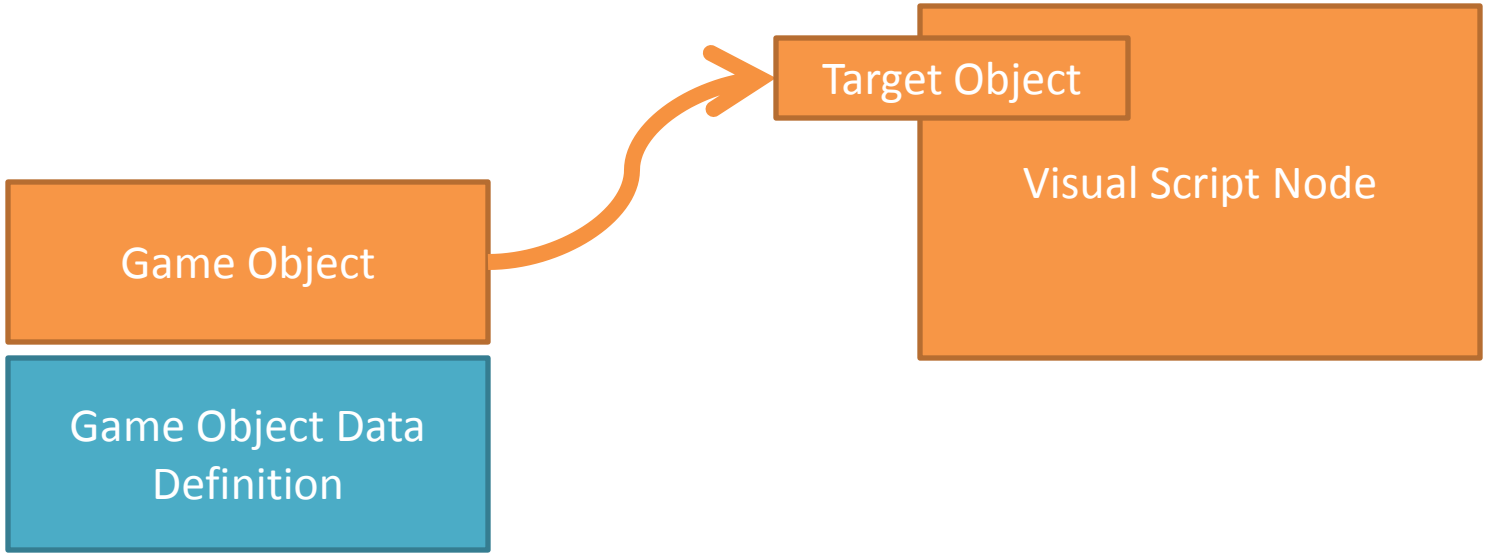
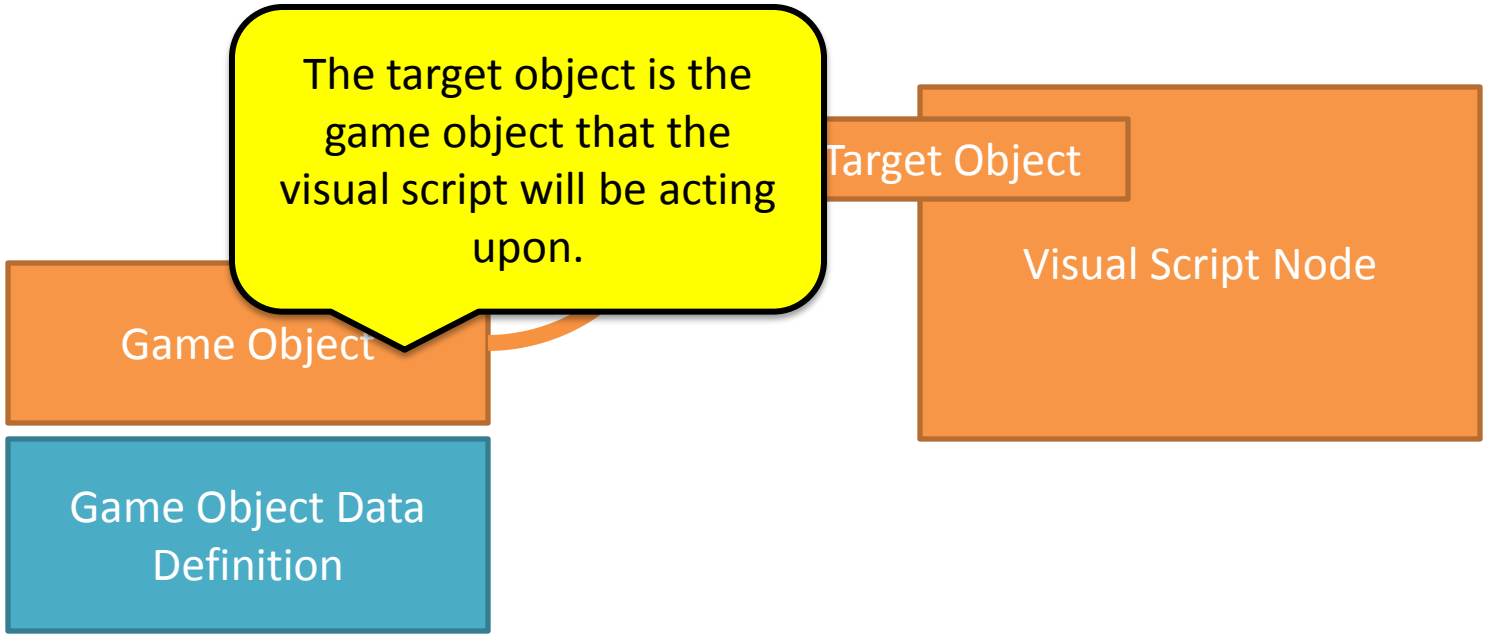
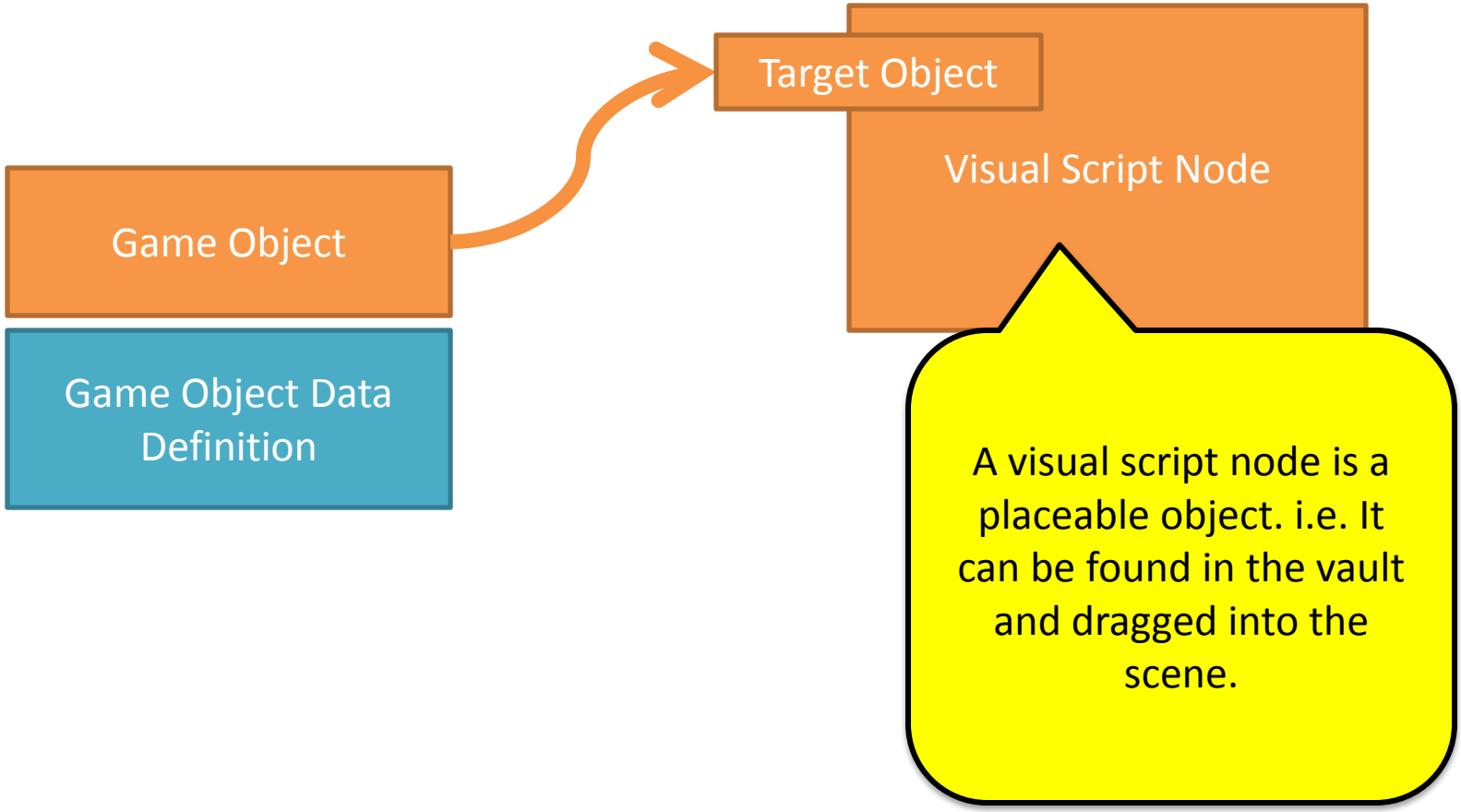
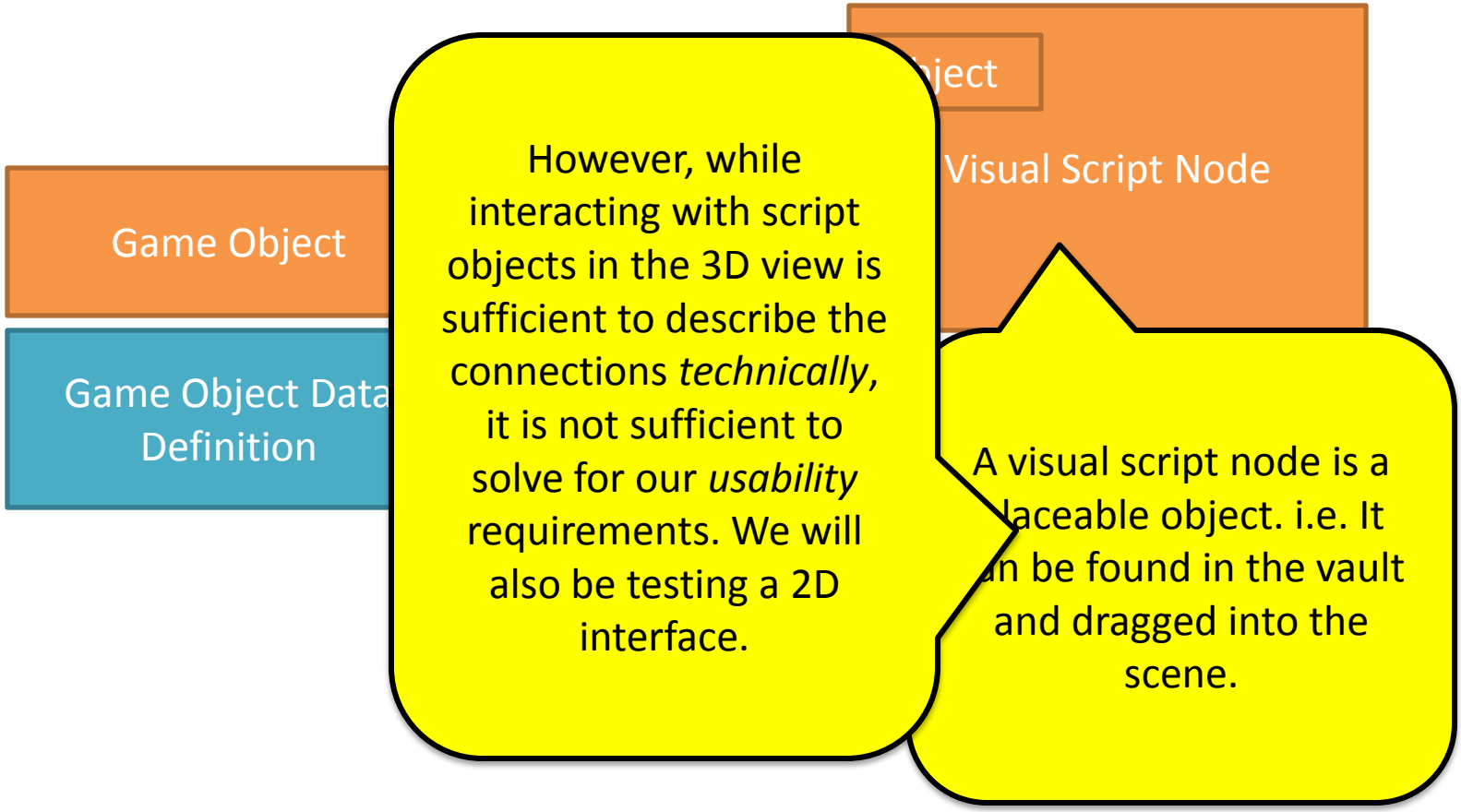
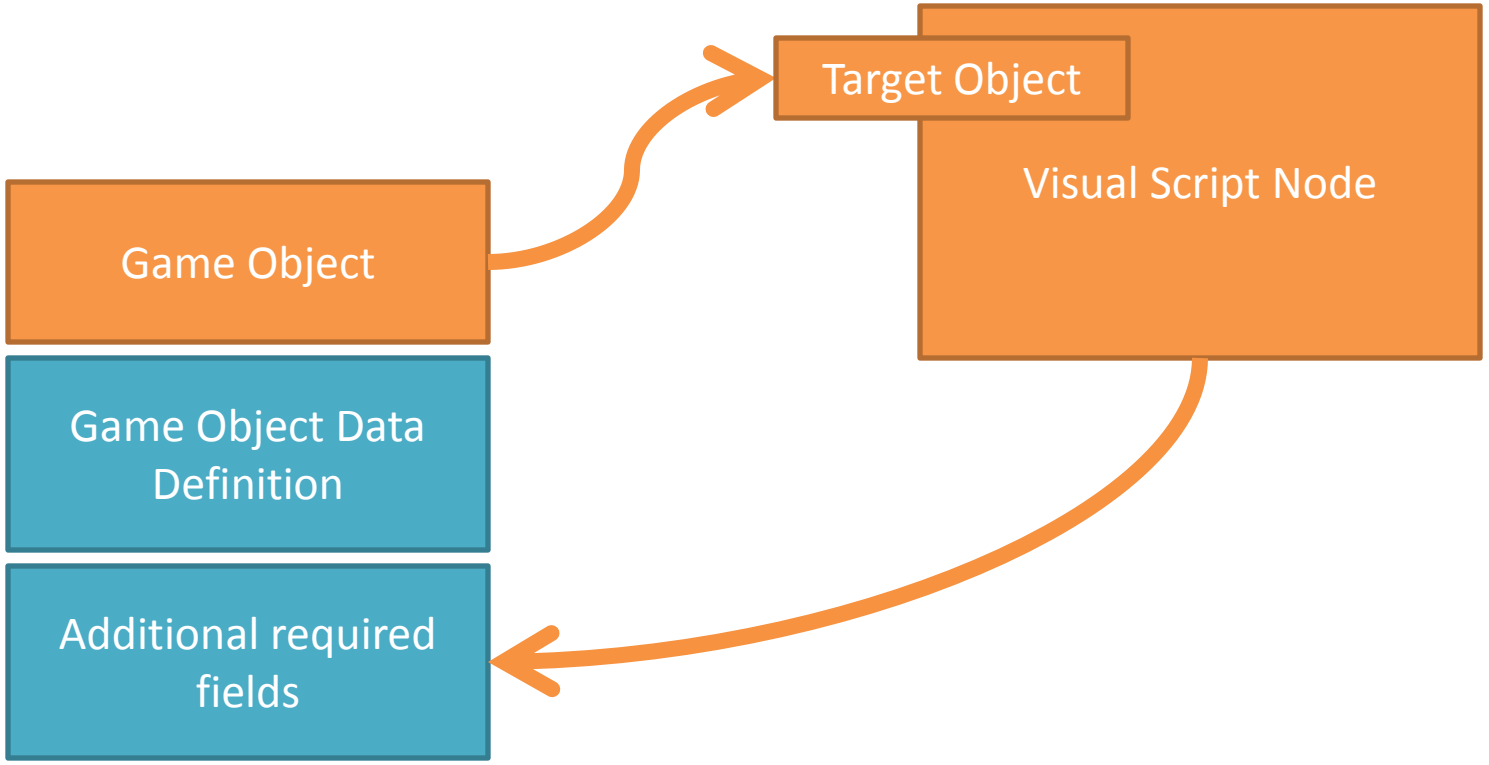
Visual Scripting
Dataflow

