigner/Encoder via a socket, like FLOSC.

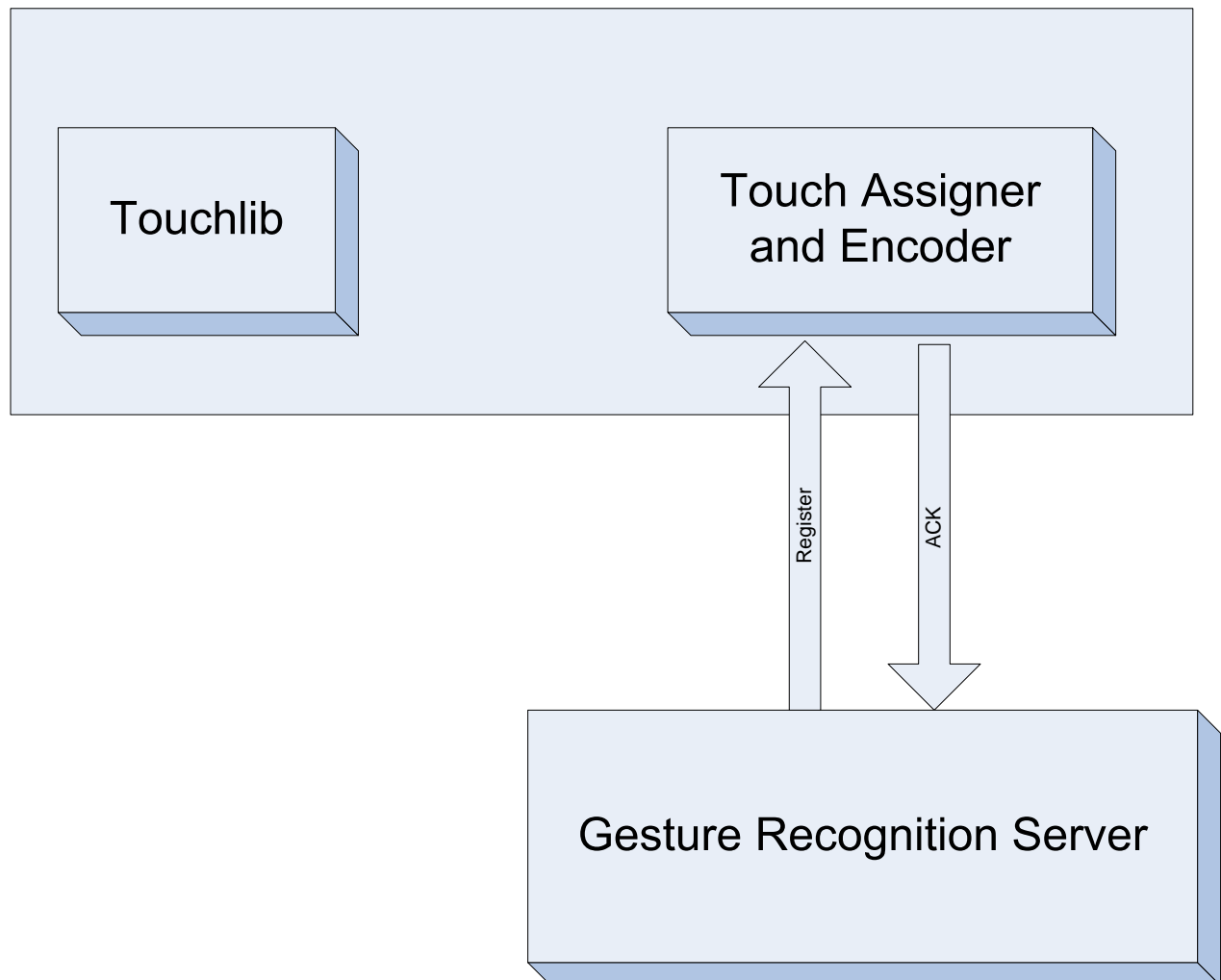


Here, the Gesture Recognizer registers with the Assigner/Encoder.

It says it accepts binary and XML TUIO packets.

