Ctrl Konnekt

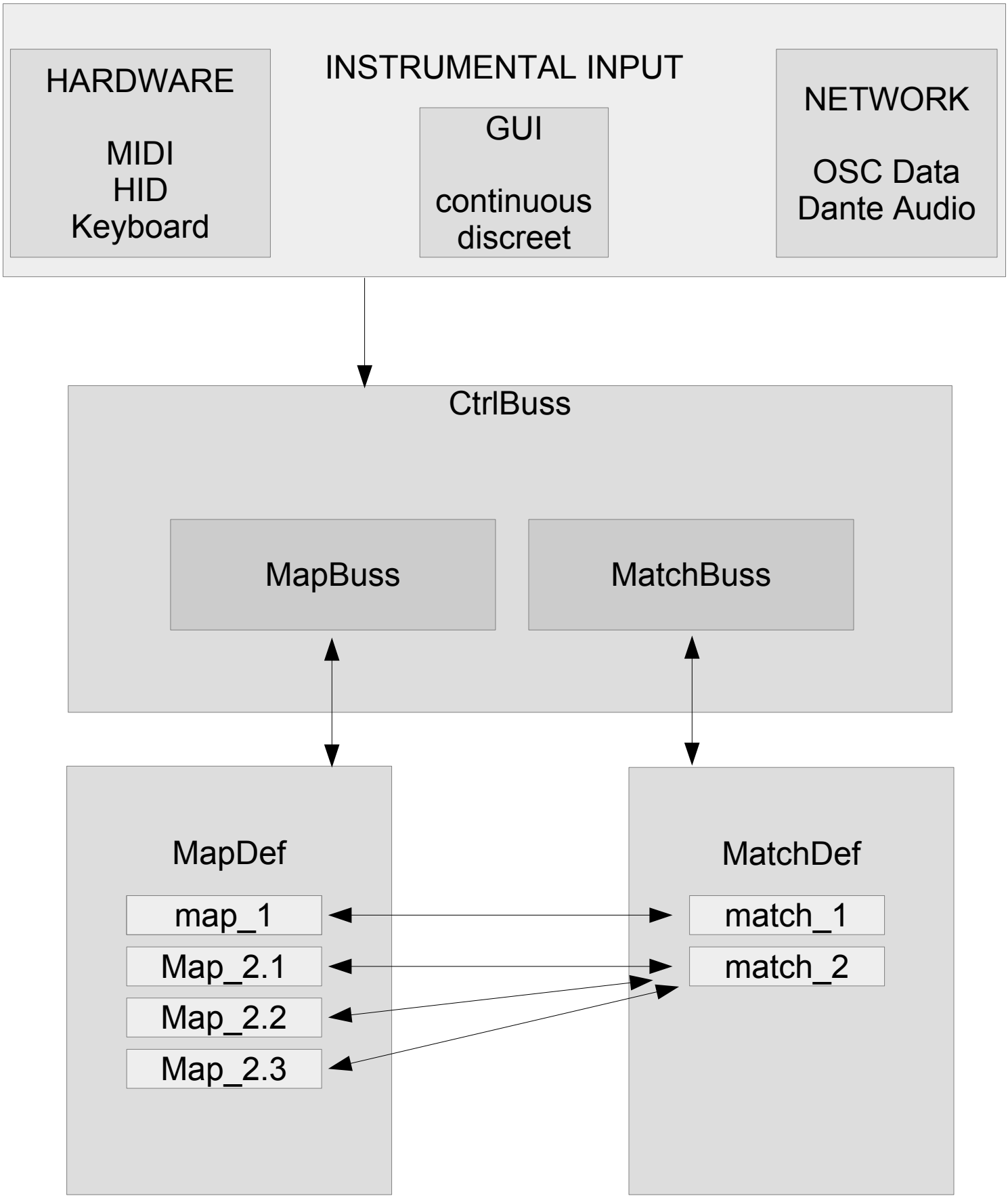
Hardware controllers

Hardware Audio Network

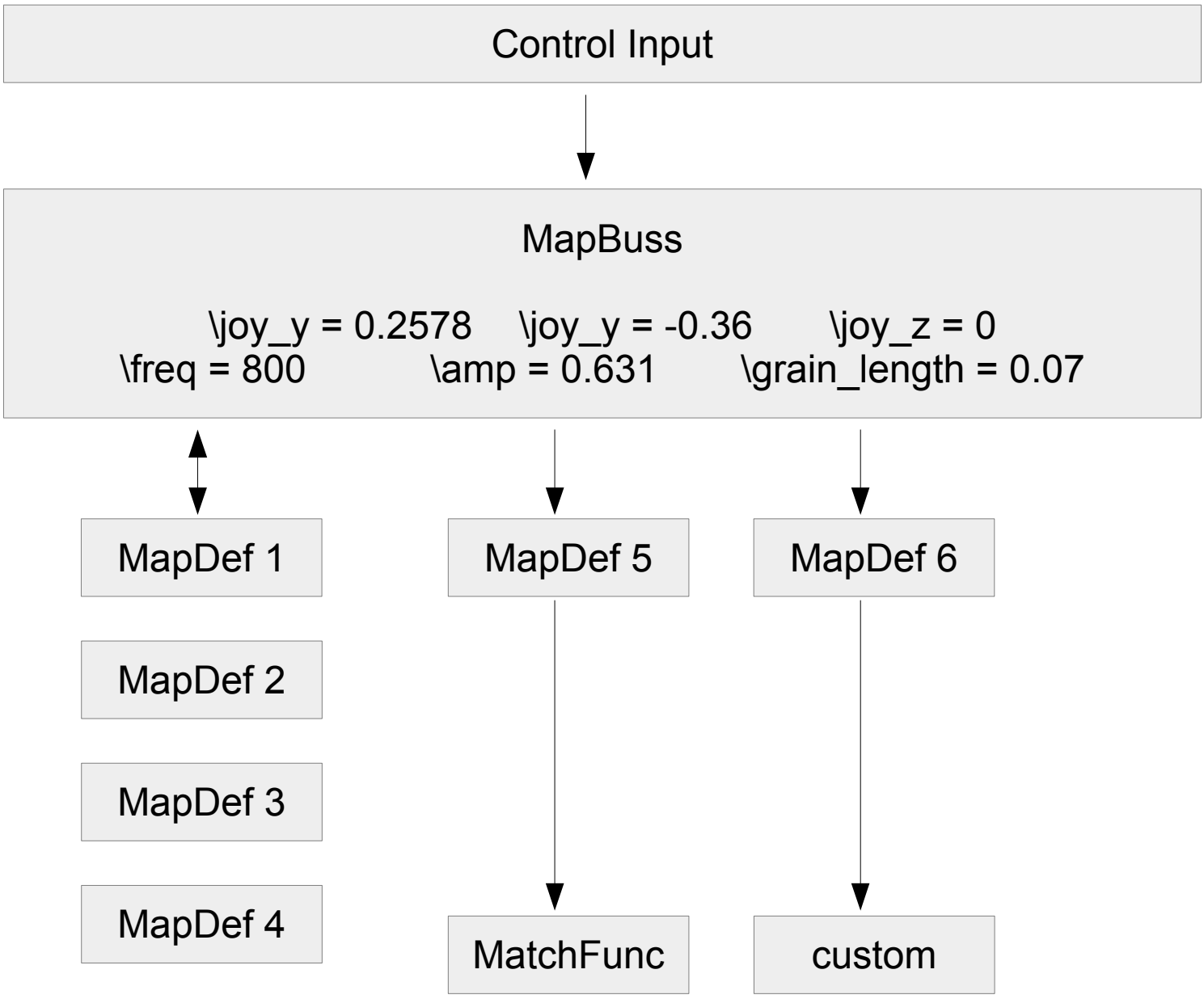
Control Mapping Software

Audio Synthesis

Instrumental design and architecture


INSTRUMENTAL INPUT
HARDWARE
MIDI
HID
Keyboard
GUI
continuous
discreet
NETWORK
OSC Data
Dante Audio
CtrlBuss
MapBuss
MatchBuss
MapDef
map_1
Map_2.1
Map_2.2
Map_2.3
MatchDef
match_1
match_2