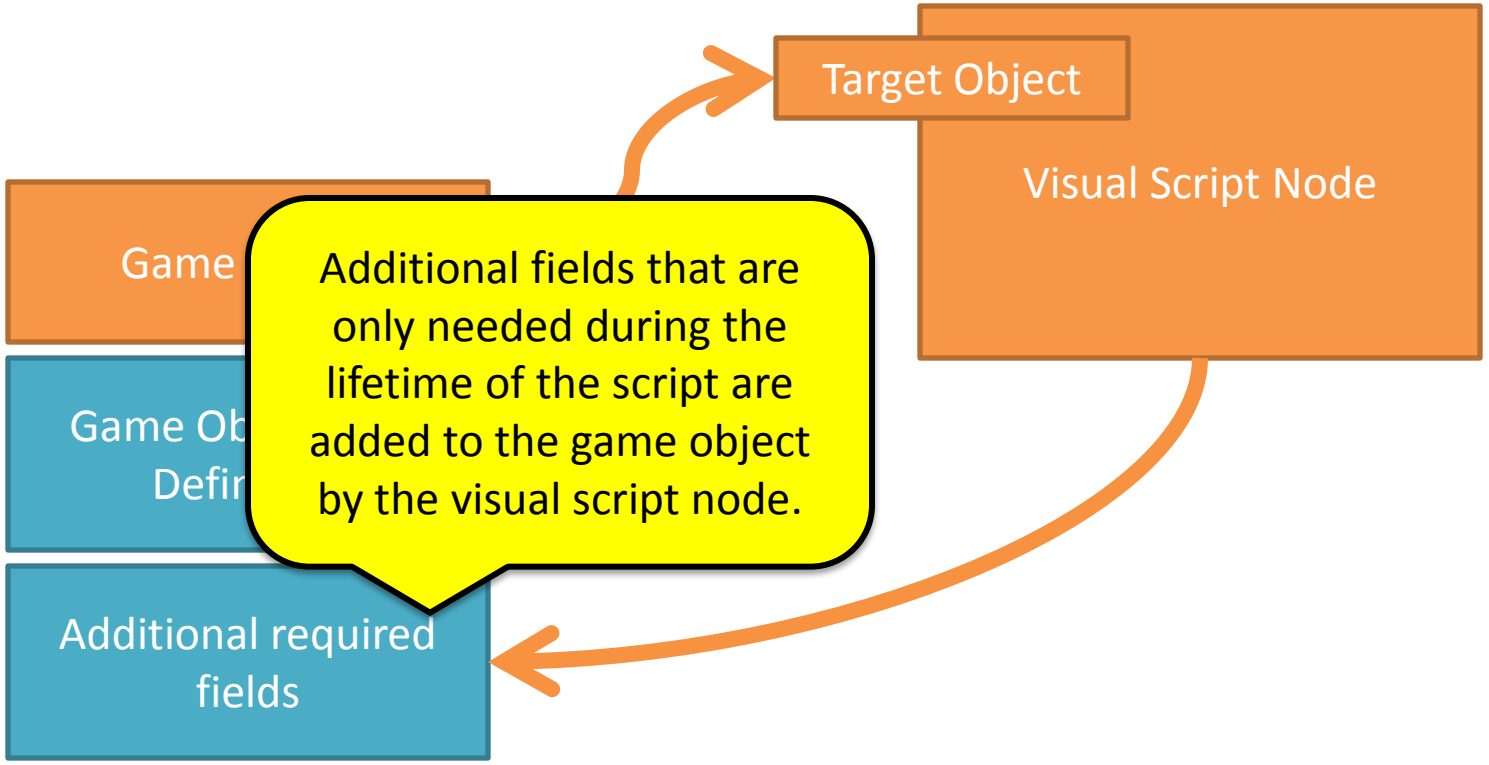

Target Object
Visual Script Node
Game Object
Game Object Data Definition

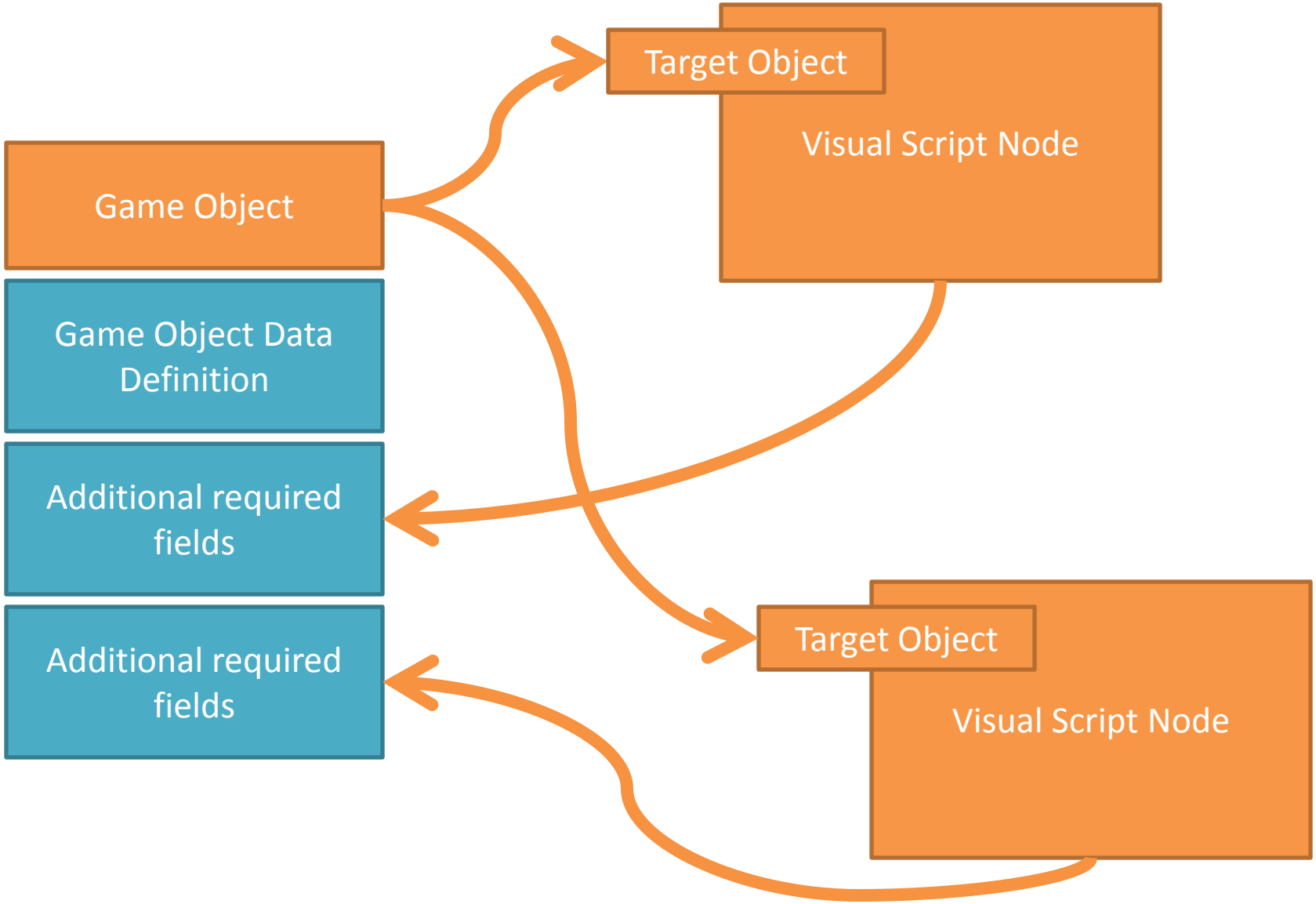

The target object is the game object that the visual script will be acting upon.
Target Object
Visual Script Node
Game Object
Game Object Data Definition

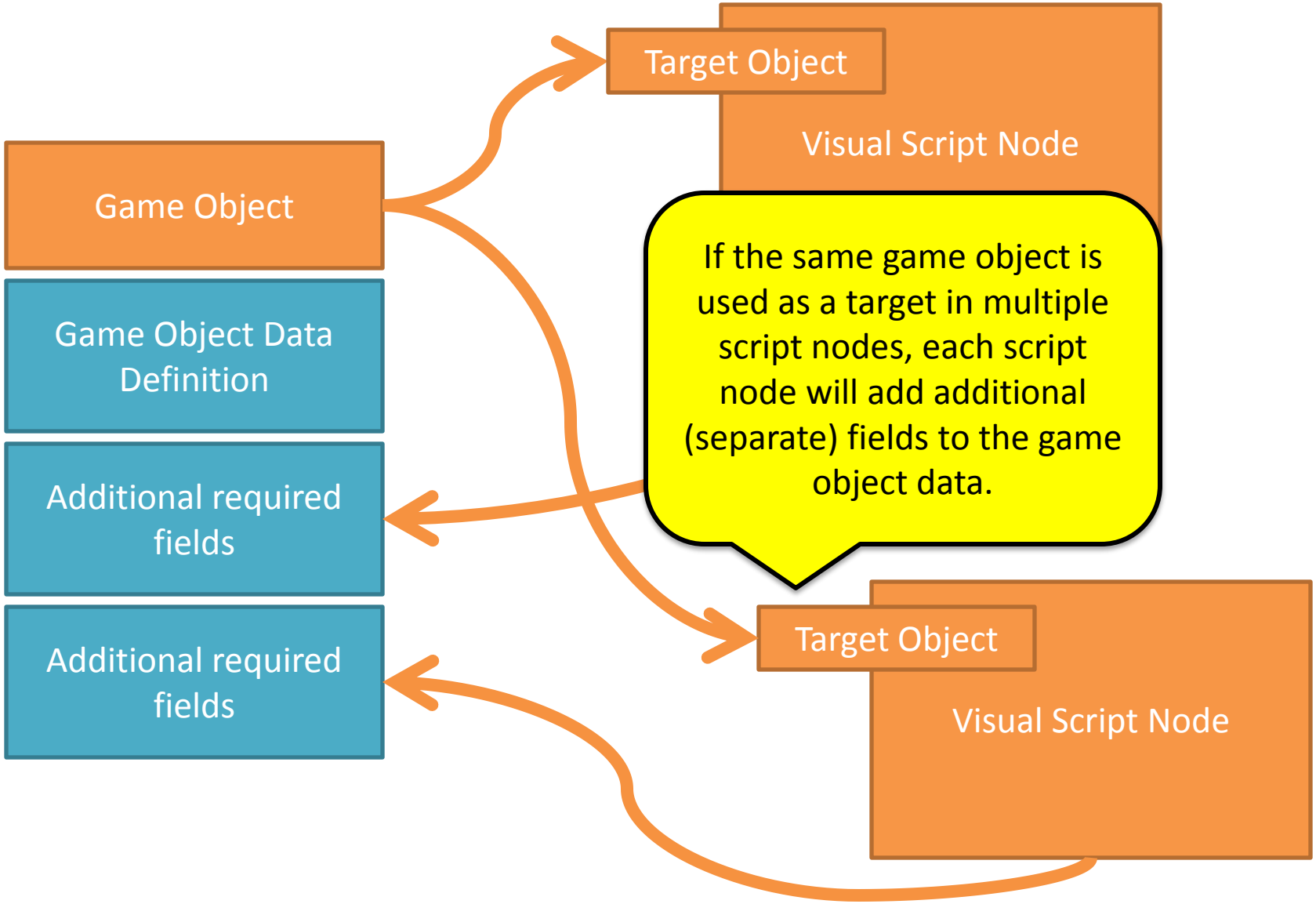

Target Object
Visual Script Node
Game Object
Game Object Data Definition
A visual script node is a placeable object. i.e. It can be found in the vault and dragged into the scene.

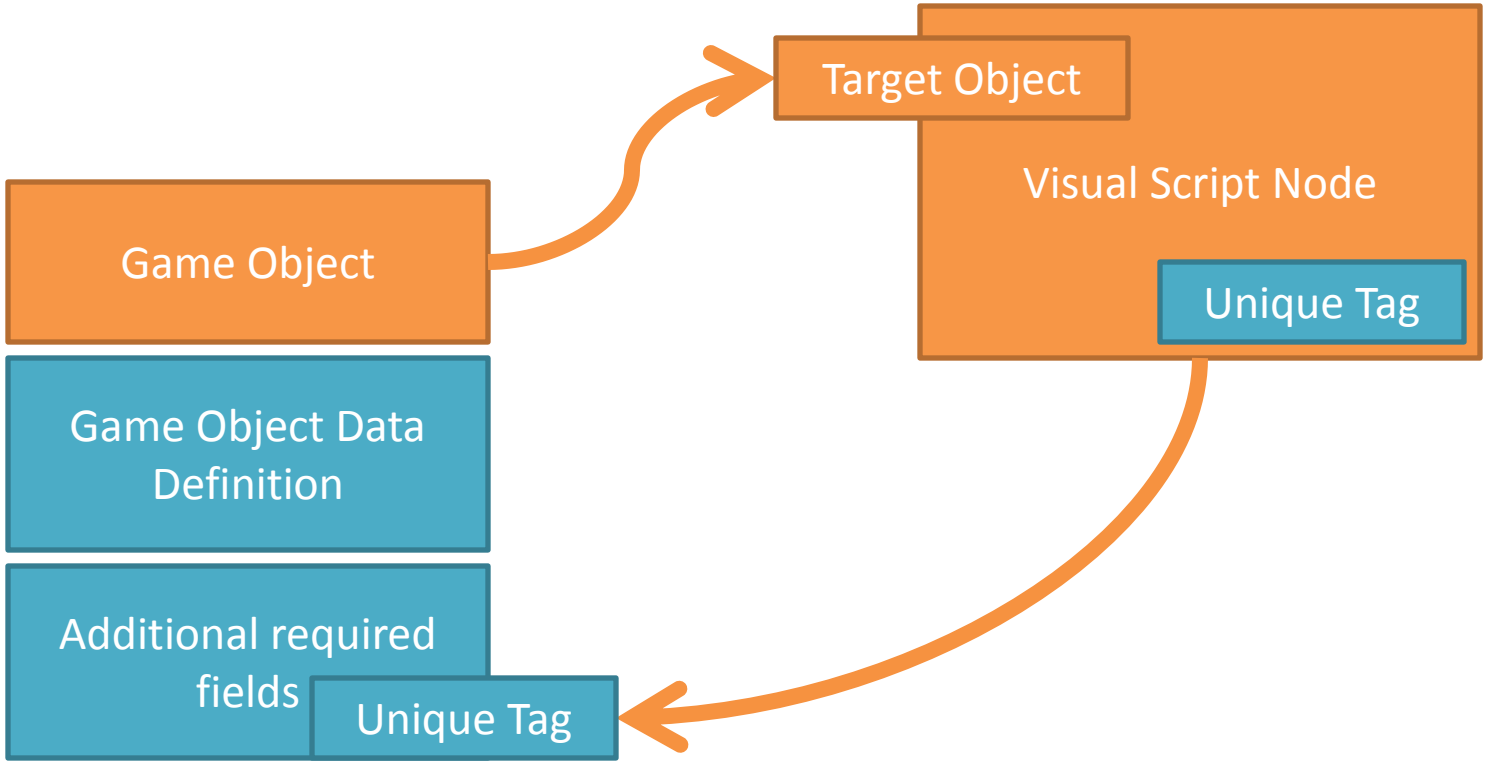

Game Object
Game Object Data Definition
ject
Visual Script Node
However, while interacting with script objects in the 3D view is sufficient to describe the connections *technically*, it is not sufficient to solve for our *usability* requirements. We will also be testing a 2D interface.
A visual script node is a laceable object. i.e. It n be found in the vault and dragged into the scene.

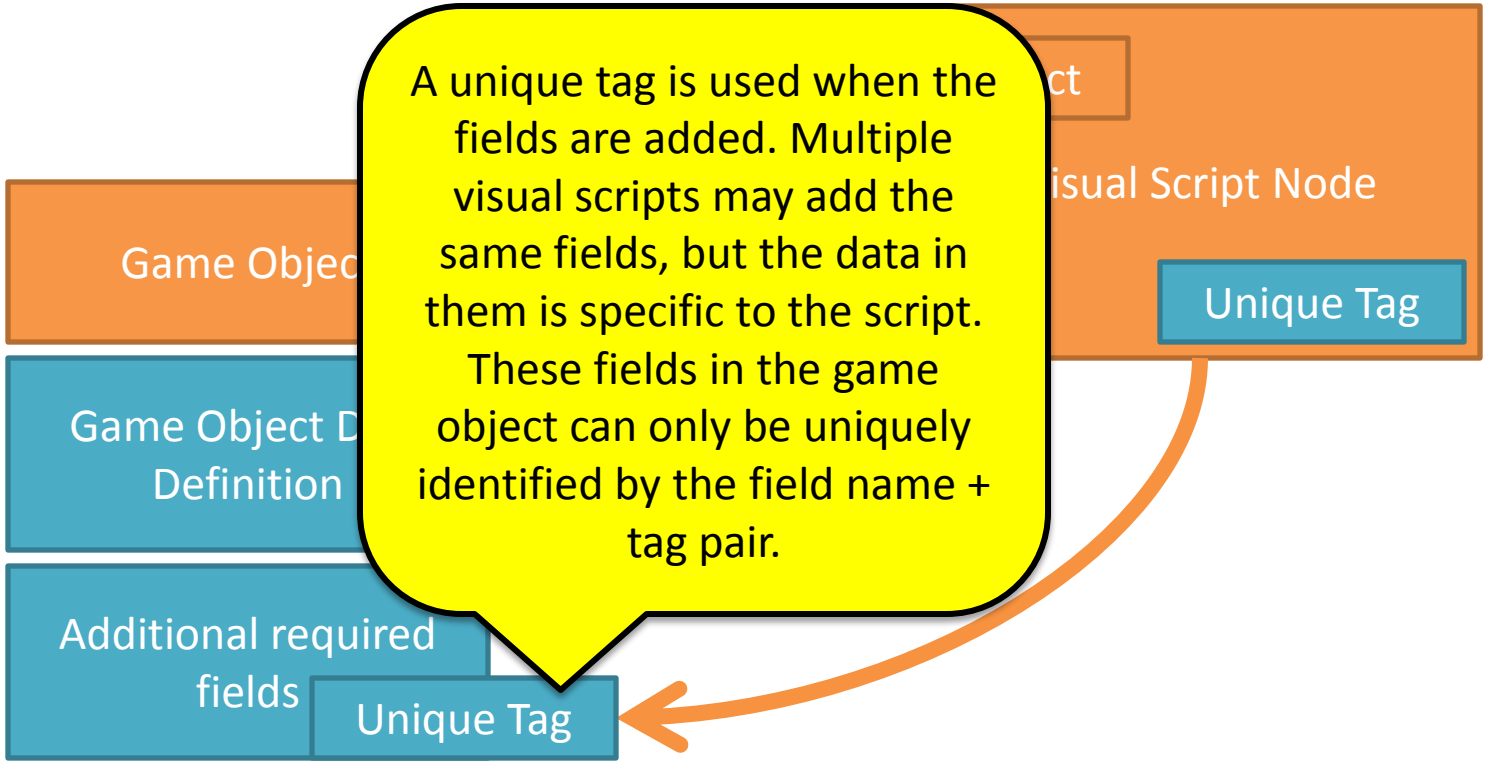

Target Object
Visual Script Node
Game Object
Game Object Data Definition
Additional required fields

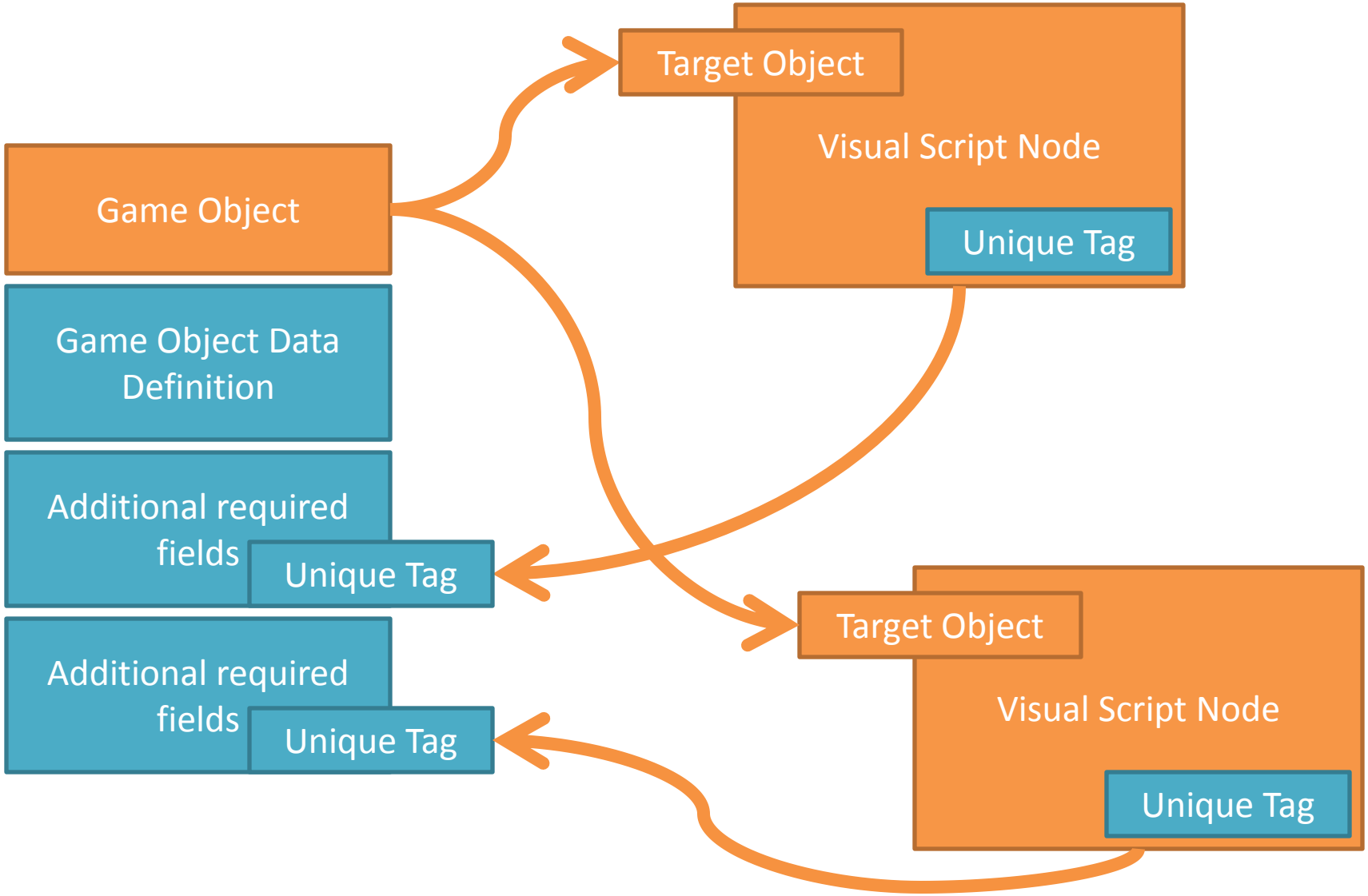

Target Object
Visual Script Node
Game
Game Ob
Defir
Additional fields that are only needed during the lifetime of the script are added to the game object by the visual script node.
Additional required fields

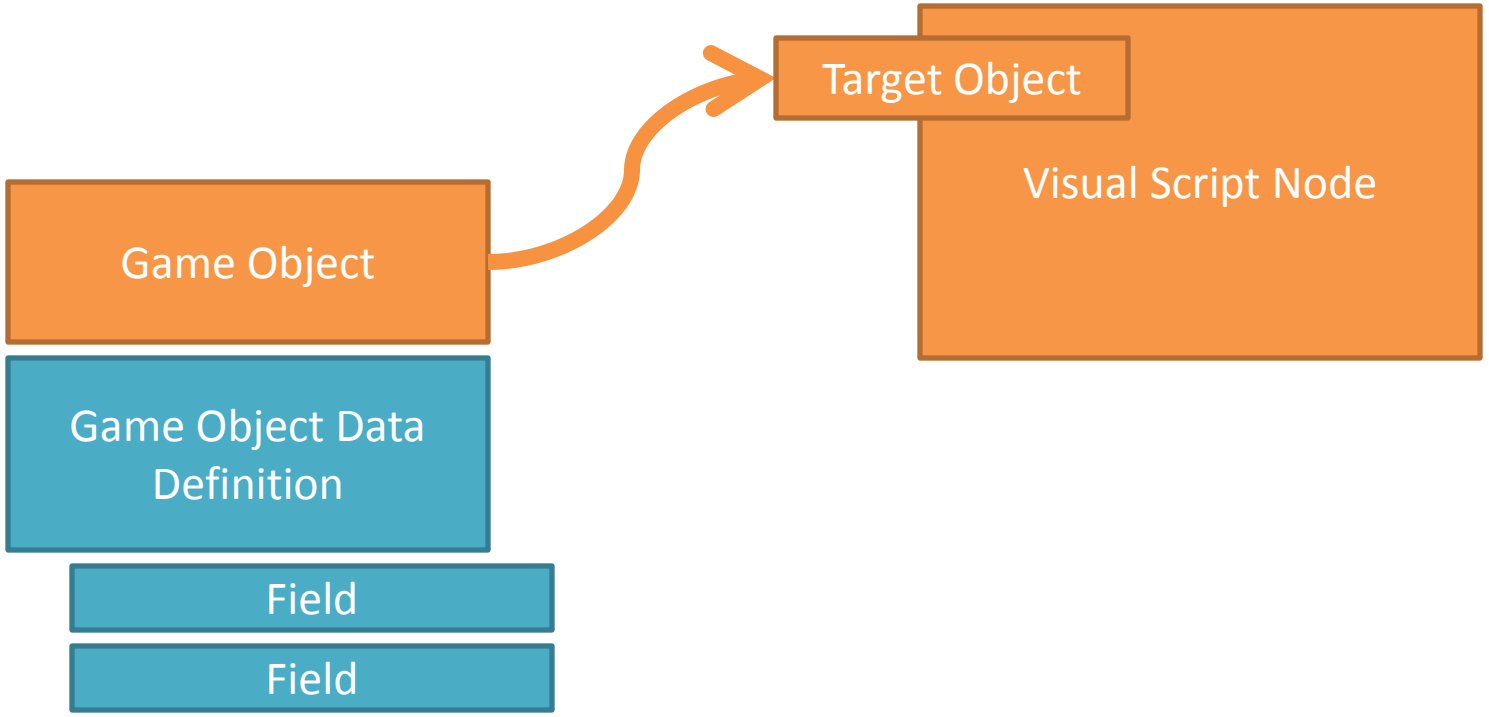

Target Object
Visual Script Node
Game Object
Game Object Data Definition
Additional required fields
Additional required fields
Target Object
Visual Script Node

