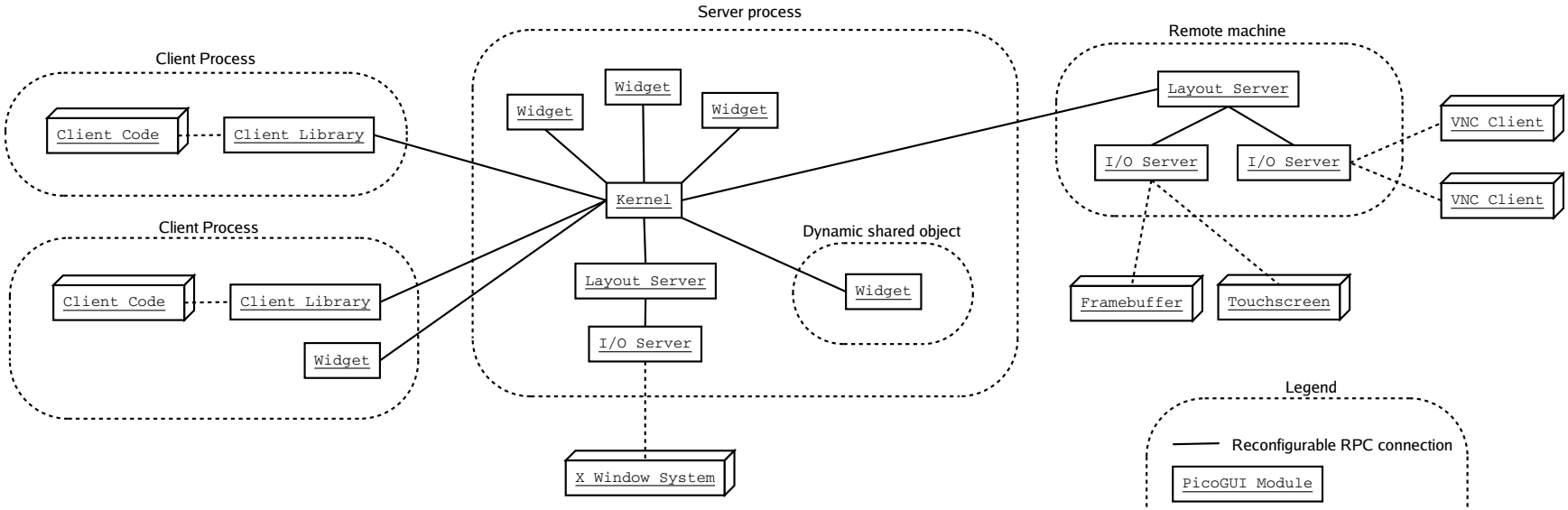

Kernel:

Provides a database of objects, with types, security, and optional network transparency. Any other PicoGUI module can create objects in the kernel, and (according to the permissions) modify, delete, and observe objects.

Widget:

Monitors a set of properties which may be modified by the client, may also maintain internal state in a kernel object. Provides an abstract representation of itself when requested by the layout server.

Accpets events from the layout server, and when necessary generates events for clients.

Layout Server:

Observes a tree of widget instances in the kernel. Requests abstract information about the widget when necessary, and uses themes and other device-specific information to construct a scene graph. A set of layout primitives are provided that can't be expanded without modifying the layout server, but should be general enough for any application- just like the instruction set of a CPU.

Accepts input events from the I/O server and transforms them from device coordinates to a widget ID and coordinates within that widget.

I/O Server:

Observes the layout server's scene graph, and generates primitives to render when necessary. This contains both the rendering algorithm (S2 engine, libart..) and an interface to the low-level video system in use.

Accepts input events from the low-level event system in use, and forwards them to the layout server.

Reconfigurable RPC:

Provides a connection between PicoGUI Modules that can, at compile time, be configured as a very fast zero-copy procedure call, as a call to a dynamically loaded object, or as a network-transparent call using something similar to PicoGUI's existing protocol.

This also provides the mechanism to easily multiplex components, and rearrange them at runtime.