MapBuss maintains a list of named values and connects them to mapping transformations and intermediaries. Mappings are invoked when their input value (a named value on the buss) changes. Derived data can be routed to other objects or trigger custom functions when updated.

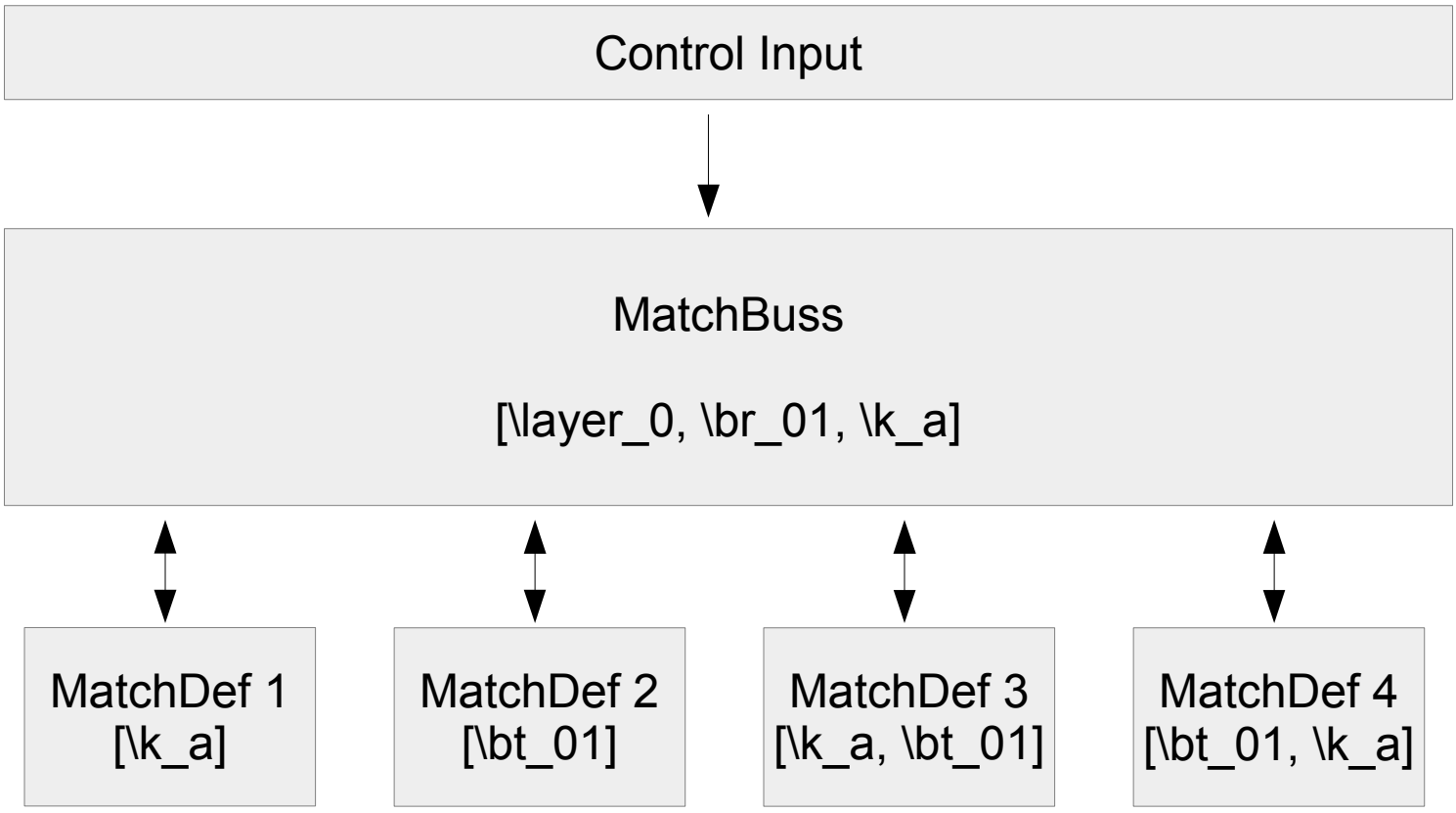


MatchBuss maintains a sequence of logical flags that represent the state of input controllers and functions. As input changes, matches of subsequences start and stop musical functions (MatchDef).