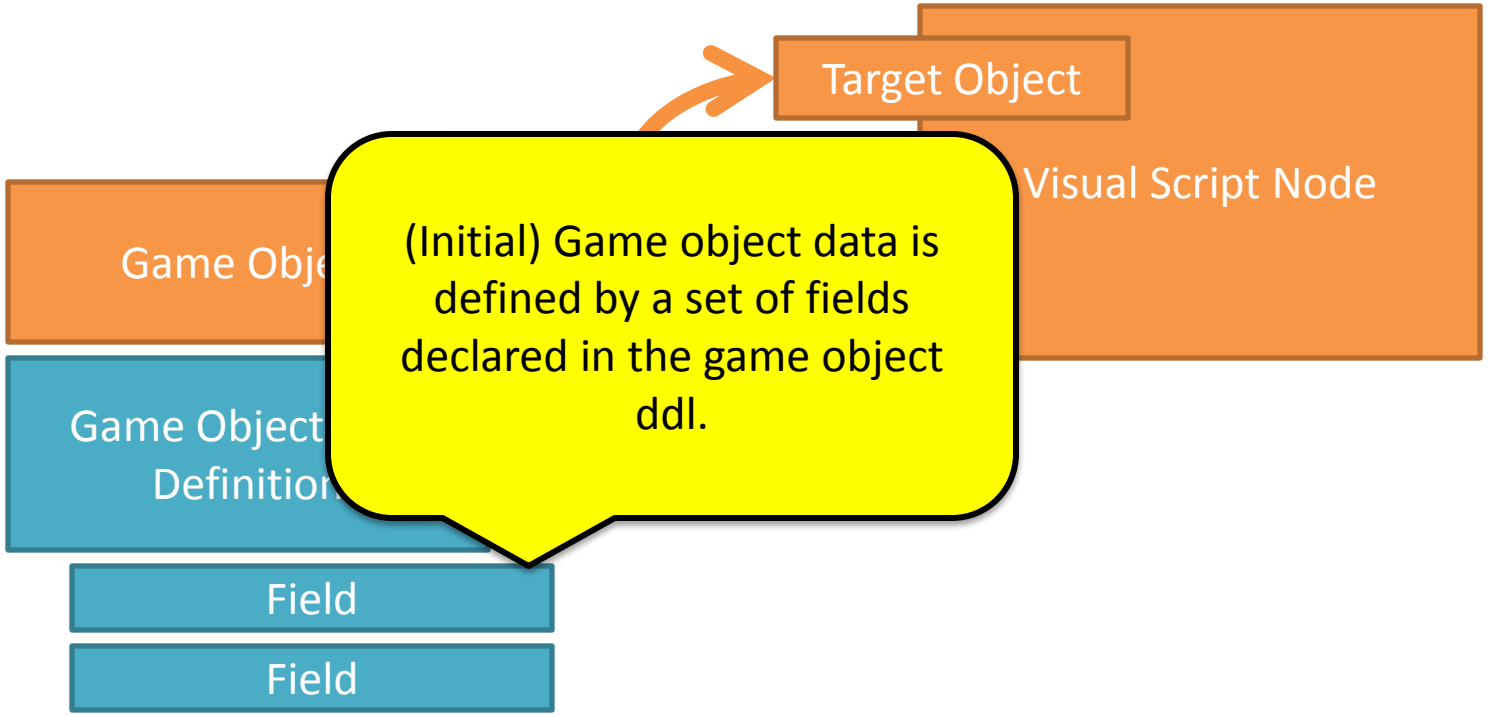

Target Object
Visual Script Node
Game Object
Game Object Data Definition
Additional required fields
Additional required fields
If the same game object is used as a target in multiple script nodes, each script node will add additional (separate) fields to the game object data.
Target Object
Visual Script Node

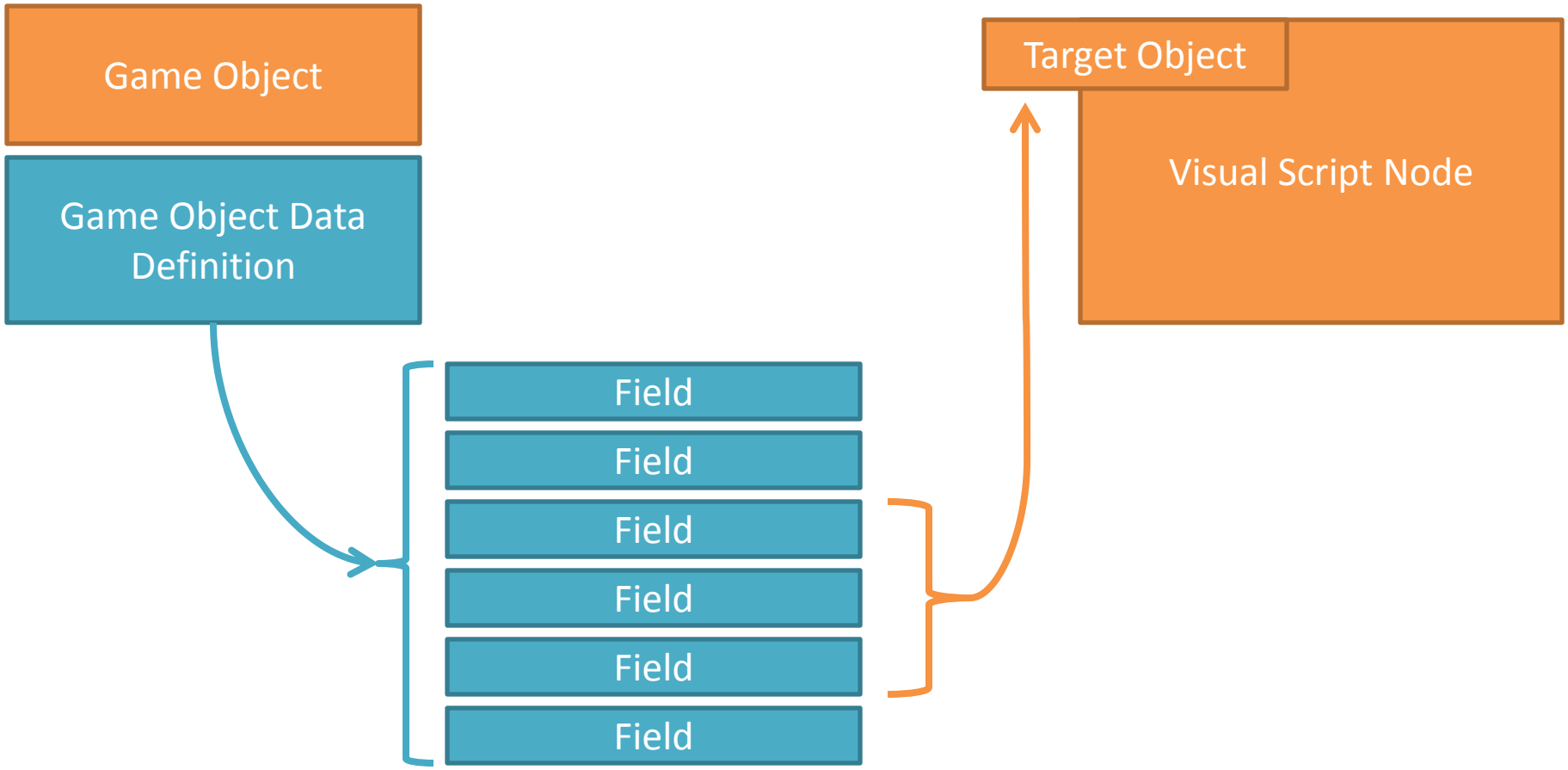

Target Object
Visual Script Node
Unique Tag
Game Object
Game Object Data Definition
Additional required fields
Unique Tag

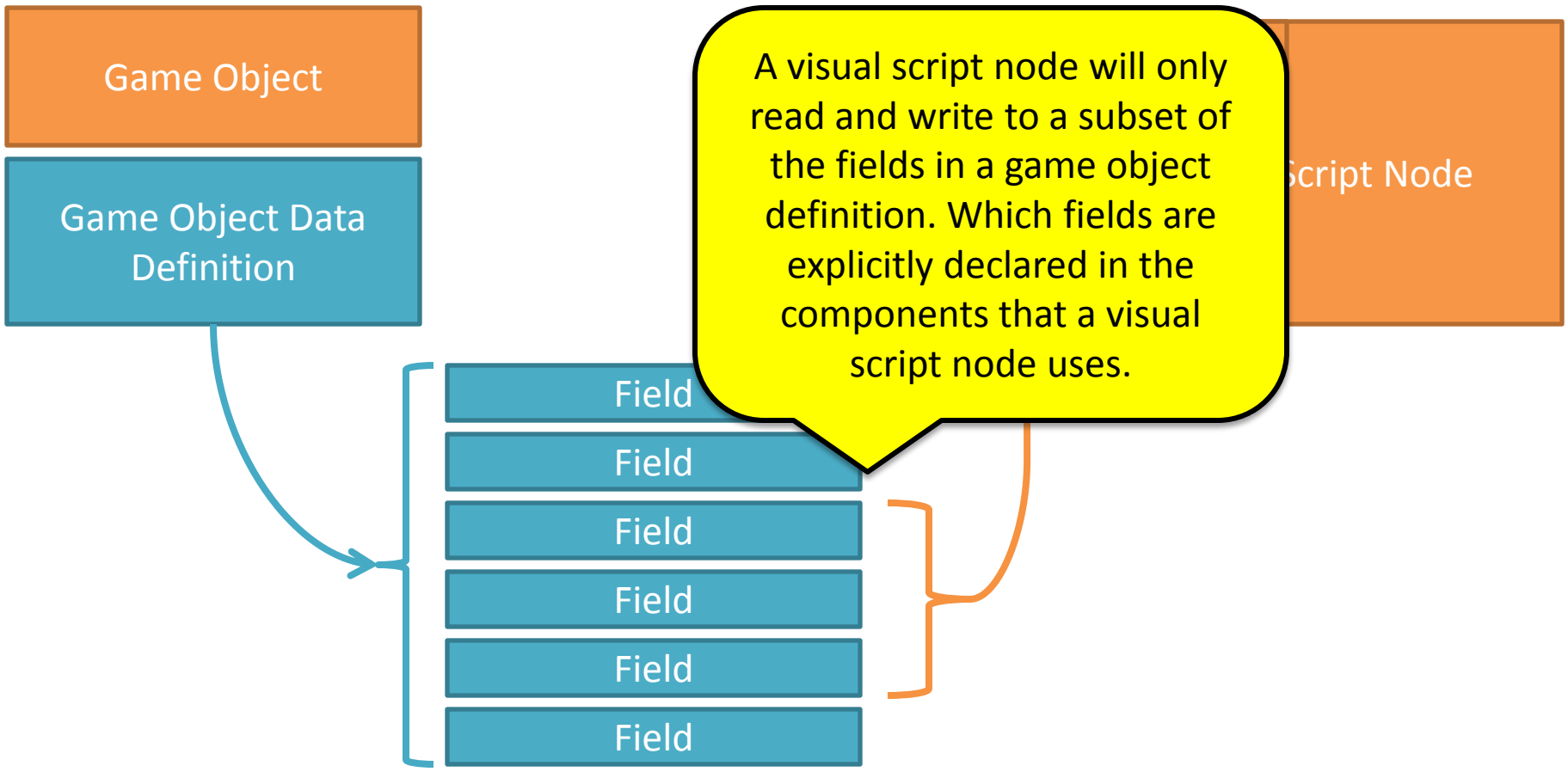

A unique tag is used when the fields are added. Multiple visual scripts may add the same fields, but the data in them is specific to the script. These fields in the game object can only be uniquely identified by the field name + tag pair.
ct
isual Script Node
Unique Tag
Game Objec
Game Object D
Definition
Additional required fields
Unique Tag

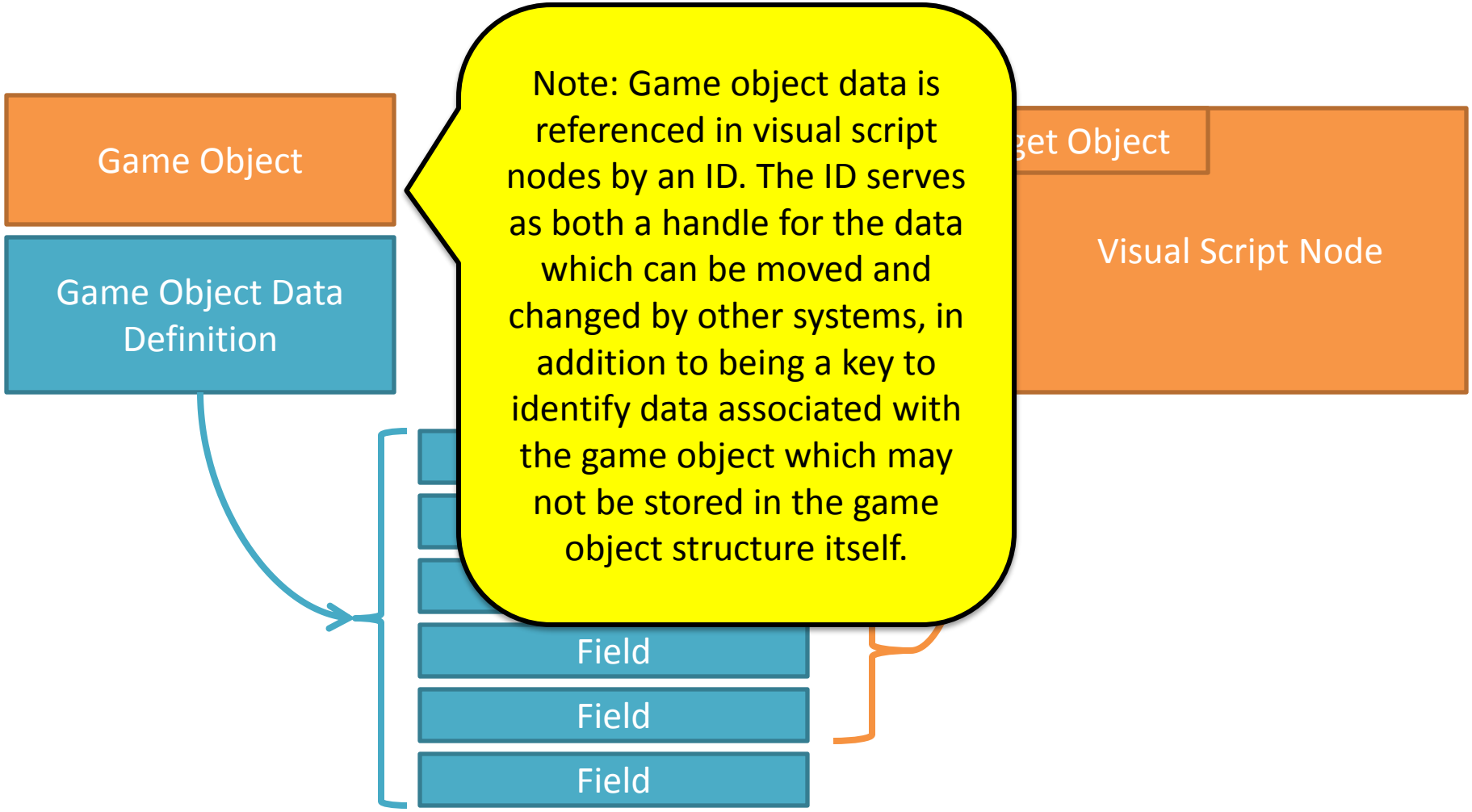

Target Object
Visual Script Node
Unique Tag
Game Object
Game Object Data Definition
Additional required fields
Unique Tag
Additional required fields
Unique Tag
Target Object
Visual Script Node
Unique Tag

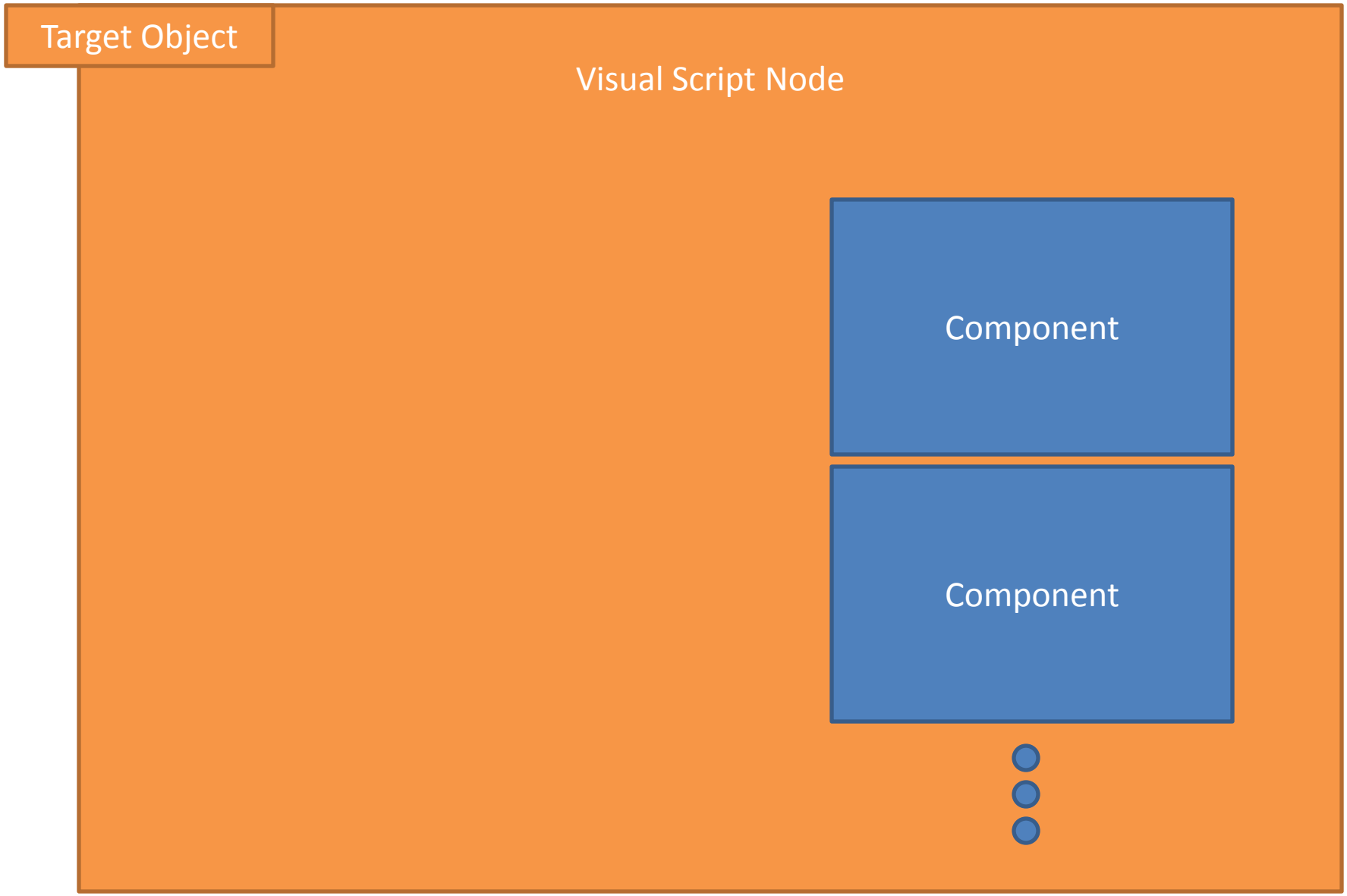

Target Object
Visual Script Node
Game Object
Game Object Data Definition
Field
Field