The Assigner/Encoder will inform applications of the GRS's presence when they register

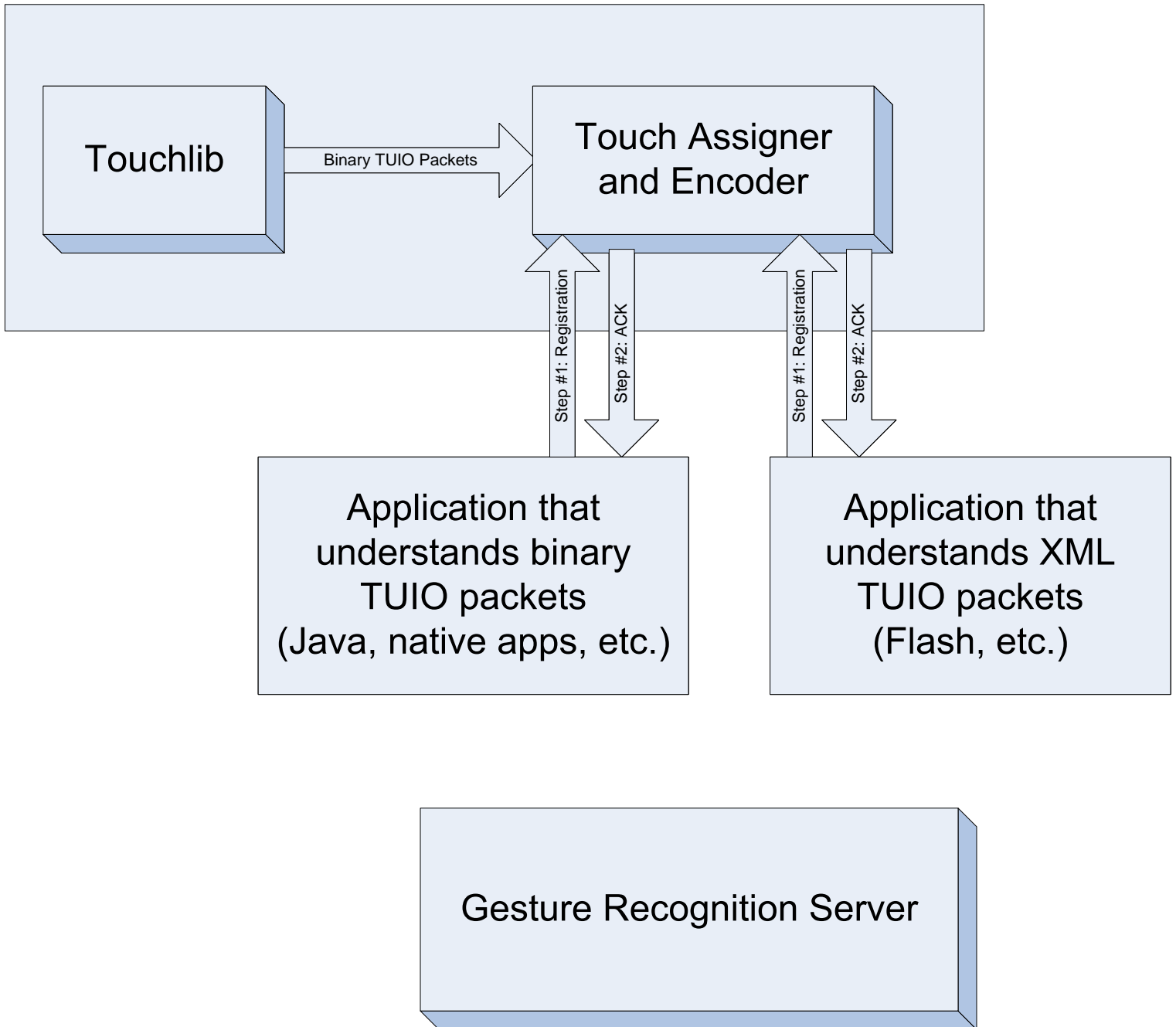
This makes the applications' discovery of the GRS easy.



Here, the applications register with the assigner/encoder, requesting a specific encoding of TUIO packets. (Binary or XML)

The assigner/encoder acknowledges them.

The assigner/encoder will tell them about a Gesture Recognizer if it knows of one.



Here, applications register with the GRS.

This is important, because the GRS needs to remember the last few touch events that are tied to a specific application, so that it may associate them and build gestures.

Also, applications inform the GRS of their preferred TUIO encoding (binary vs XML).

Touchlib

Touch Assigner
and Encoder

Application that
understands binary
TUIO packets
(Java, native apps, etc.)

Application that
understands XML
TUIO packets
(Flash, etc.)

Register (! use binary!)

OK!

Register (! use XML!)

OK!

Gesture Recognition Server

Here, everything is running along.

Assigner/Encoder will assign and encode TUIO events from Touchlib and distribute them to client applications.

The Assigner/Encoder keeps track of the client applications' positions (If it is the window manager, this is easy. Otherwise, they must notify it)

Gesture recognition server accepts TUIO packets, remembers them, and publishes gesture events back to the applications when appropriate.