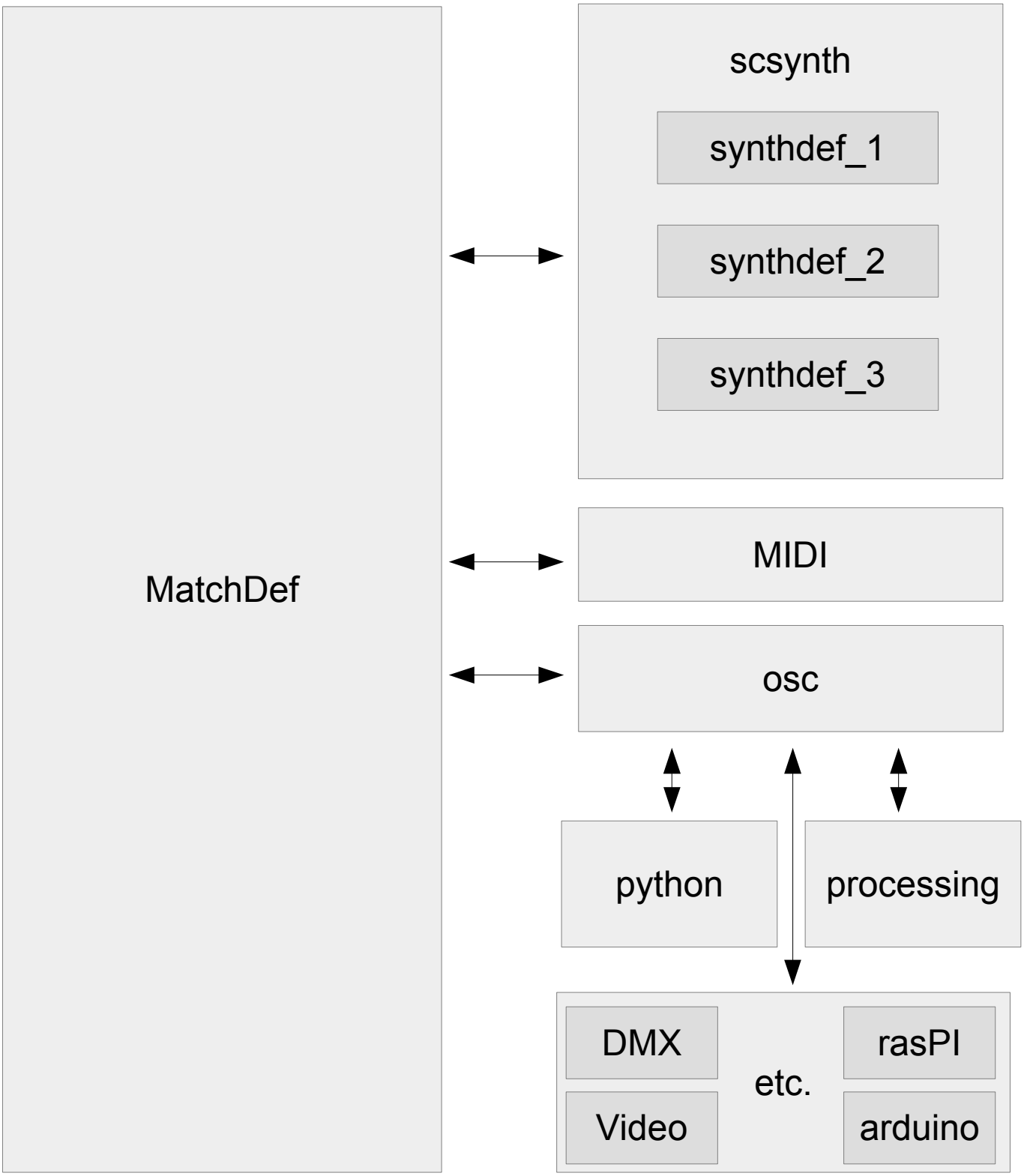
For example, a flag like \bt_01 is added to the sequence when MIDI button one is pressed. Or, \k_a is added to the sequence when the "a" key on the computer keyboard is pressed.

With the two flags \bt_01 and \k_a there are four possible matches: [\k_a], [\bt_01], [\k_a, \bt_01], [\bt_01, \k_a]