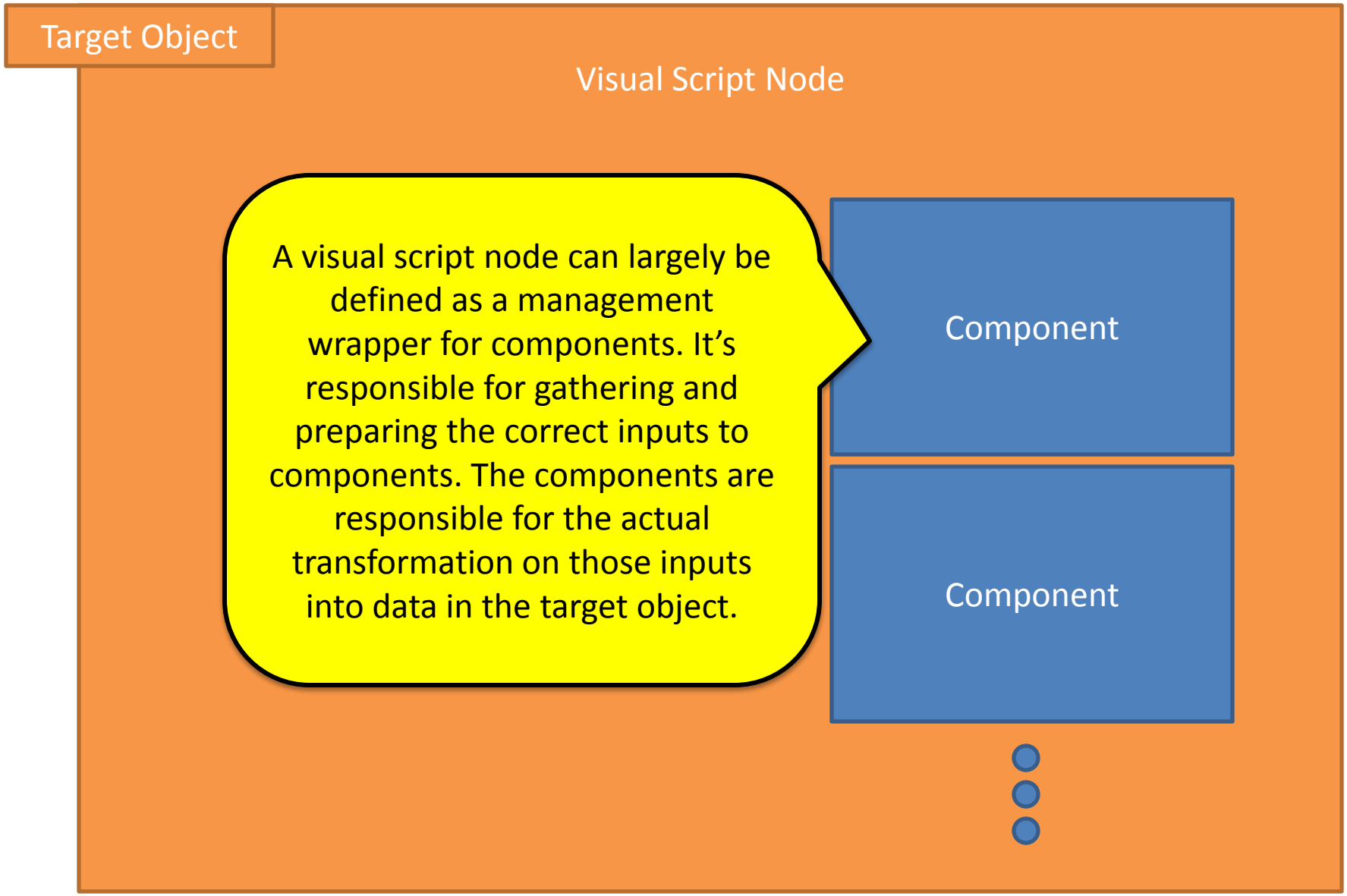

Target Object
Visual Script Node
Game Obje
Game Object
Definitio
(Initial) Game object data is defined by a set of fields declared in the game object ddl.
Field
Field

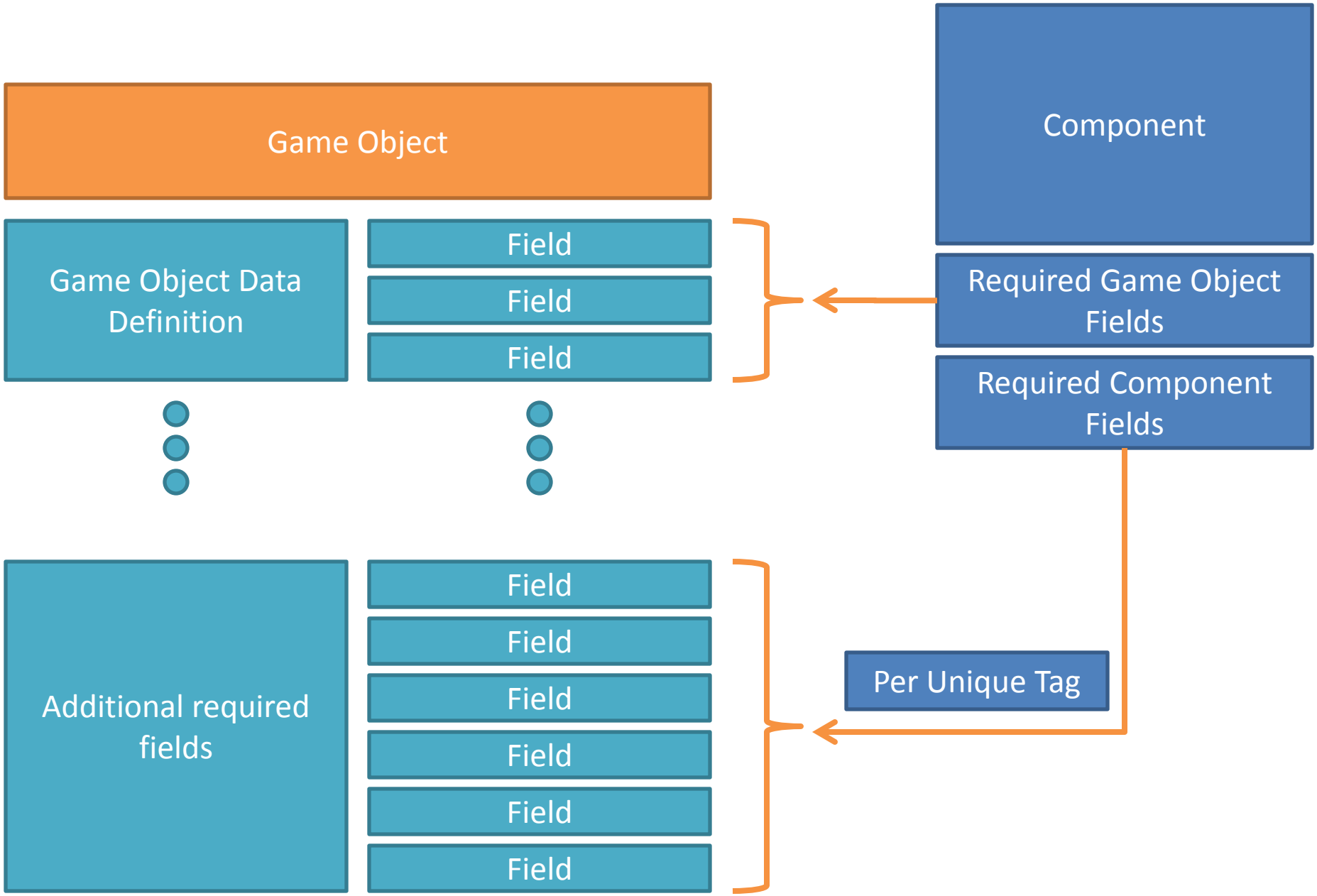

Game Object
Game Object Data Definition
Field
Field
Field
Field
Field
Field
Target Object
Visual Script Node

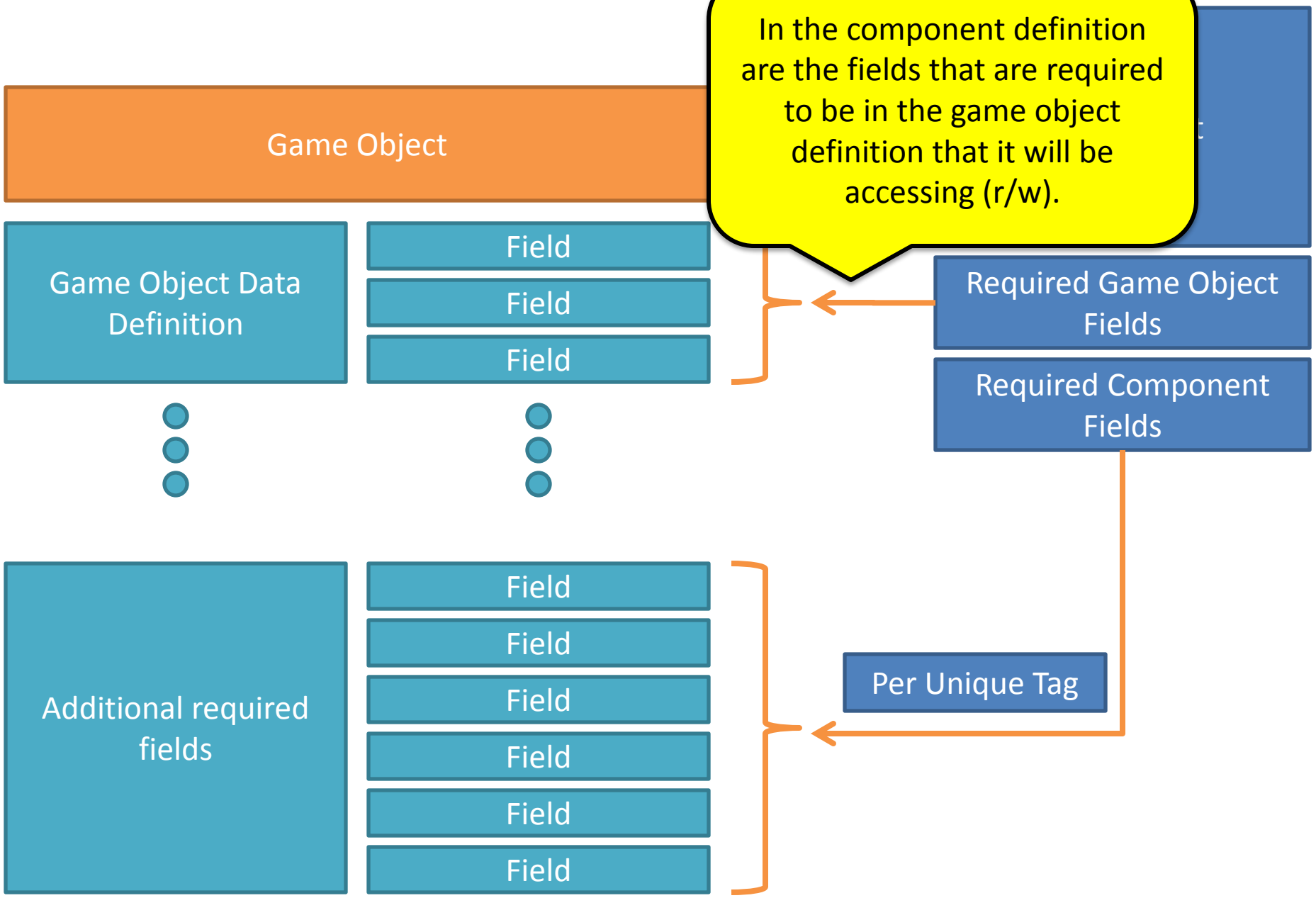

Game Object
Game Object Data Definition
Field
Field
Field
Field
Field
Field
A visual script node will only read and write to a subset of the fields in a game object definition. Which fields are explicitly declared in the components that a visual script node uses.
Script Node

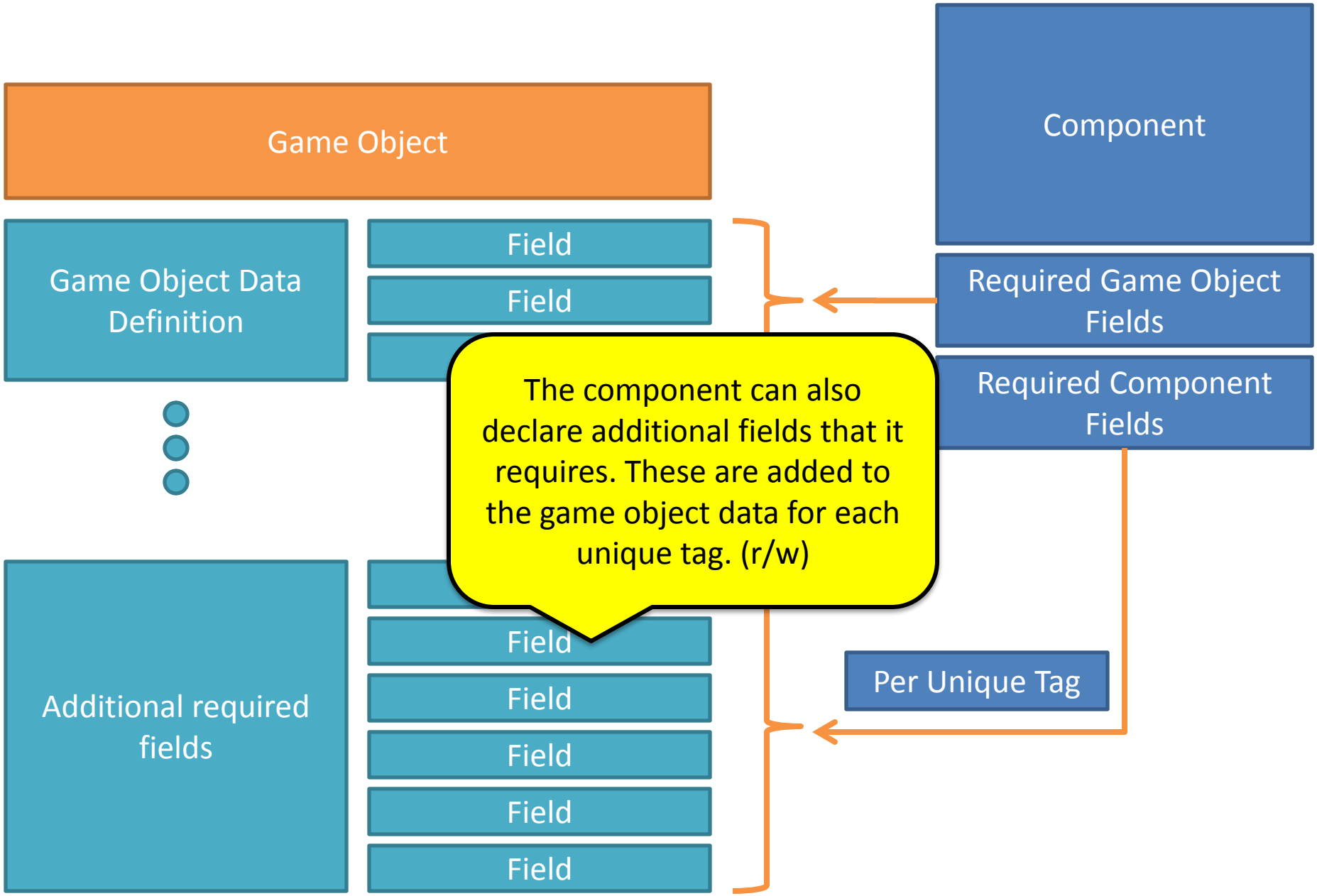

Game Object
Game Object Data Definition
Note: Game object data is referenced in visual script nodes by an ID. The ID serves as both a handle for the data which can be moved and changed by other systems, in addition to being a key to identify data associated with the game object which may not be stored in the game object structure itself.
get Object
Visual Script Node
Field
Field
Field

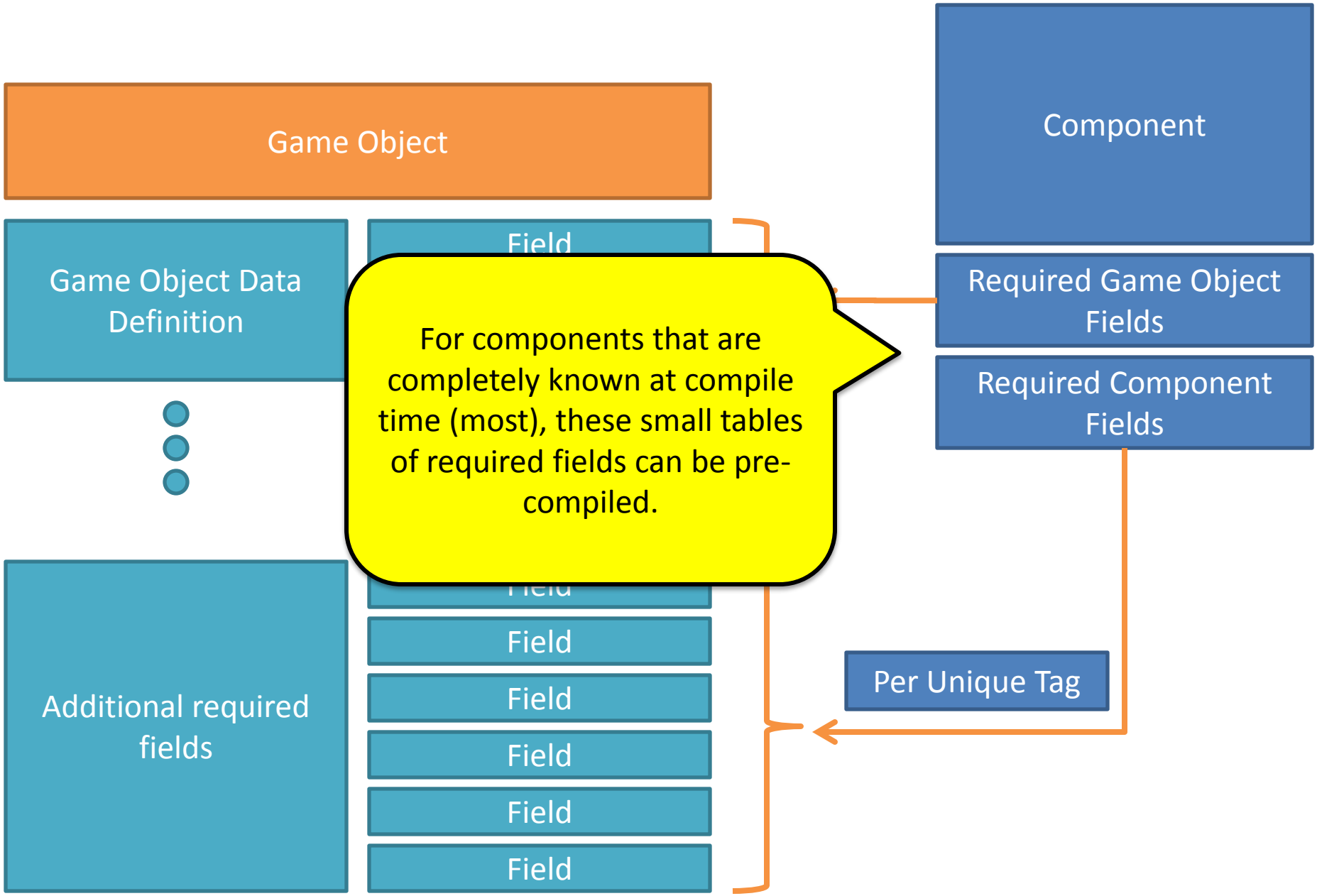

Target Object
Visual Script Node
Component
Component

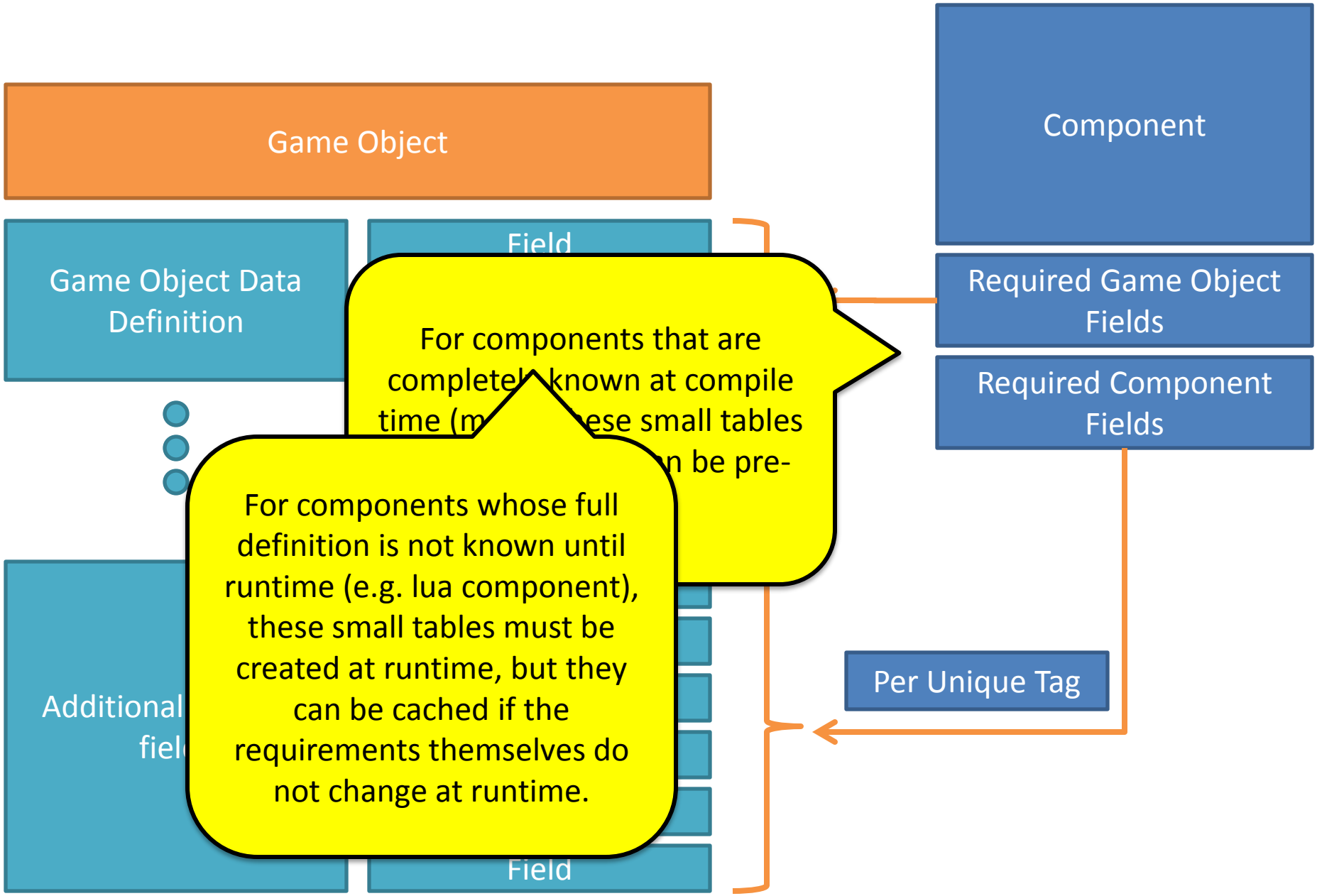

Target Object
Visual Script Node
A visual script node can largely be defined as a management wrapper for components. It's responsible for gathering and preparing the correct inputs to components. The components are responsible for the actual transformation on those inputs into data in the target object.
Component
Component

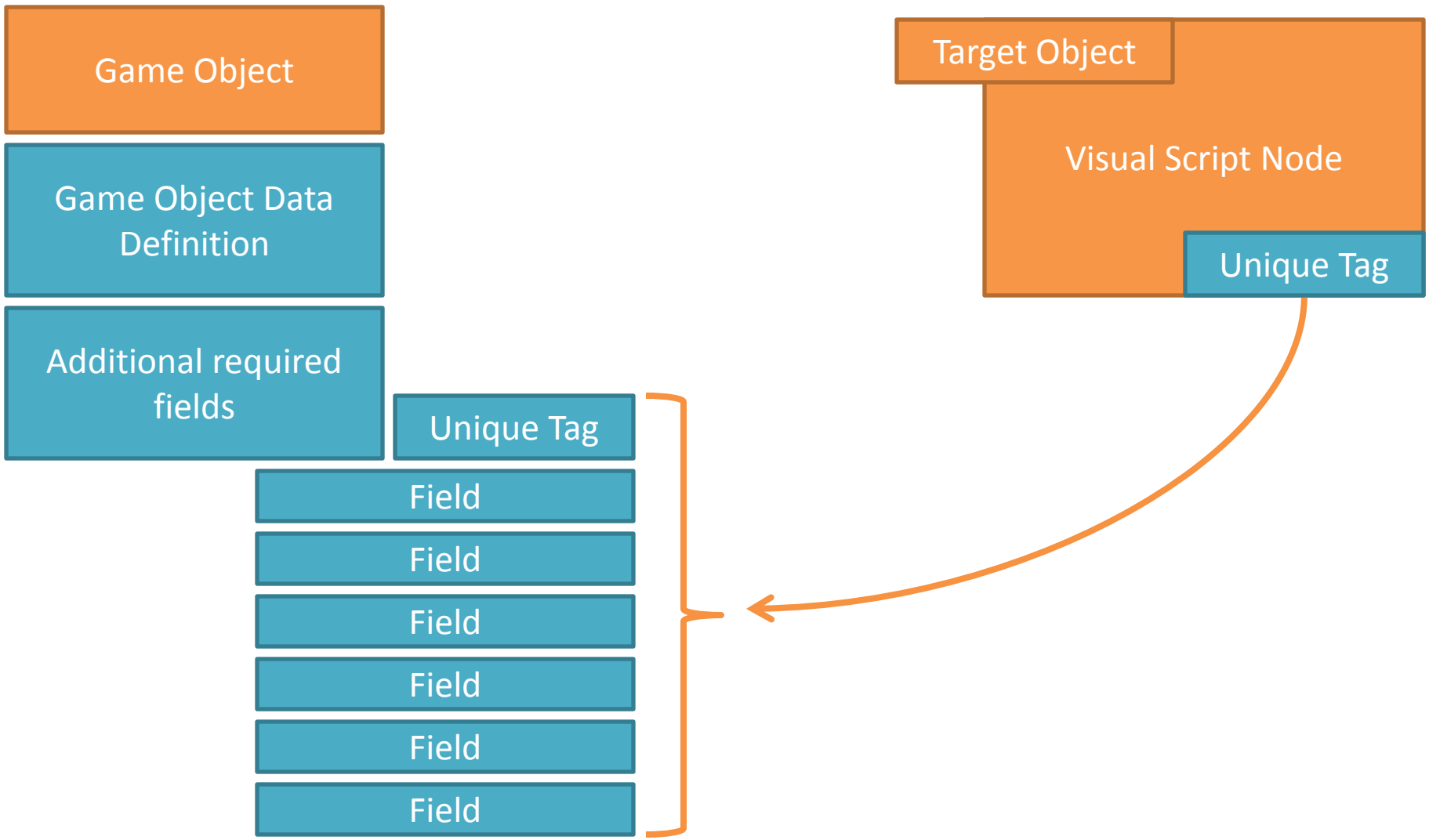

Game Object
Component
Game Object Data Definition
Field
Field
Field
Required Game Object Fields
Required Component Fields
Additional required fields
Field
Field
Field
Field
Field
Field
Per Unique Tag

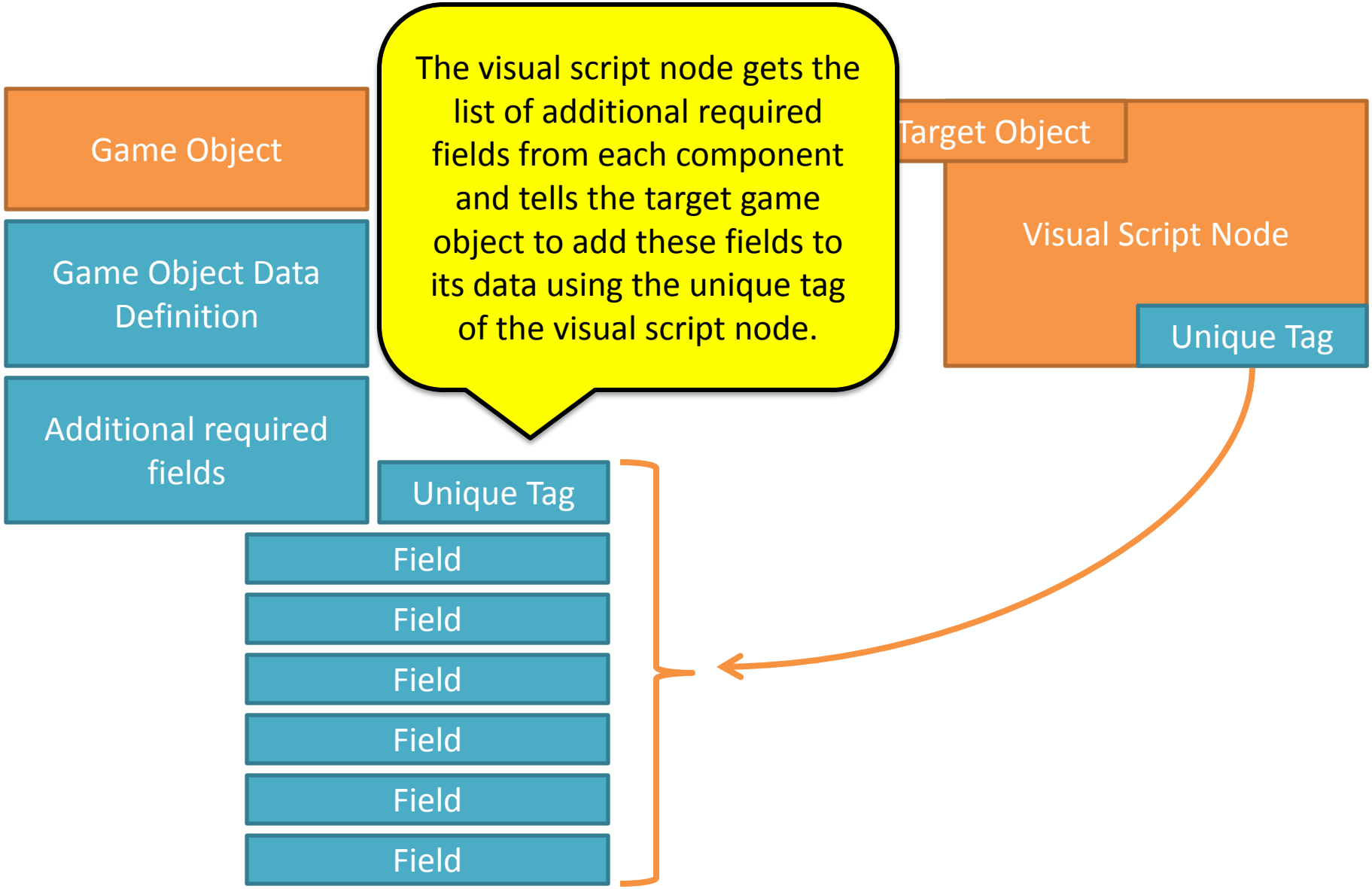

In the component definition are the fields that are required to be in the game object definition that it will be accessing (r/w).
Game Object
Game Object Data Definition
Field
Field
Field
Required Game Object Fields
Required Component Fields
Per Unique Tag
Additional required fields
Field
Field
Field
Field
Field
Field

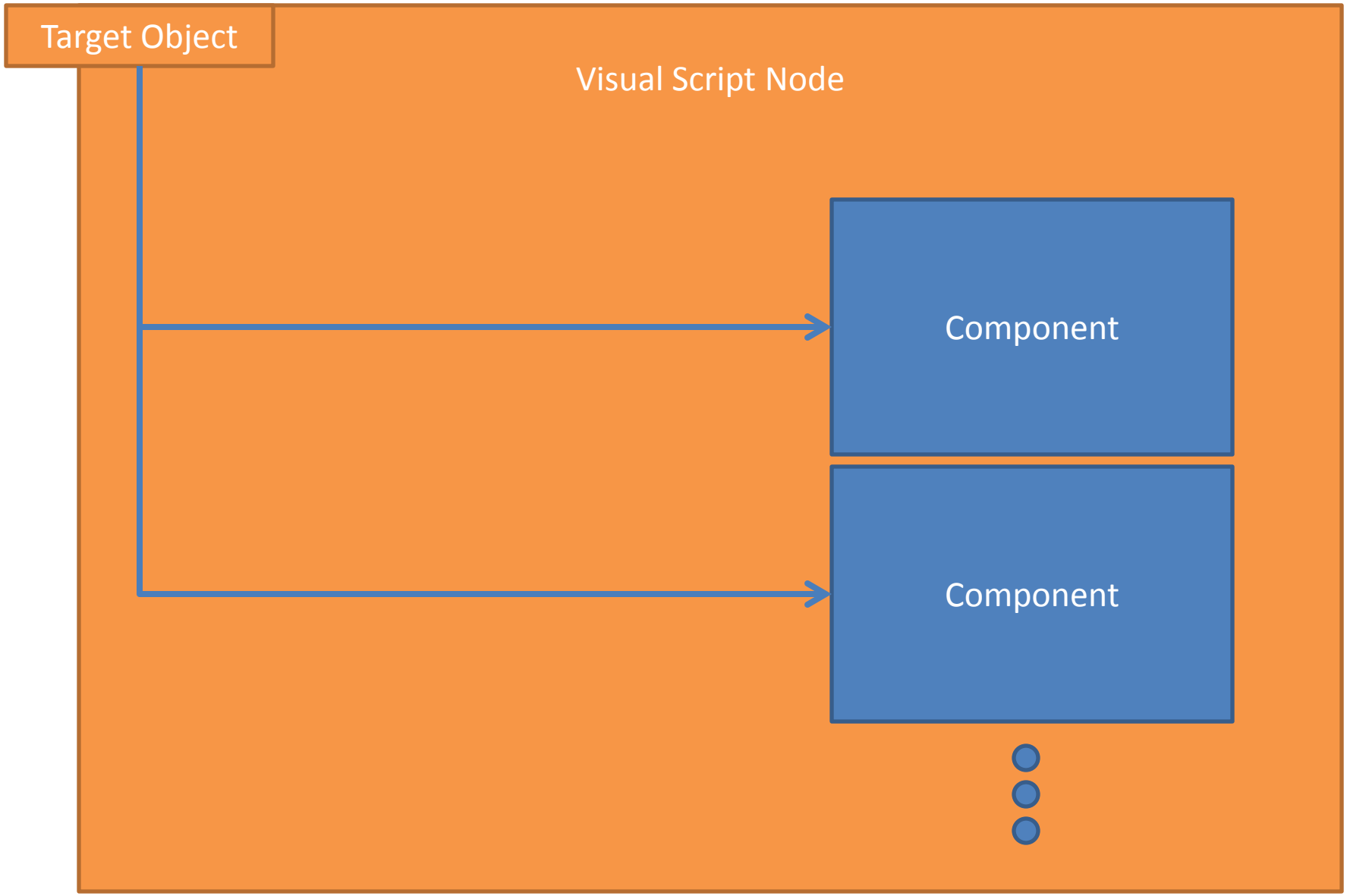

Game Object
Component
Game Object Data Definition
Field
Field
Required Game Object Fields
Required Component Fields
The component can also declare additional fields that it requires. These are added to the game object data for each unique tag. (r/w)
Per Unique Tag
Additional required fields
Field
Field
Field
Field
Field

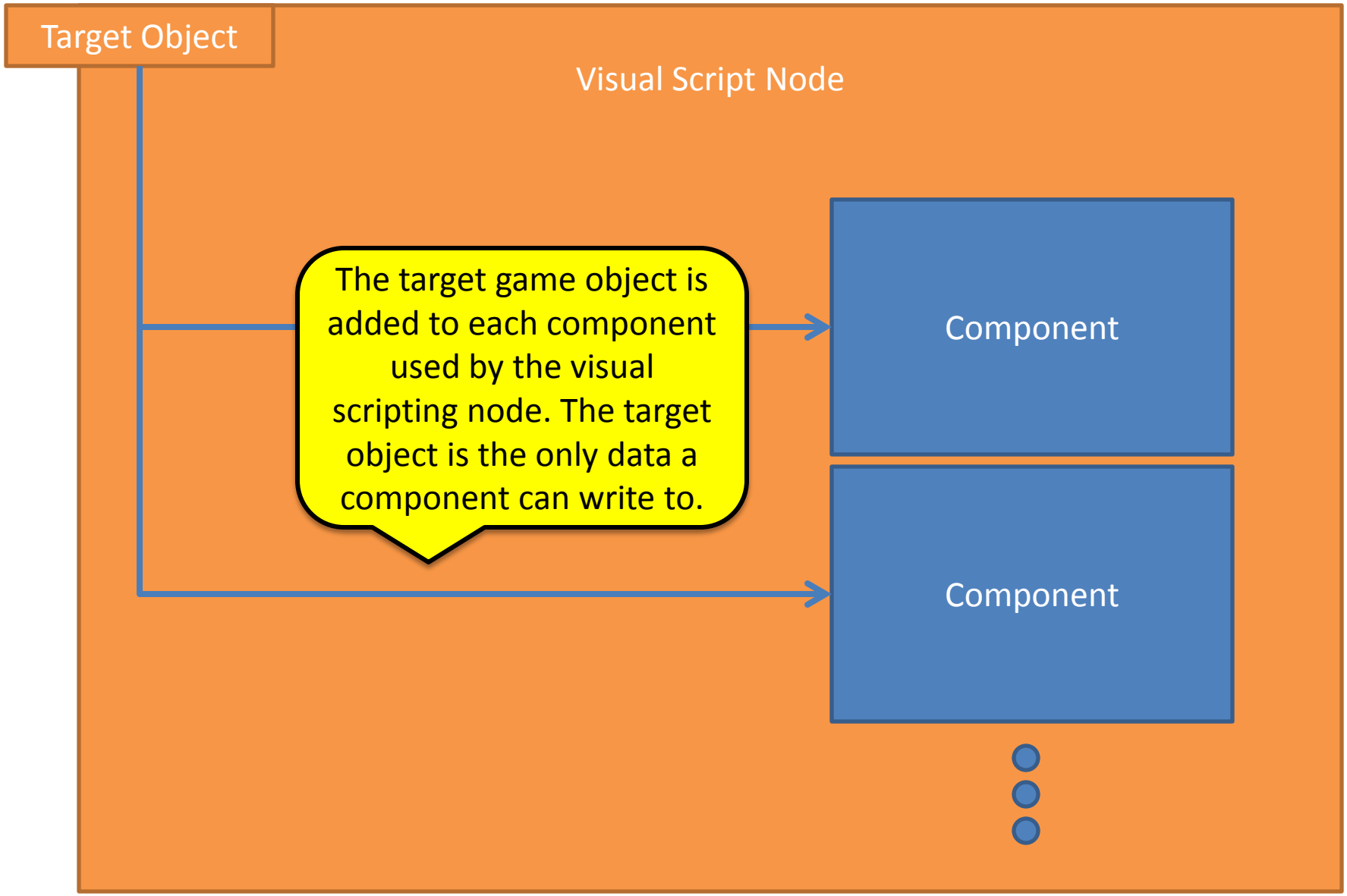

Game Object
Component
Game Object Data Definition
Field
Required Game Object Fields
Required Component Fields
For components that are completely known at compile time (most), these small tables of required fields can be pre-compiled.
Additional required fields
Field
Field
Field
Field
Field
Field
Per Unique Tag

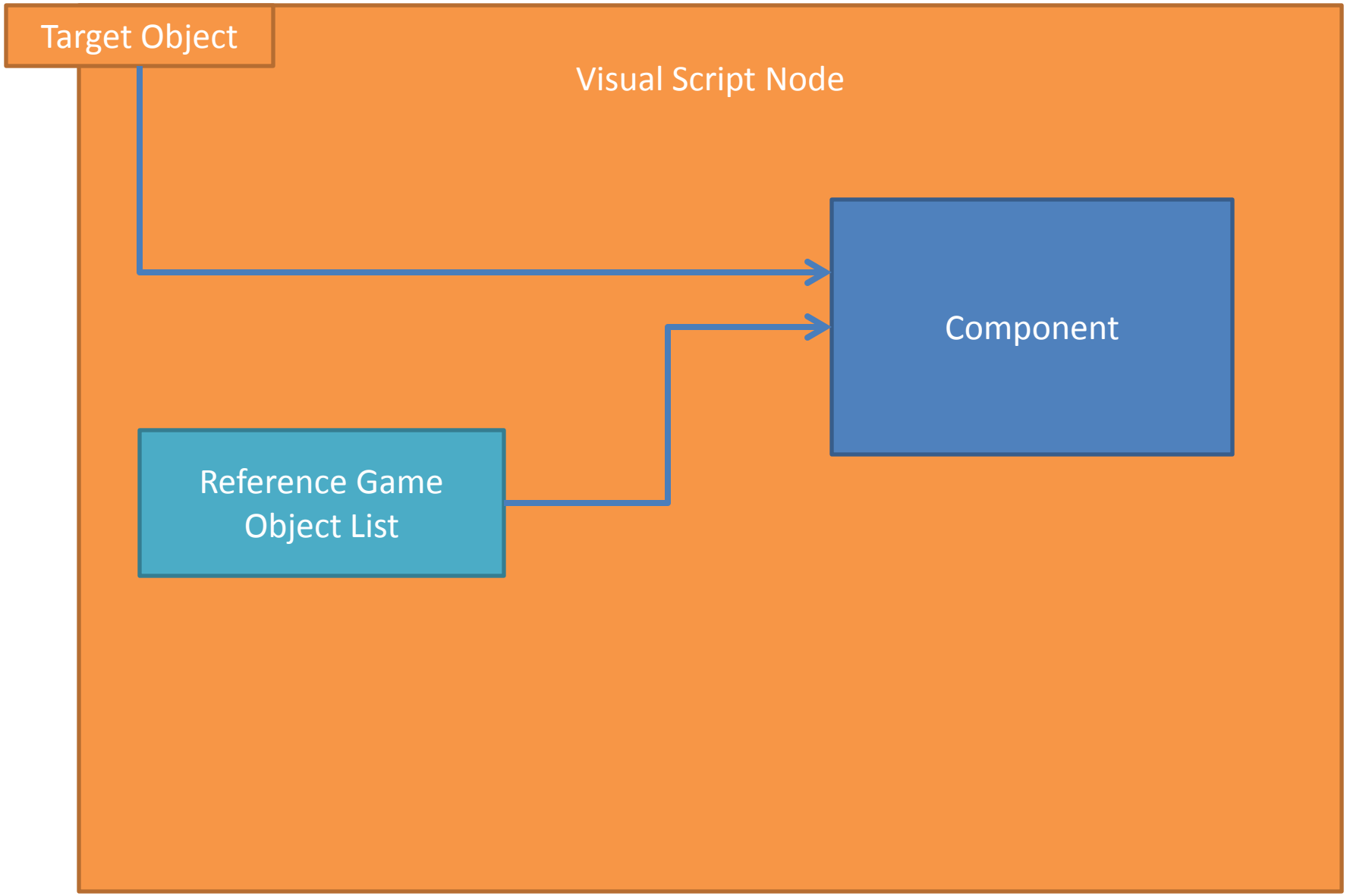

Game Object
Component
Game Object Data Definition
Field
Required Game Object Fields
Required Component Fields
For components that are completely known at compile time (m ese small tables n be pre-
For components whose full definition is not known until runtime (e.g. lua component), these small tables must be created at runtime, but they can be cached if the requirements themselves do not change at runtime.
Additional fiel
Per Unique Tag
Field