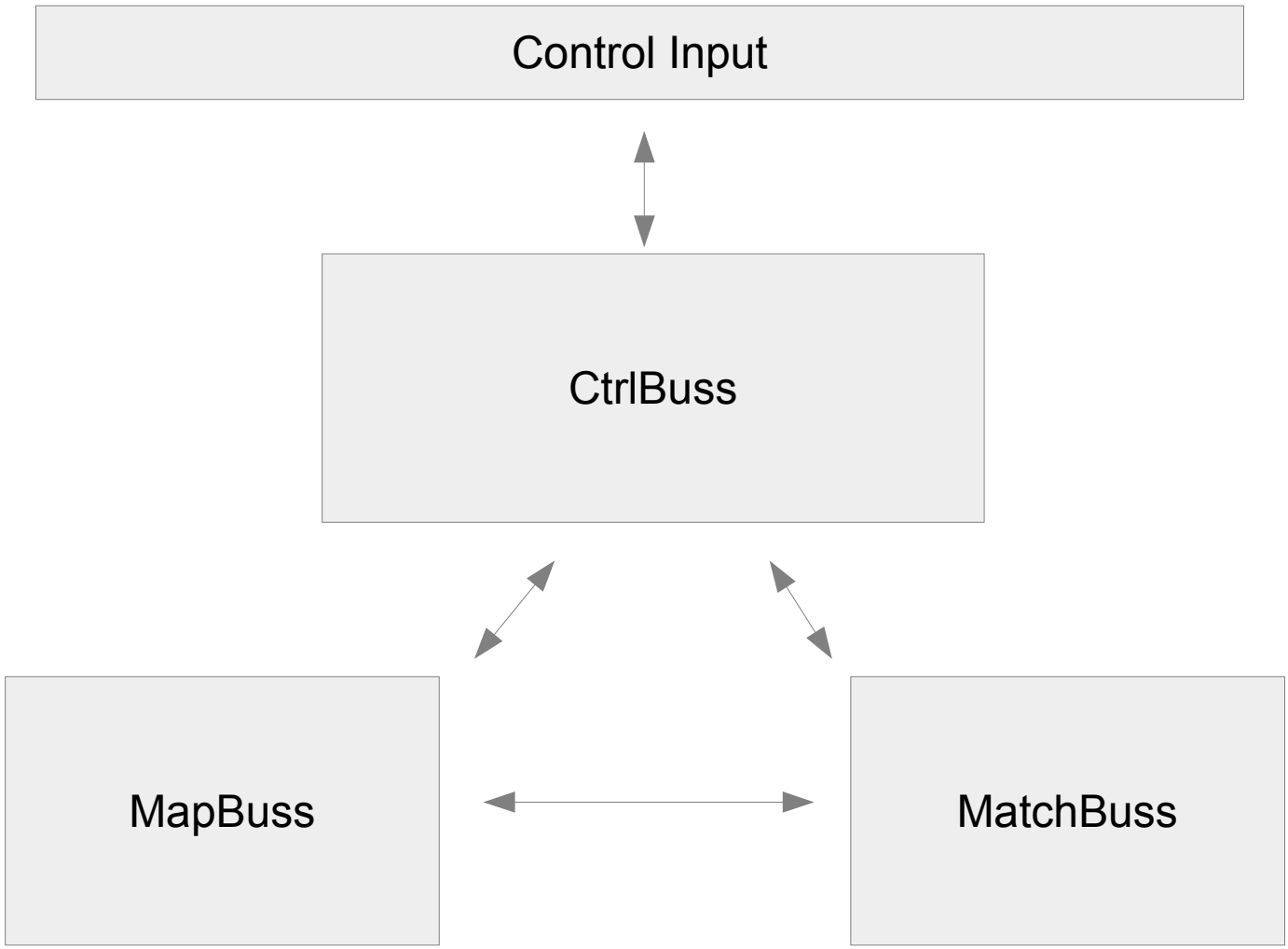
Each match can control different MatchDefs.



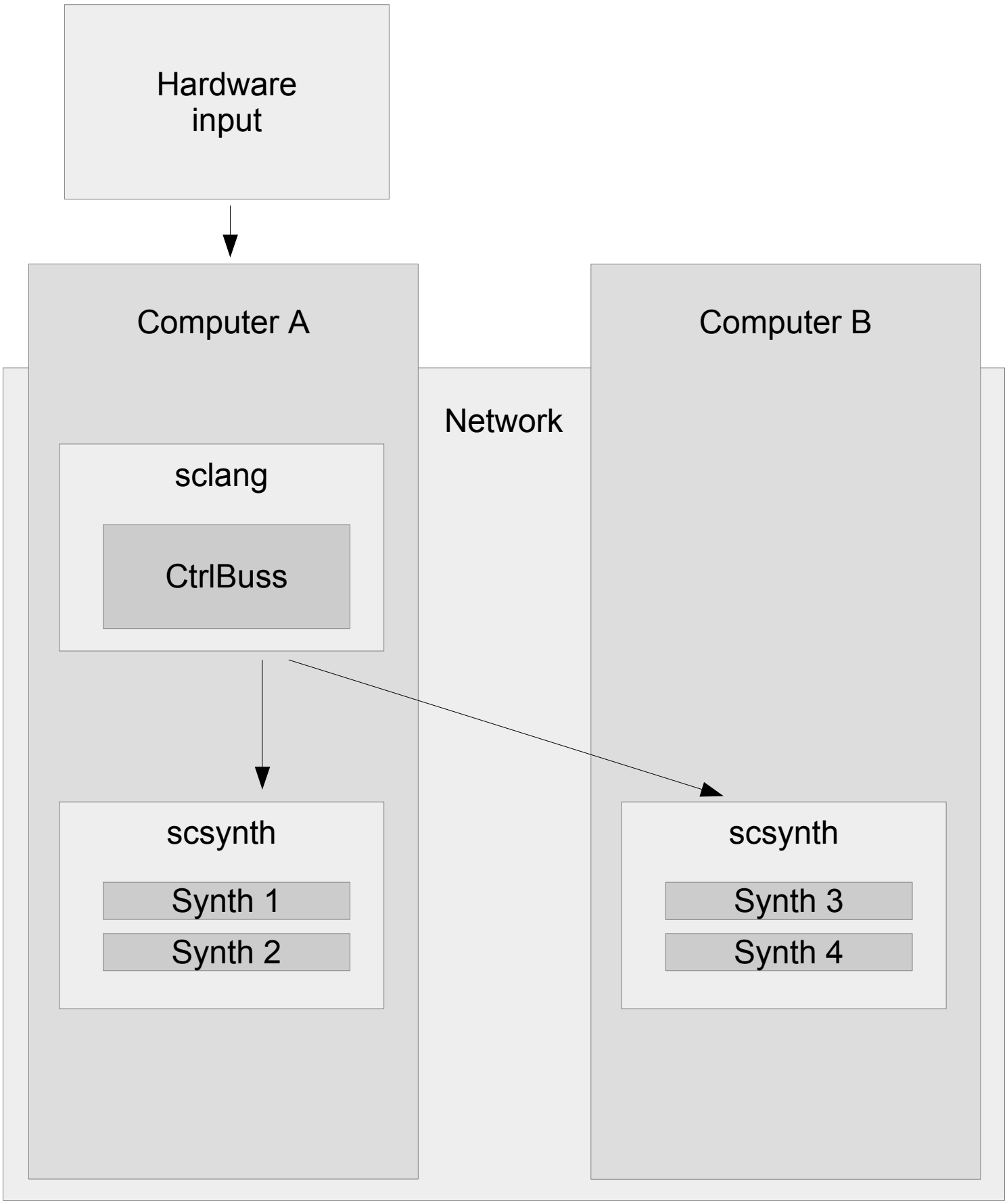
MatchDef is a mapping intermediary that is the basic unit of event control. It is the element that creates the structure for sound object composition and can control any number of synths and tasks. It has state, memory, history, can start/stop and send/receive data streams.



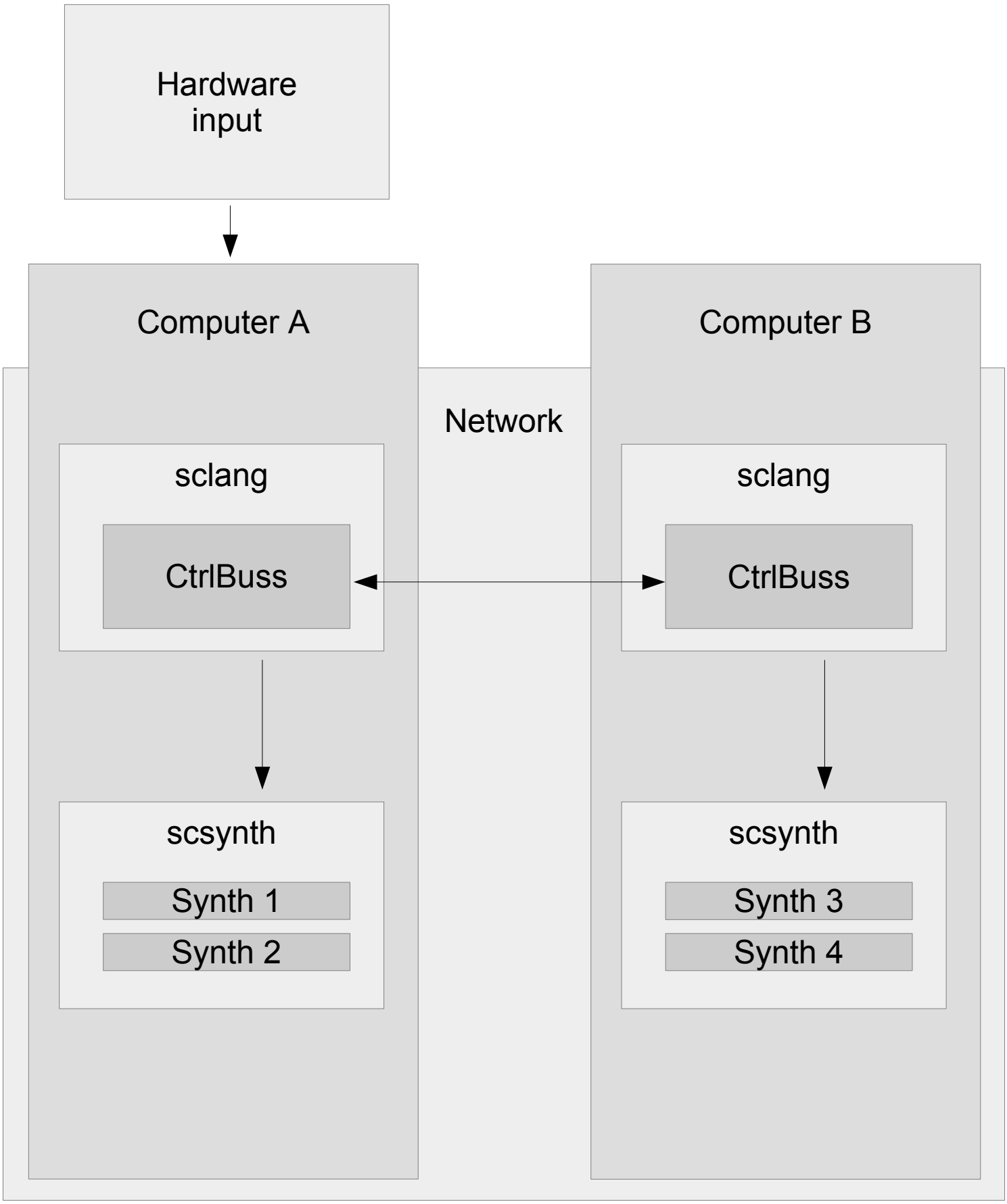
CtrlBuss is used to automatically create links between a MapBuss and a MatchBuss to send conditioned/mapped data (MapDef) for use in musical functions (MatchDef). This basic model allows for controllers to start/stop and modulate any number of musical objects without explicit hard coded connections. CtrlBuss is the proxy for input and it routes data to the MapBuss and MatchBuss.