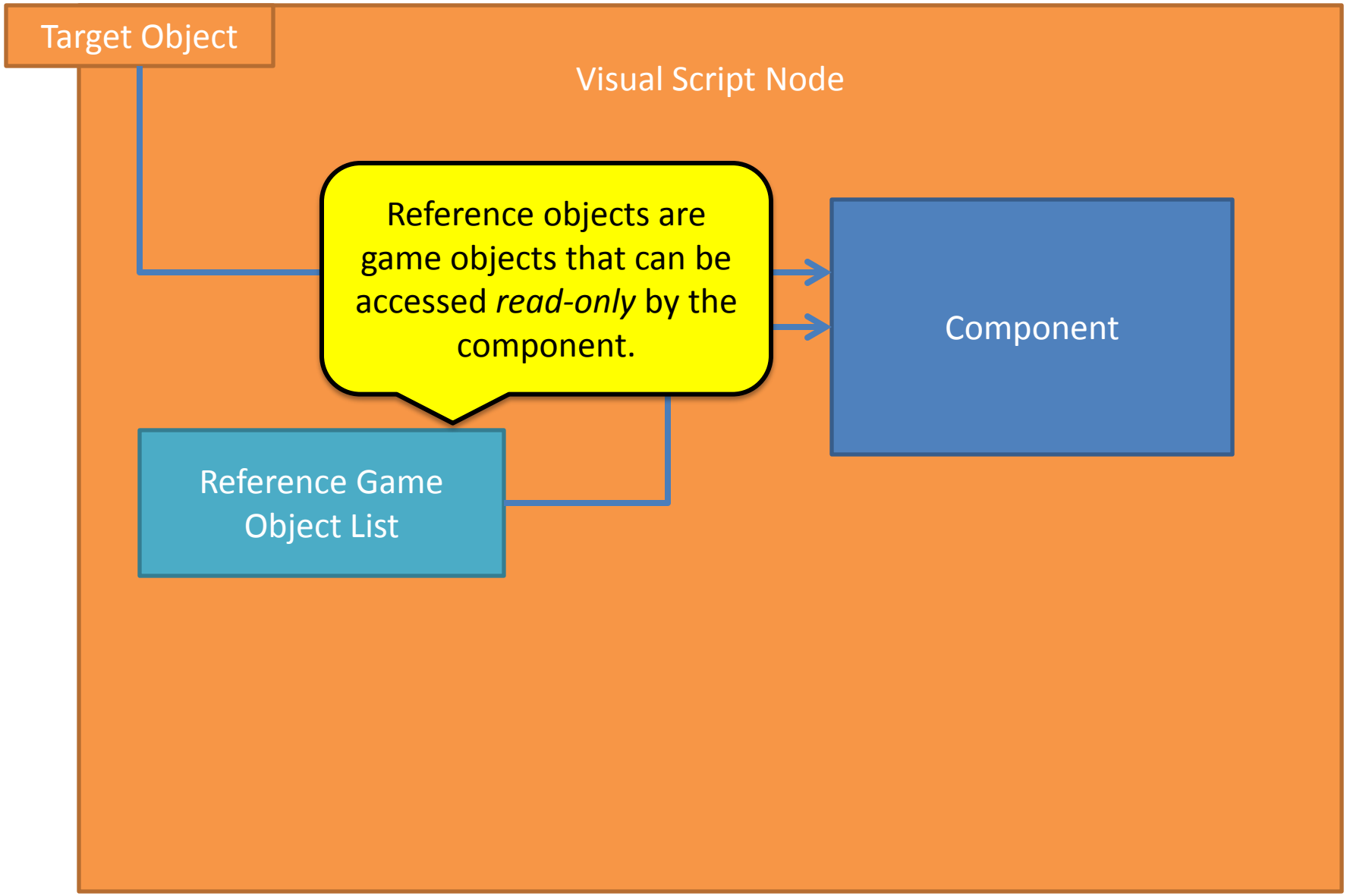

Game Object
Game Object Data Definition
Additional required fields
Unique Tag
Field
Field
Field
Field
Field
Field
Target Object
Visual Script Node
Unique Tag

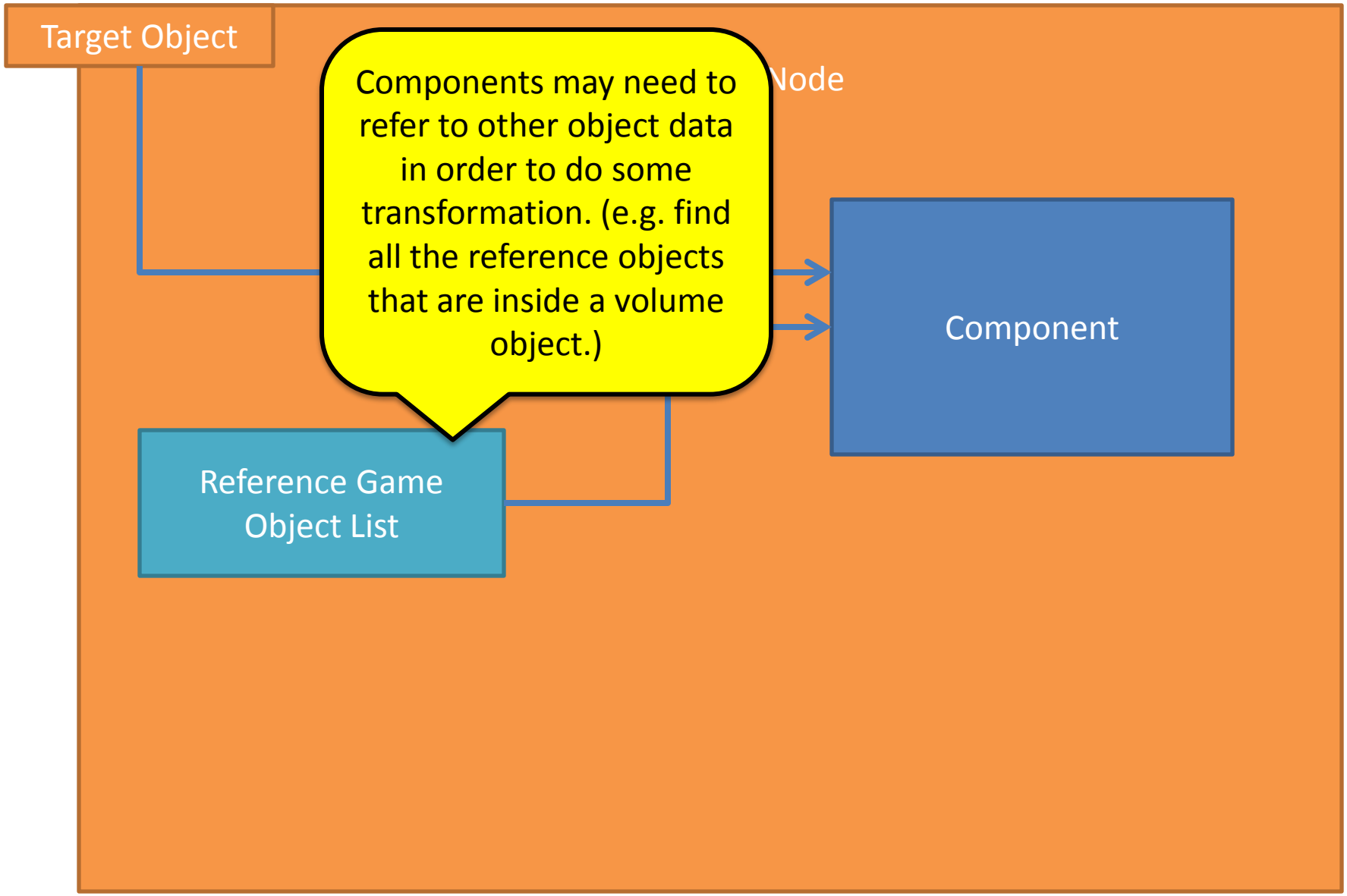

Game Object
Game Object Data Definition
Additional required fields
The visual script node gets the list of additional required fields from each component and tells the target game object to add these fields to its data using the unique tag of the visual script node.
Target Object
Visual Script Node
Unique Tag
Unique Tag
Field
Field
Field
Field
Field
Field

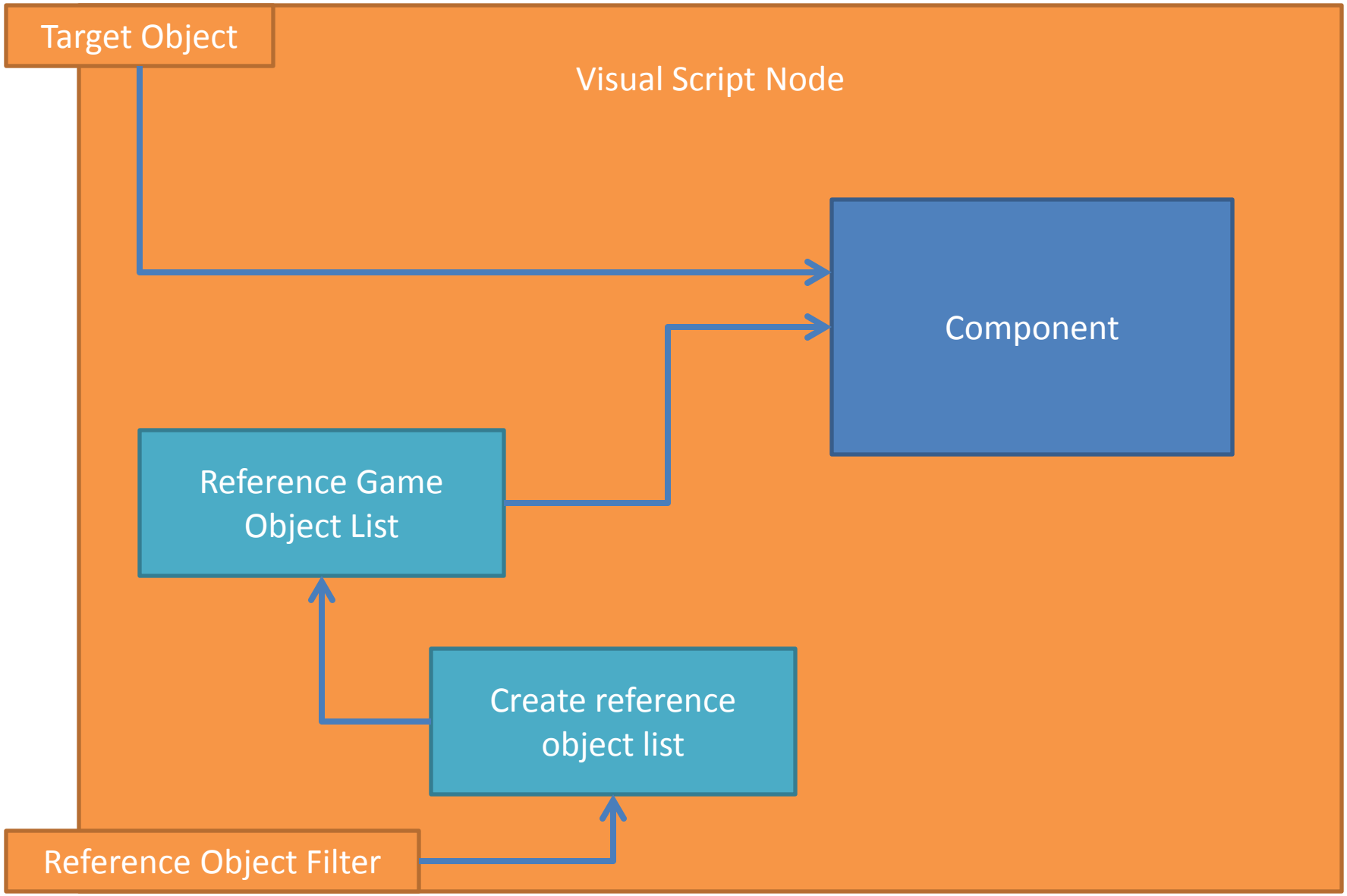

Target Object
Visual Script Node
Component
Component

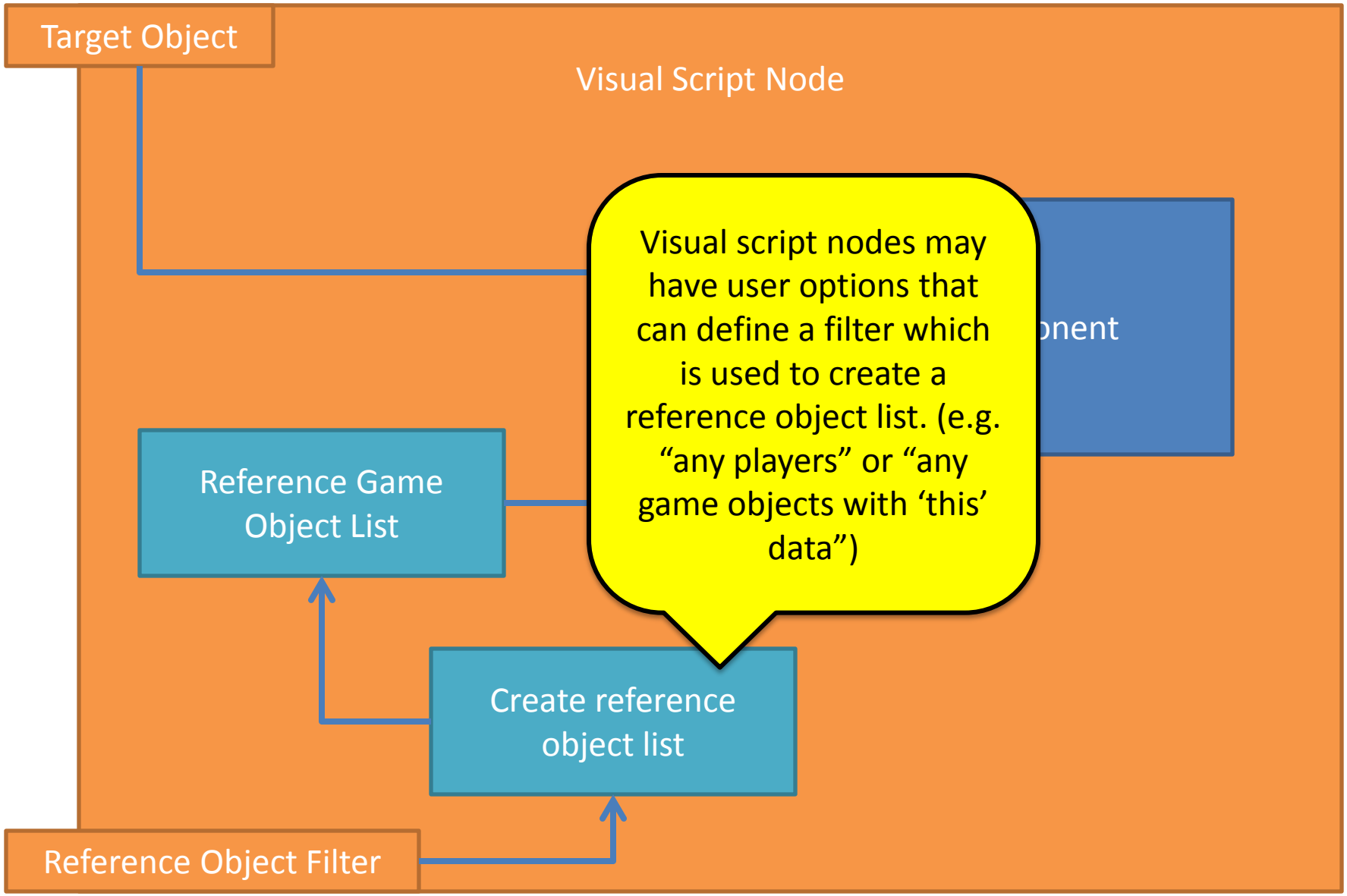

Target Object
Visual Script Node
The target game object is added to each component used by the visual scripting node. The target object is the only data a component can write to.
Component
Component