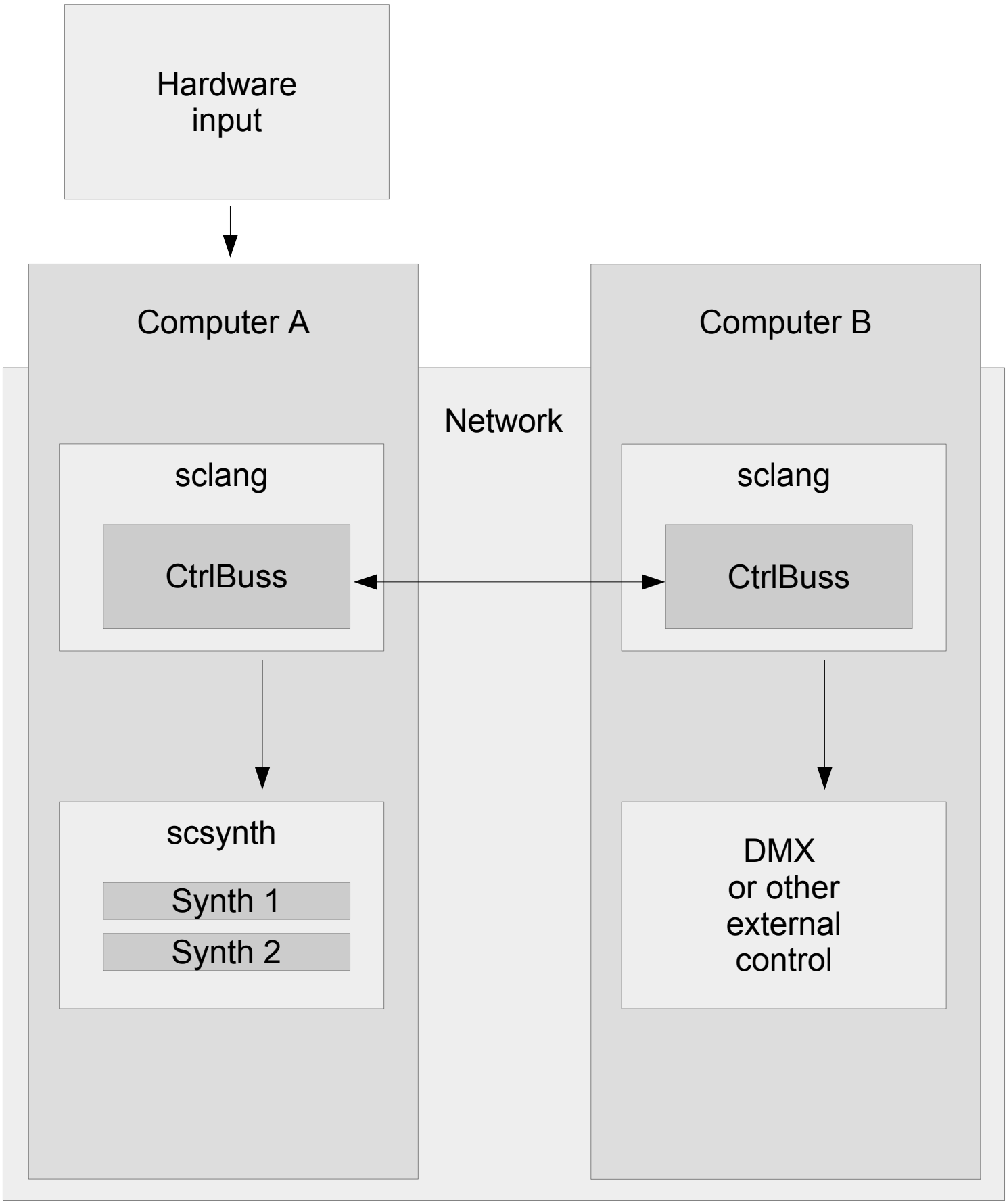
Scalability example 1



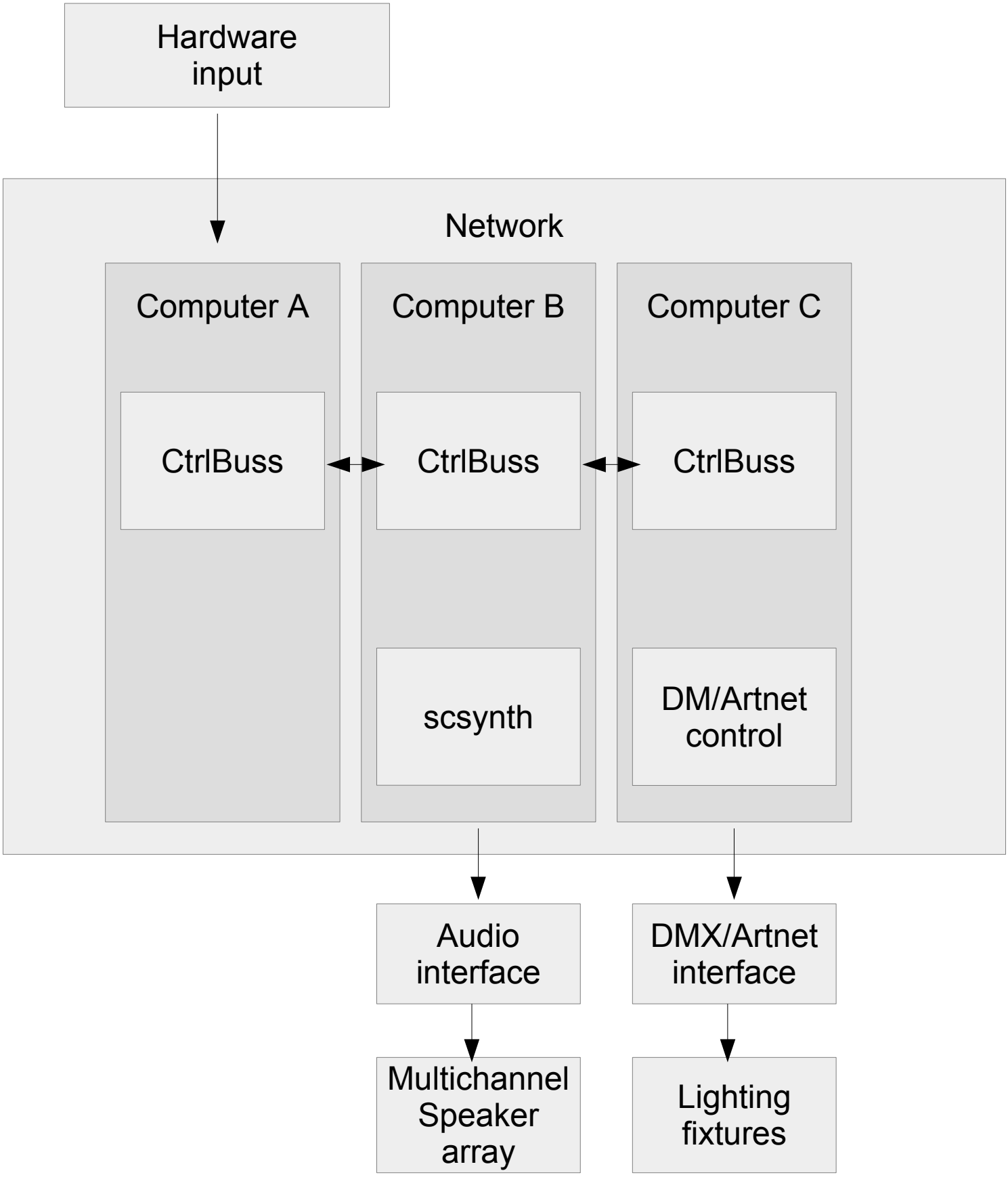
Scalability example 2