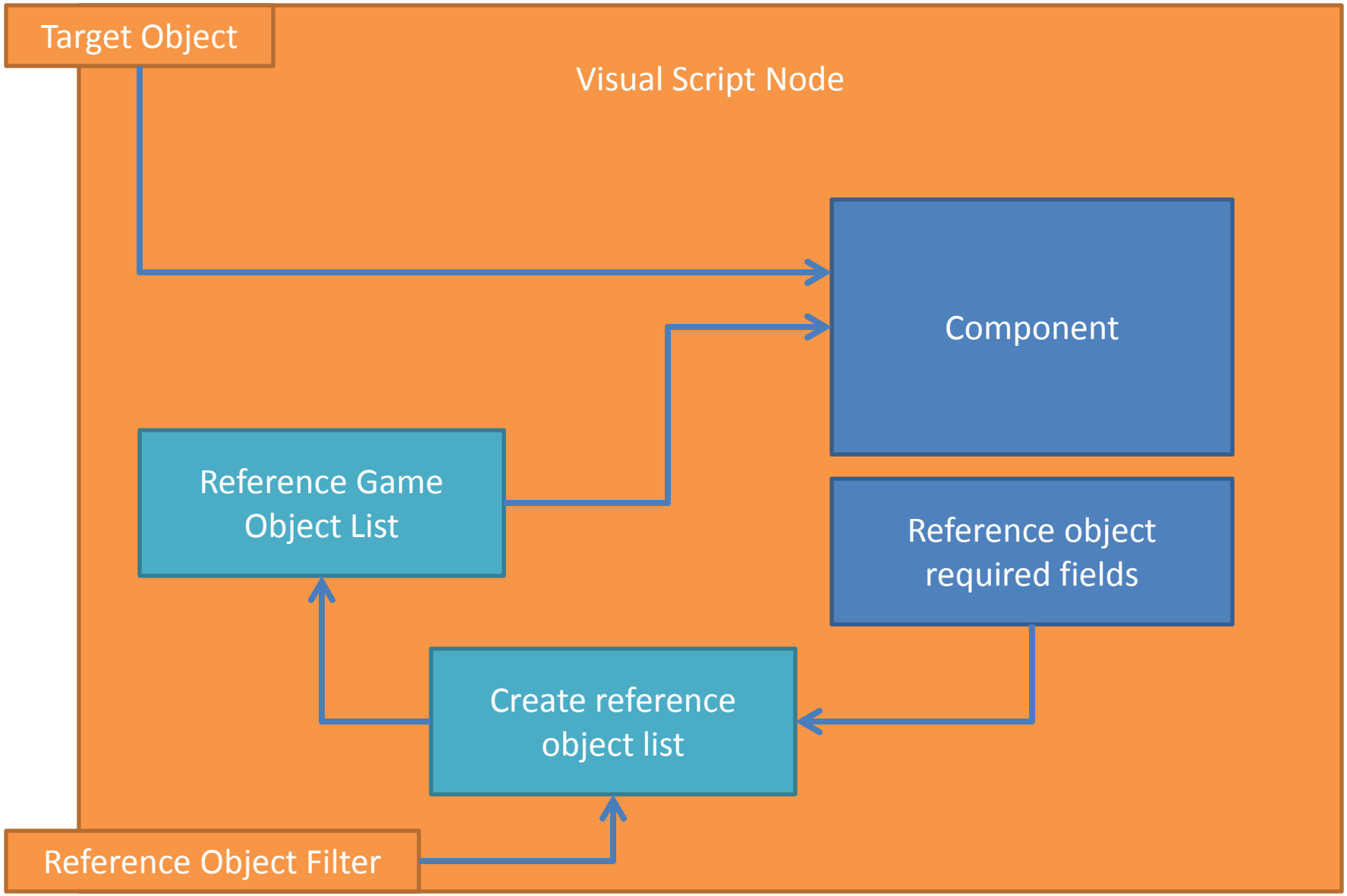

Target Object
Visual Script Node
Component
Reference Game Object List

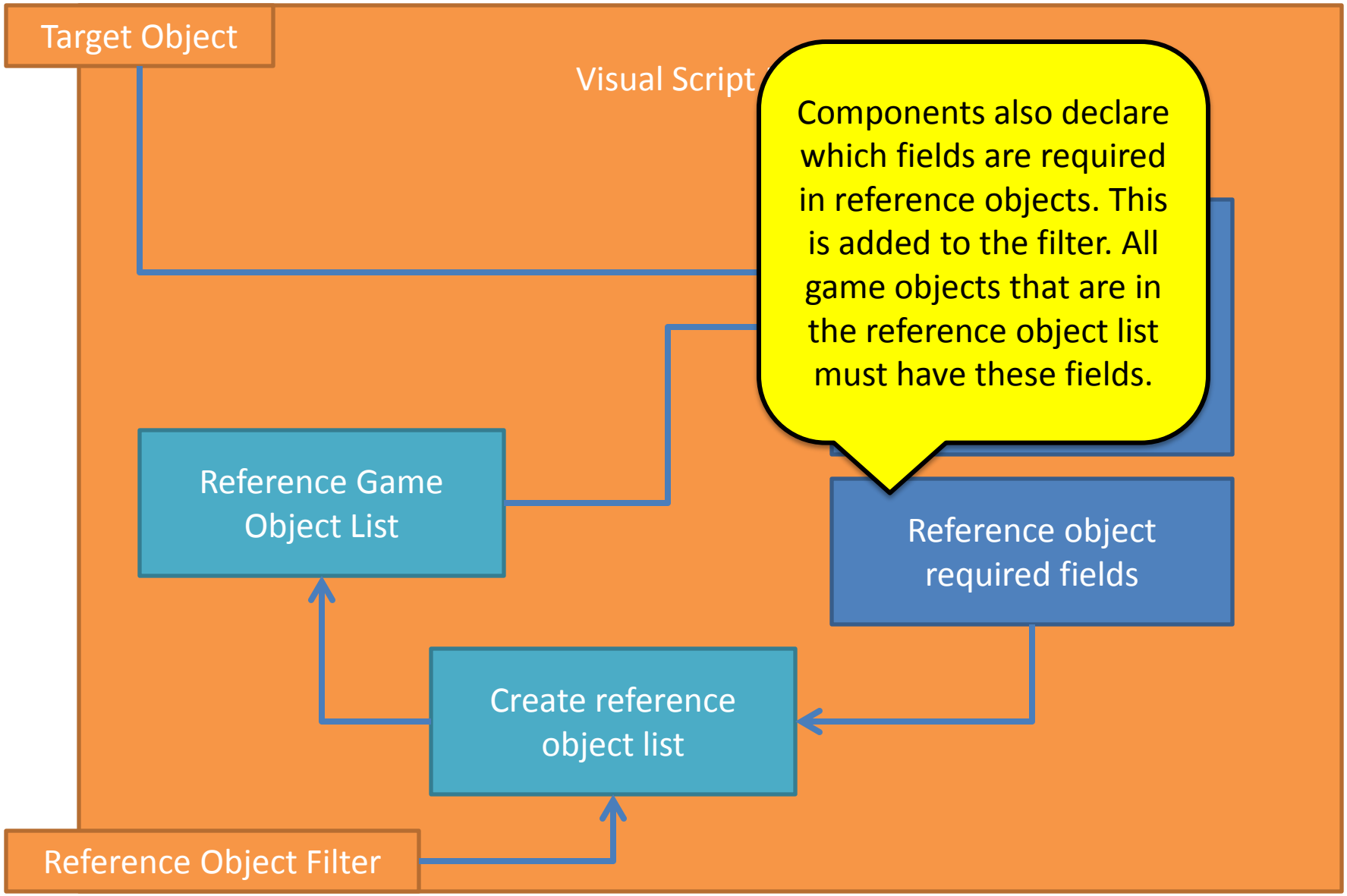

Target Object
Visual Script Node
Reference objects are game objects that can be accessed *read-only* by the component.
Component
Reference Game Object List

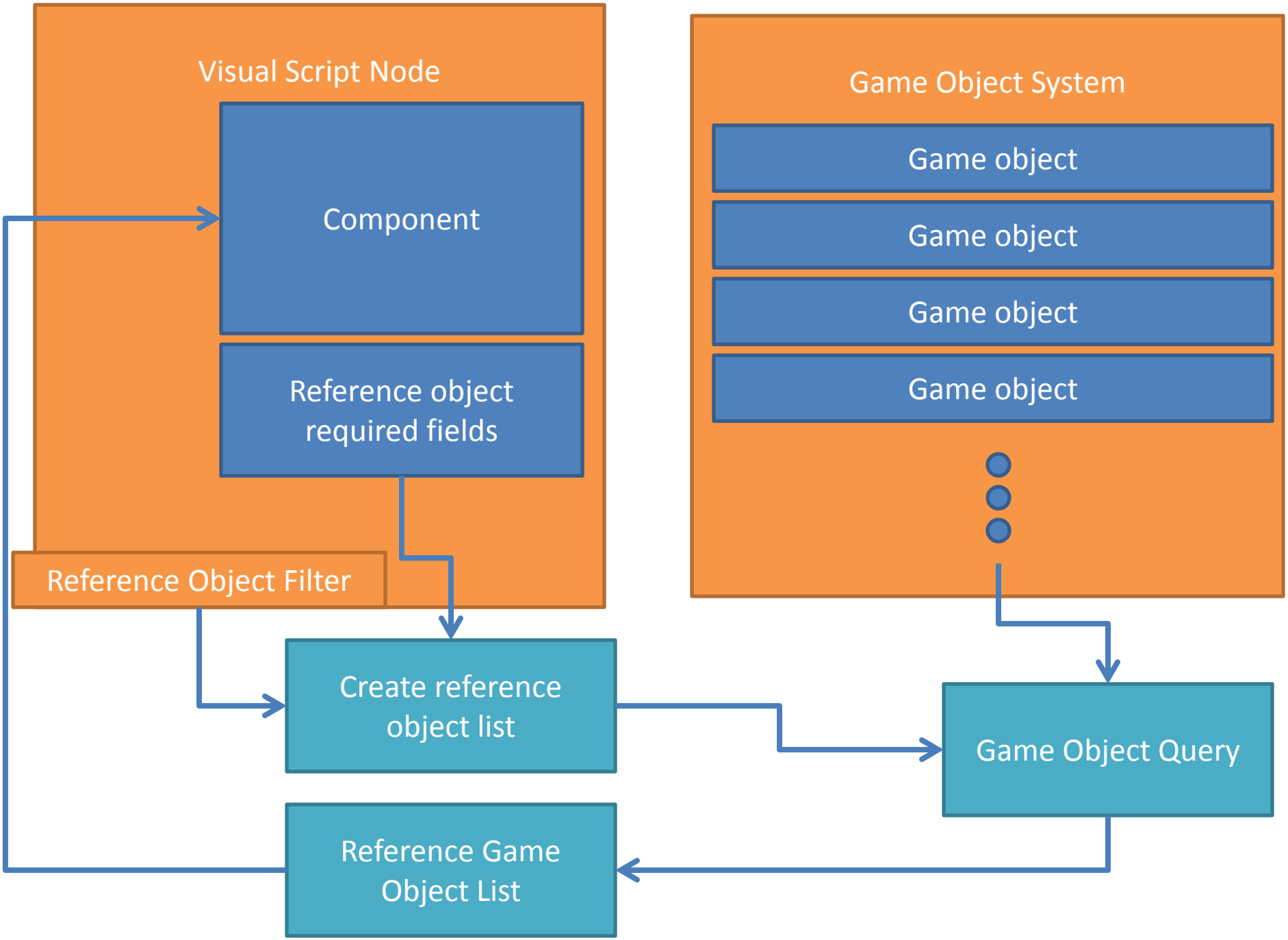

Target Object
Node
Components may need to refer to other object data in order to do some transformation. (e.g. find all the reference objects that are inside a volume object.)
Component
Reference Game Object List

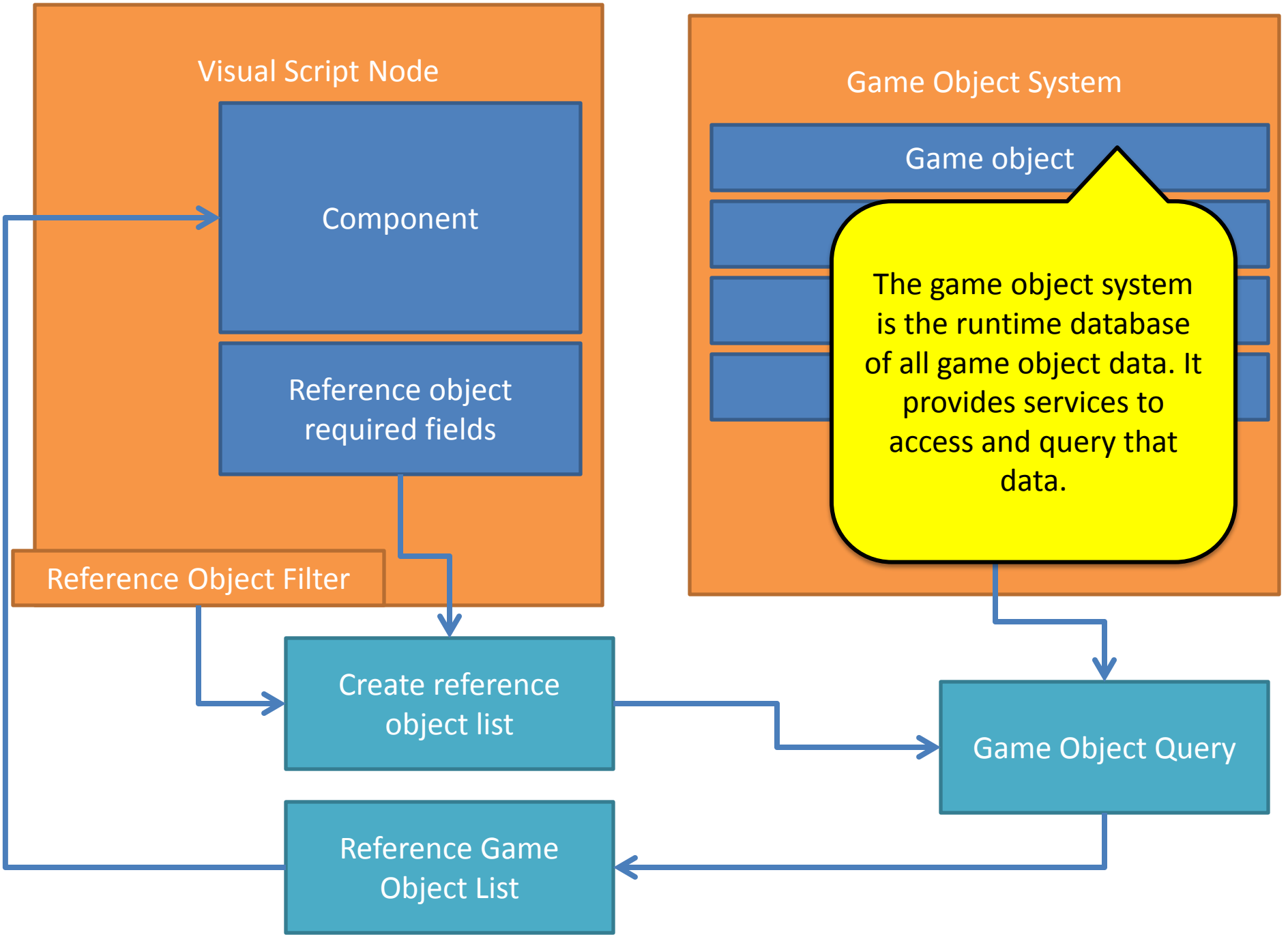

Target Object
Visual Script Node
Component
Reference Game Object List
Create reference object list
Reference Object Filter

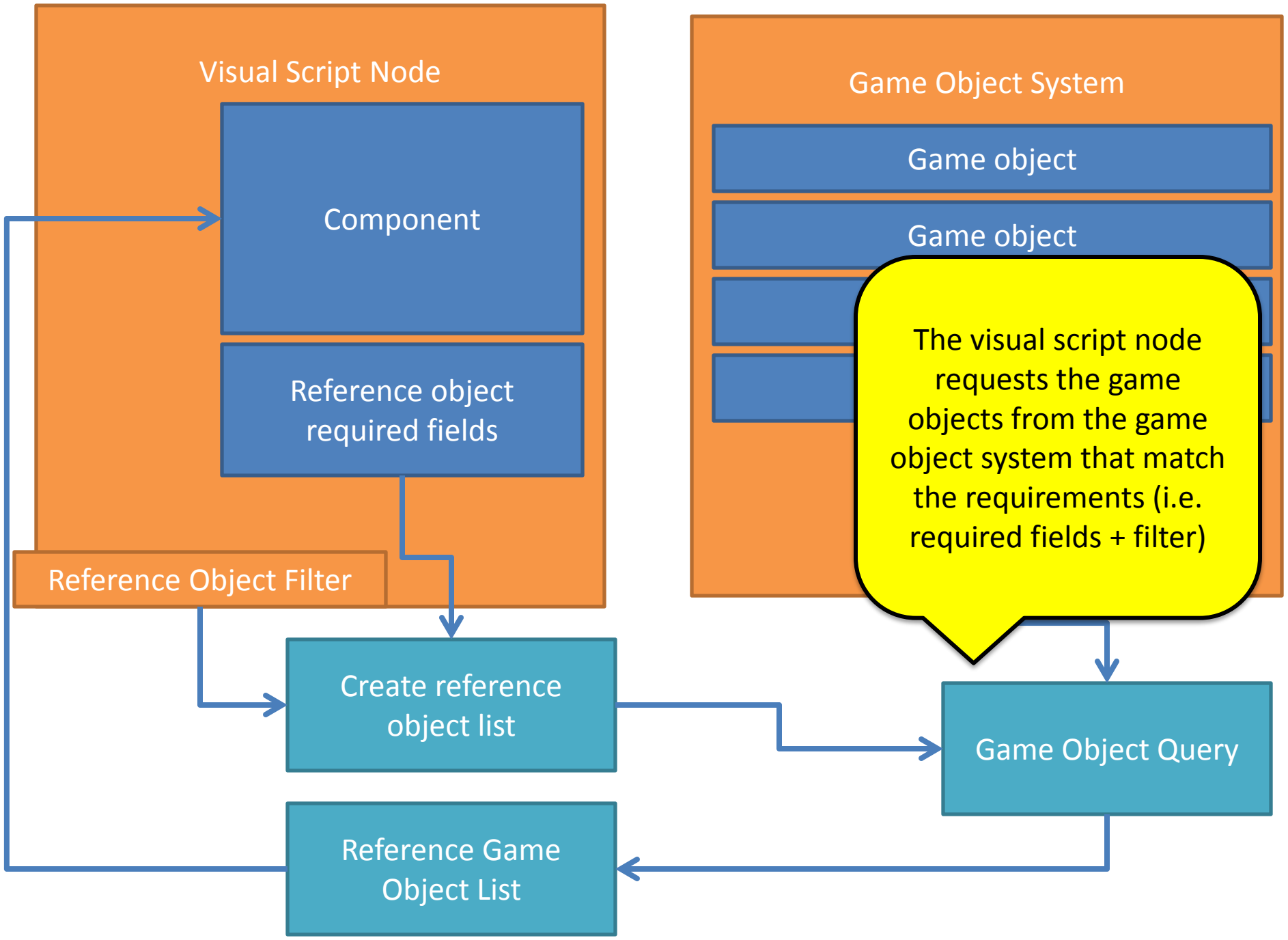

Target Object
Visual Script Node
Visual script nodes may have user options that can define a filter which is used to create a reference object list. (e.g. “any players” or “any game objects with ‘this’ data”)
onent
Reference Game Object List
Create reference object list
Reference Object Filter

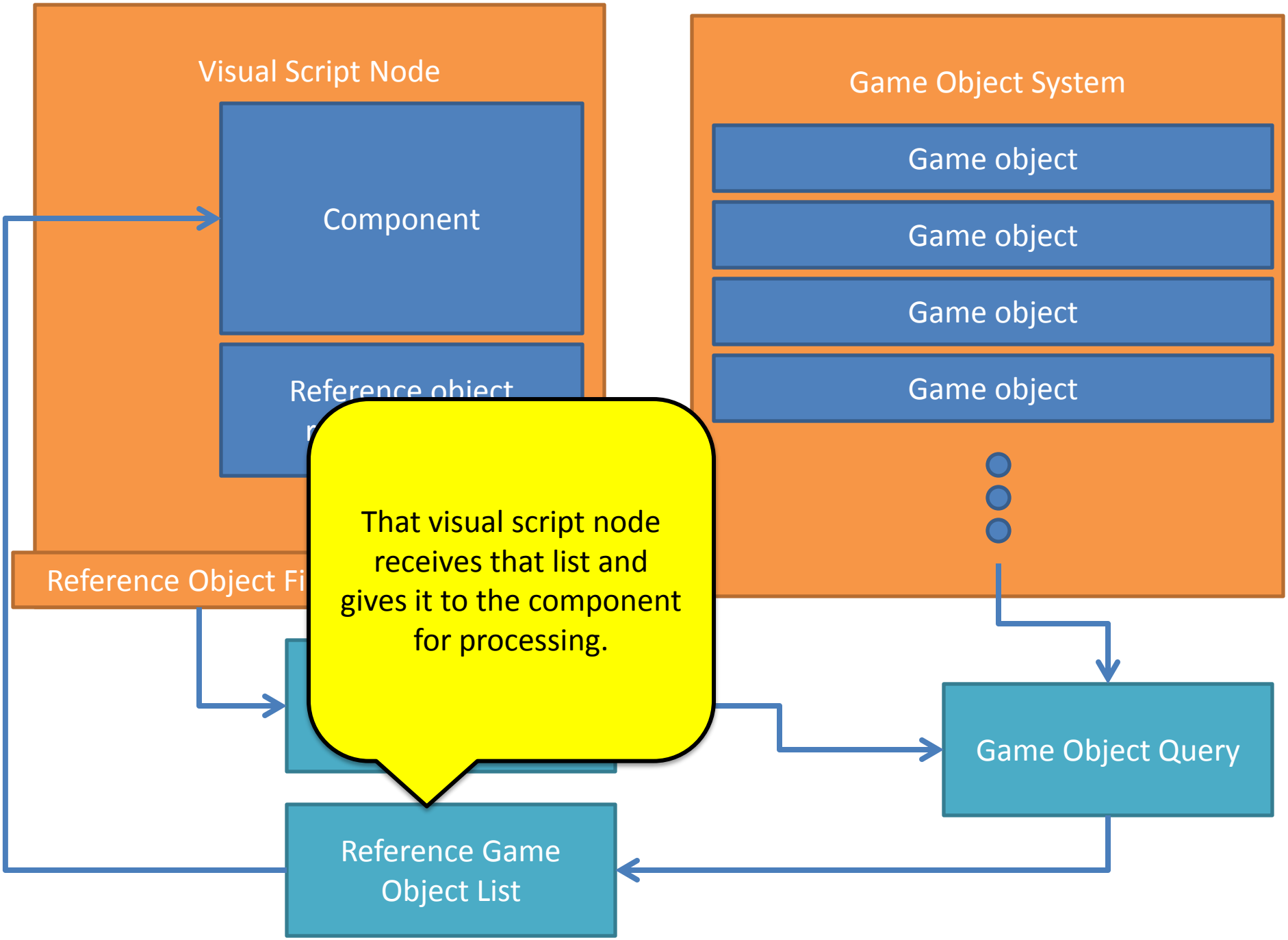

Target Object
Visual Script Node
Component
Reference Game Object List
Reference object required fields
Create reference object list
Reference Object Filter

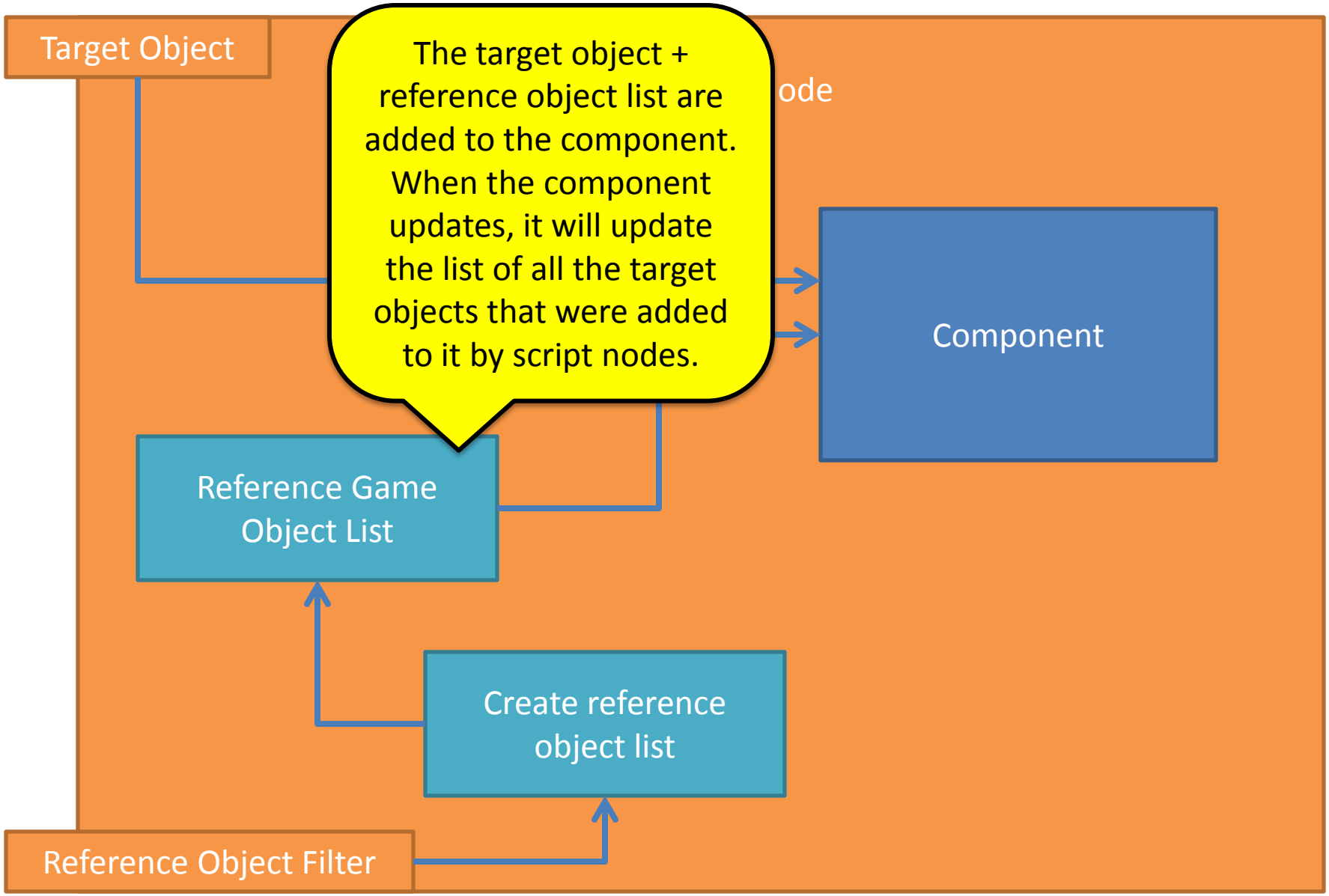

Target Object
Visual Script
Components also declare which fields are required in reference objects. This is added to the filter. All game objects that are in the reference object list must have these fields.
Reference Game Object List
Reference object required fields
Create reference object list
Reference Object Filter