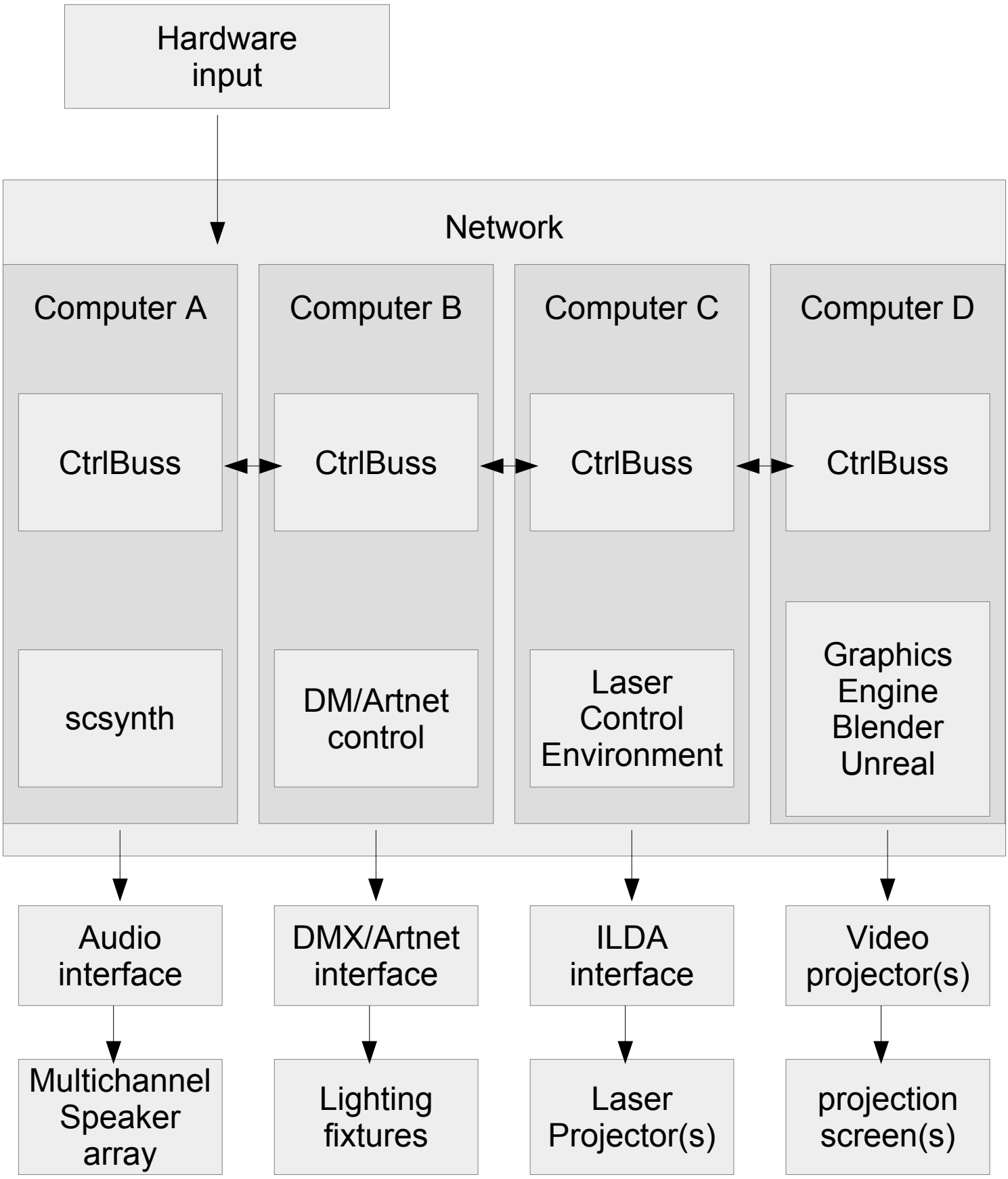
Scalability example 3



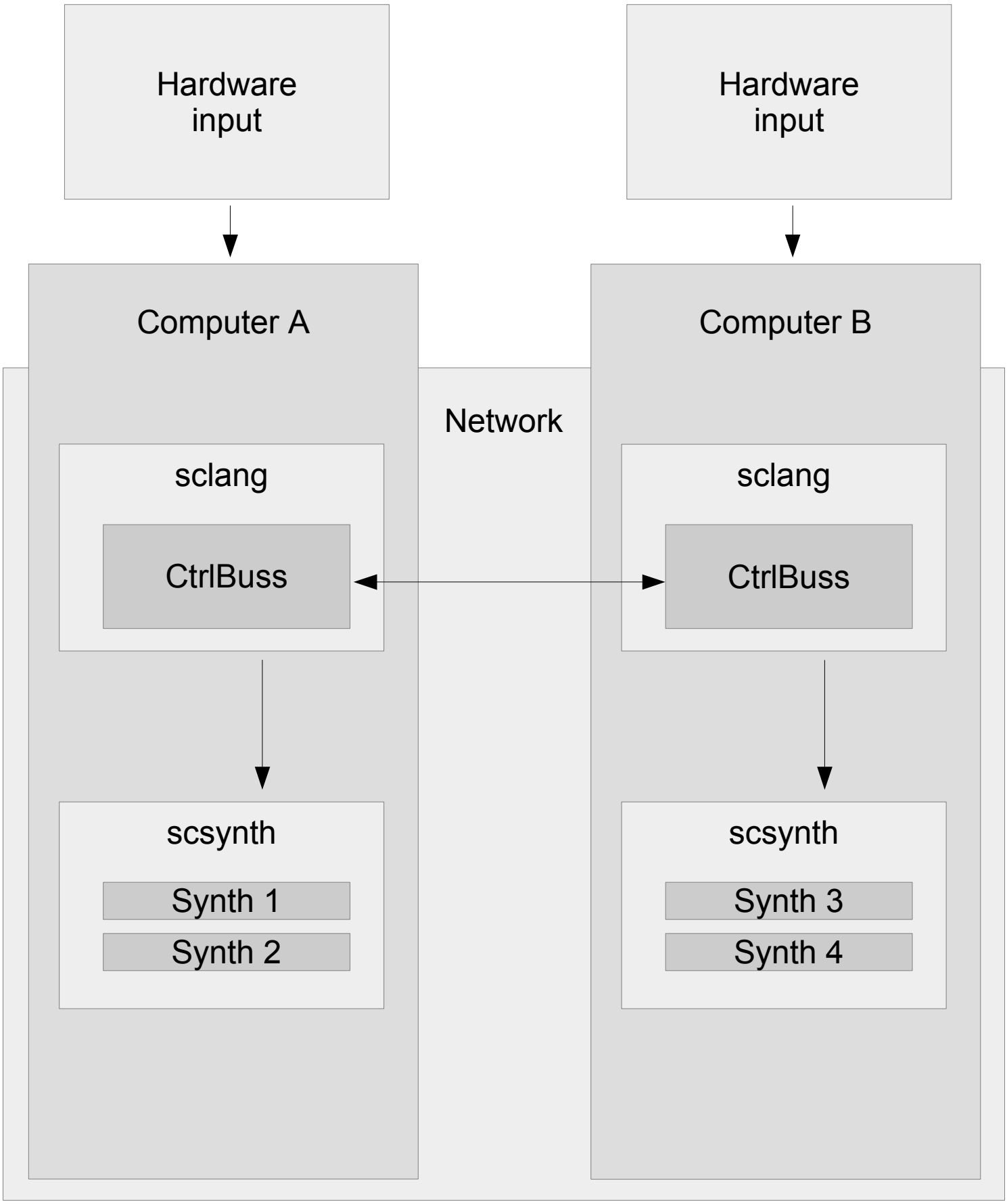
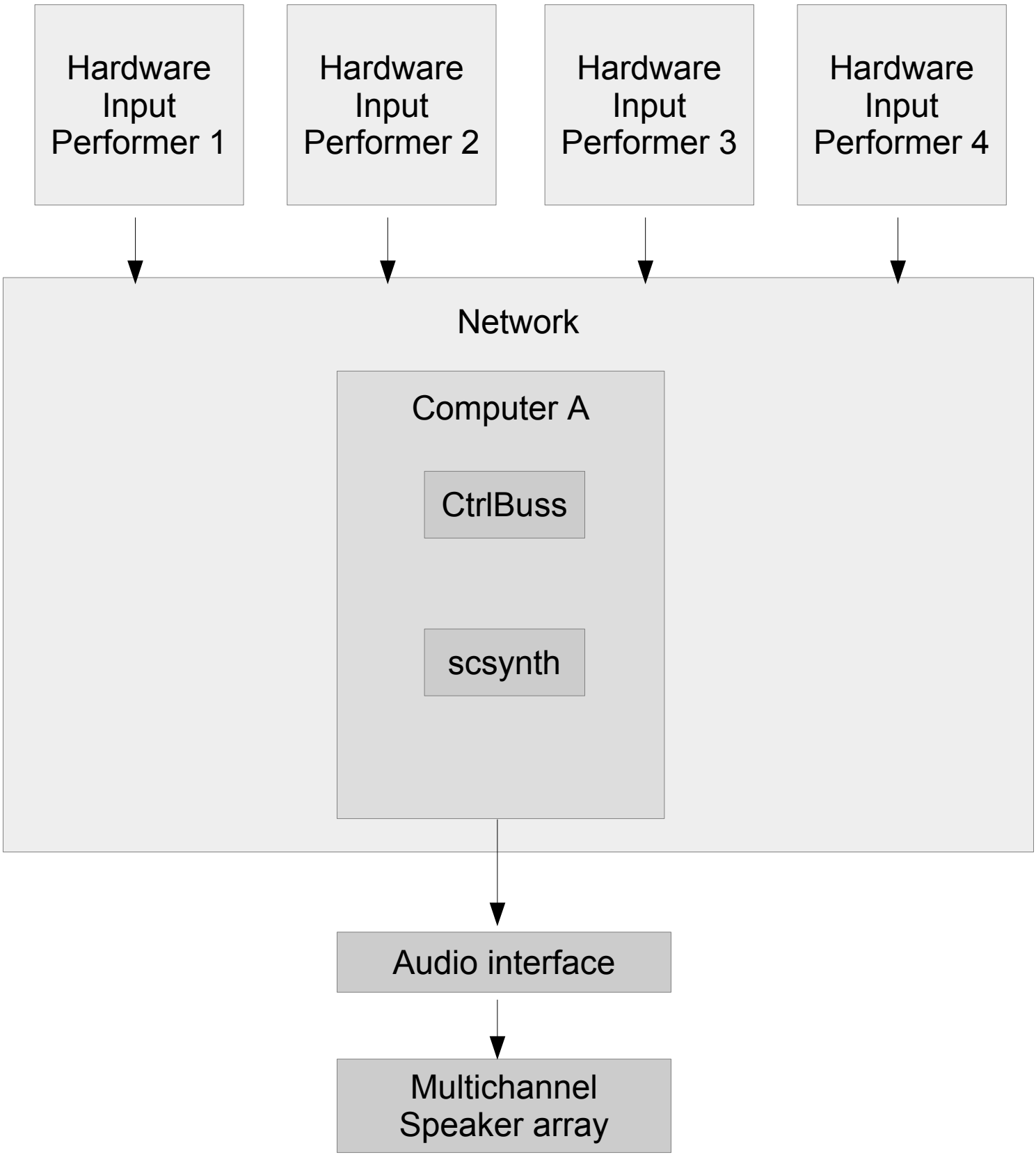
Scalability example 4



Scalability example 5



Scalability example 6



Scalability example 7