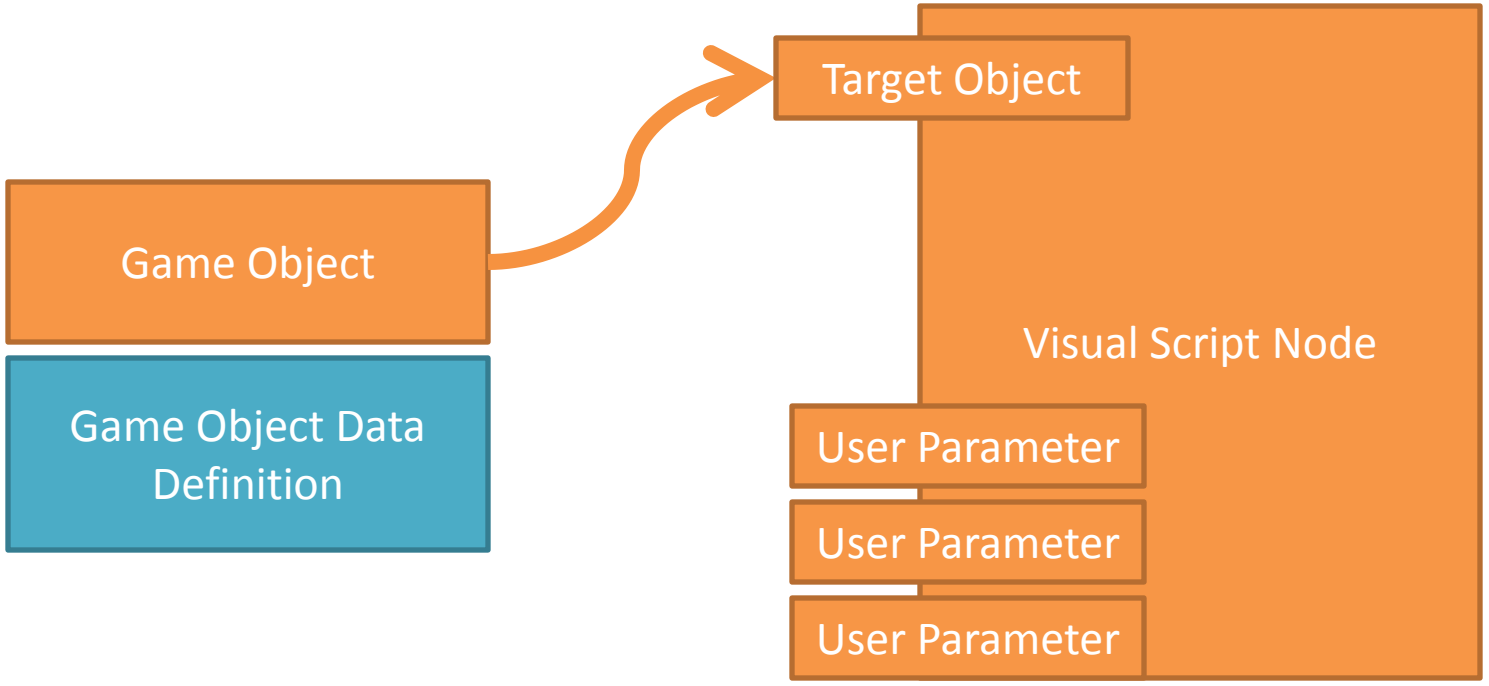

Visual Script Node
Component
Reference object required fields
Reference Object Filter
Game Object System
Game object
Game object
Game object
Game object
Create reference object list
Game Object Query
Reference Game Object List

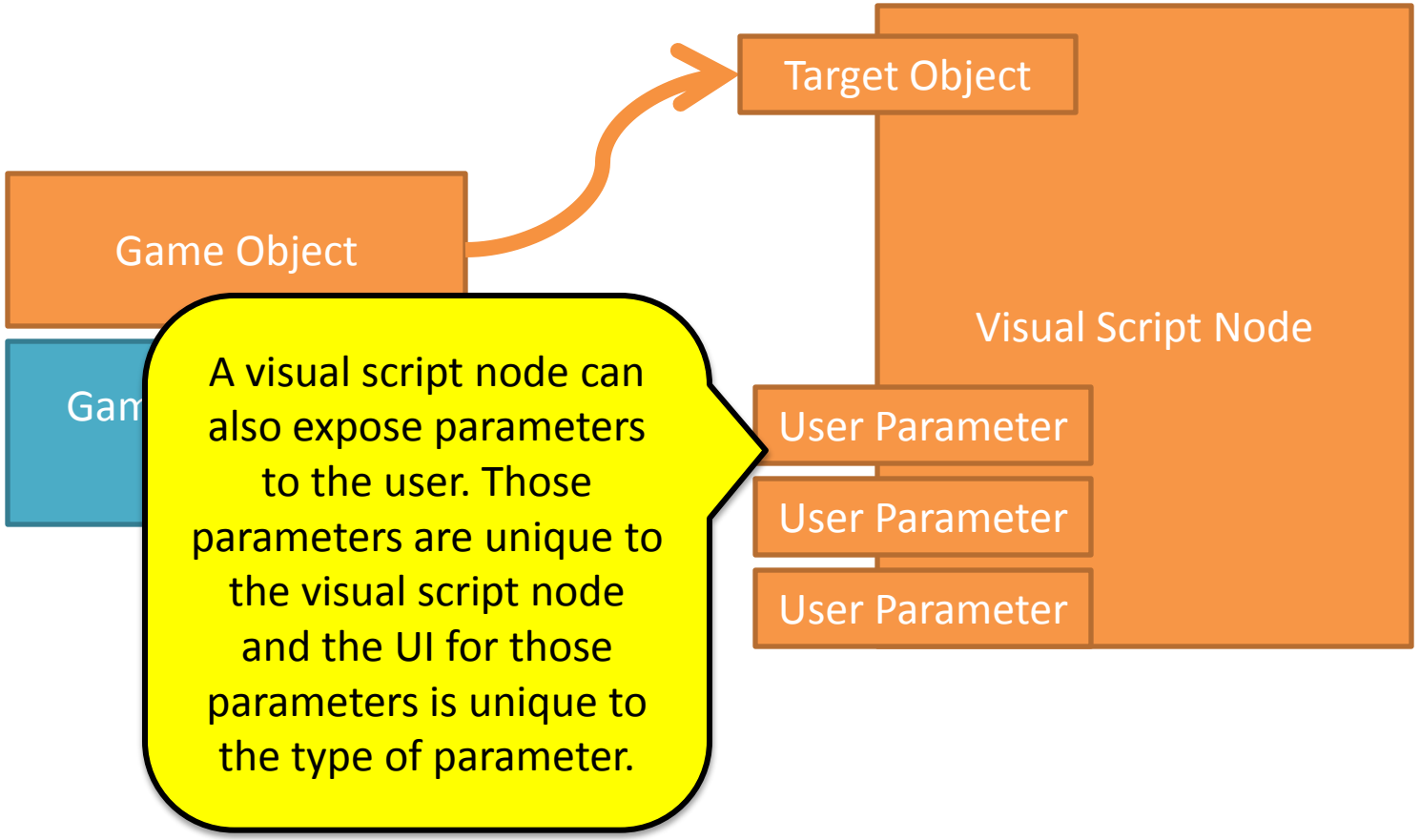

Visual Script Node
Component
Reference object required fields
Reference Object Filter
Game Object System
Game object
The game object system is the runtime database of all game object data. It provides services to access and query that data.
Create reference object list
Game Object Query
Reference Game Object List

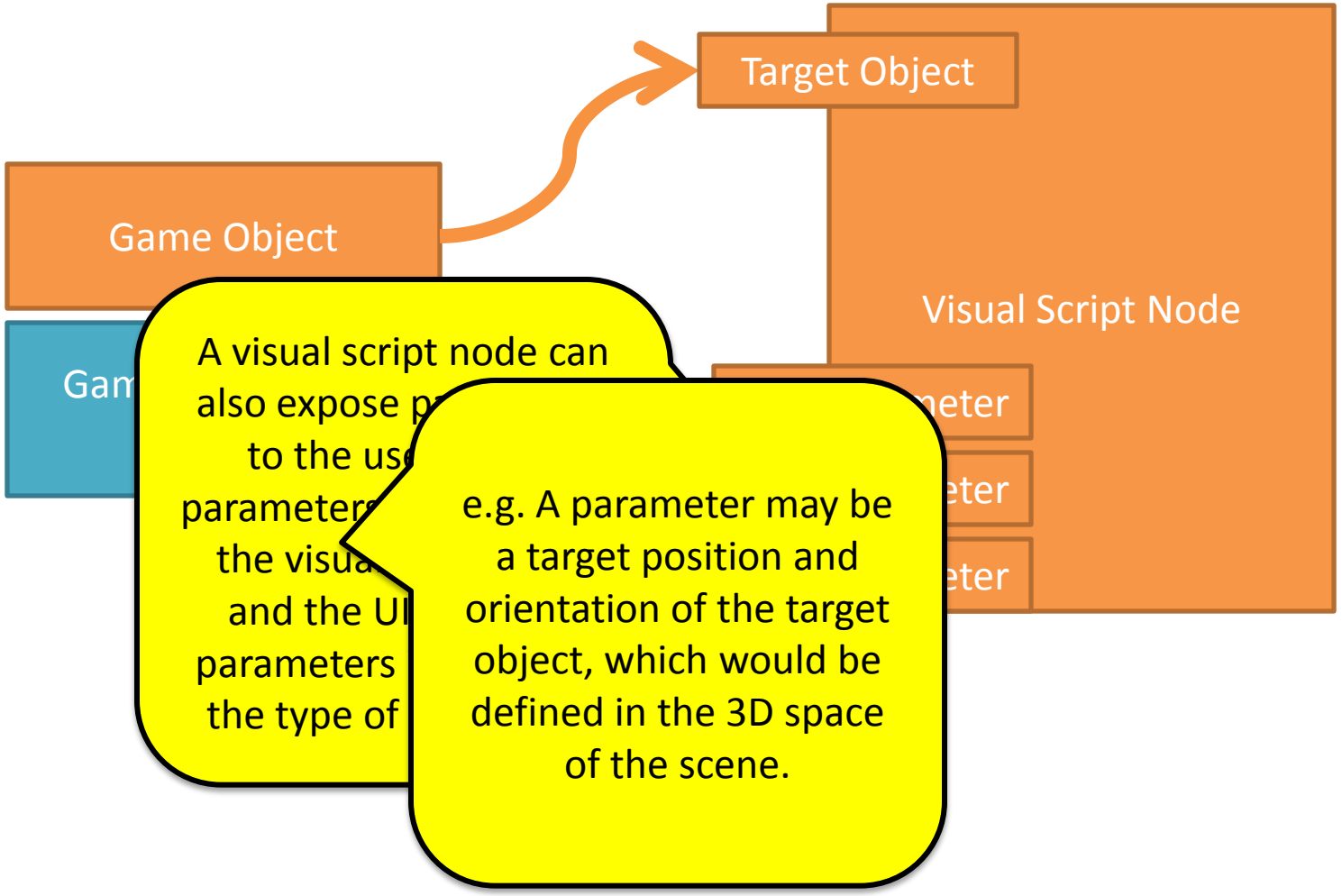

Visual Script Node
Component
Reference object required fields
Reference Object Filter
Game Object System
Game object
Game object
The visual script node requests the game objects from the game object system that match the requirements (i.e. required fields + filter)
Create reference object list
Game Object Query
Reference Game Object List

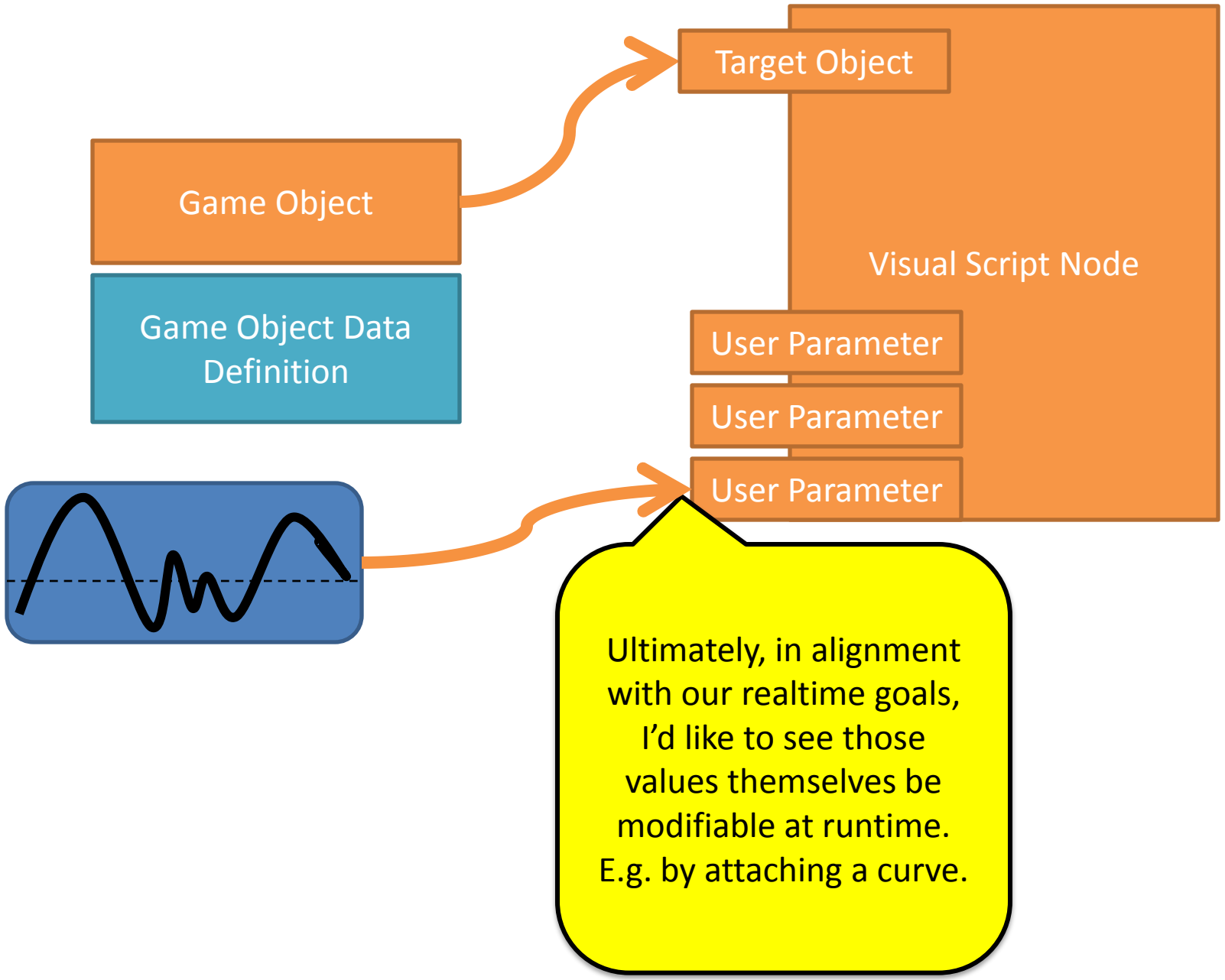

Visual Script Node
Component
Reference object
Reference Object Fi
Game Object System
Game object
Game object
Game object
Game object
That visual script node receives that list and gives it to the component for processing.
Game Object Query
Reference Game Object List

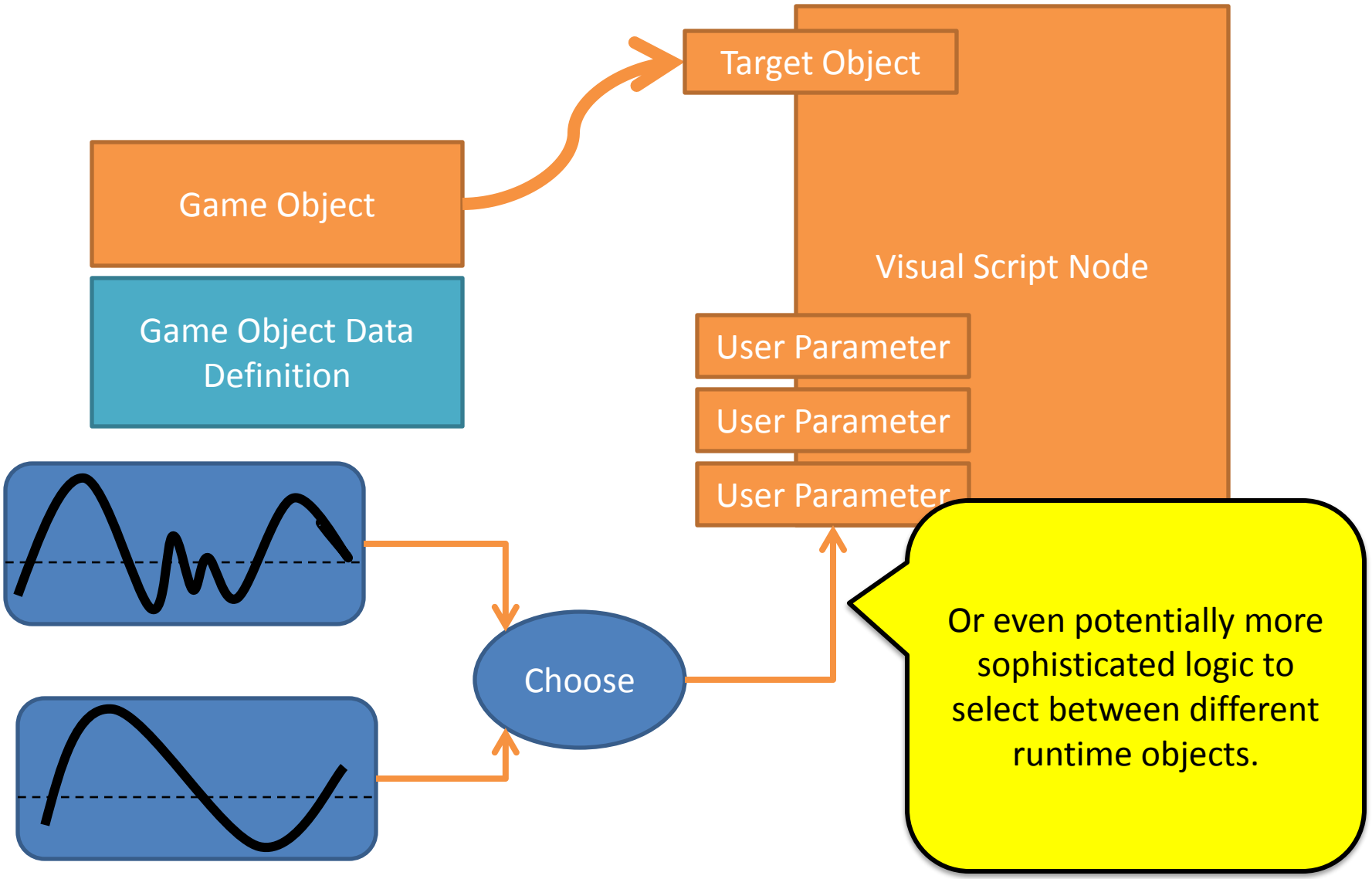

Target Object
The target object + reference object list are added to the component. When the component updates, it will update the list of all the target objects that were added to it by script nodes.
ode
Component
Reference Game Object List
Create reference object list
Reference Object Filter

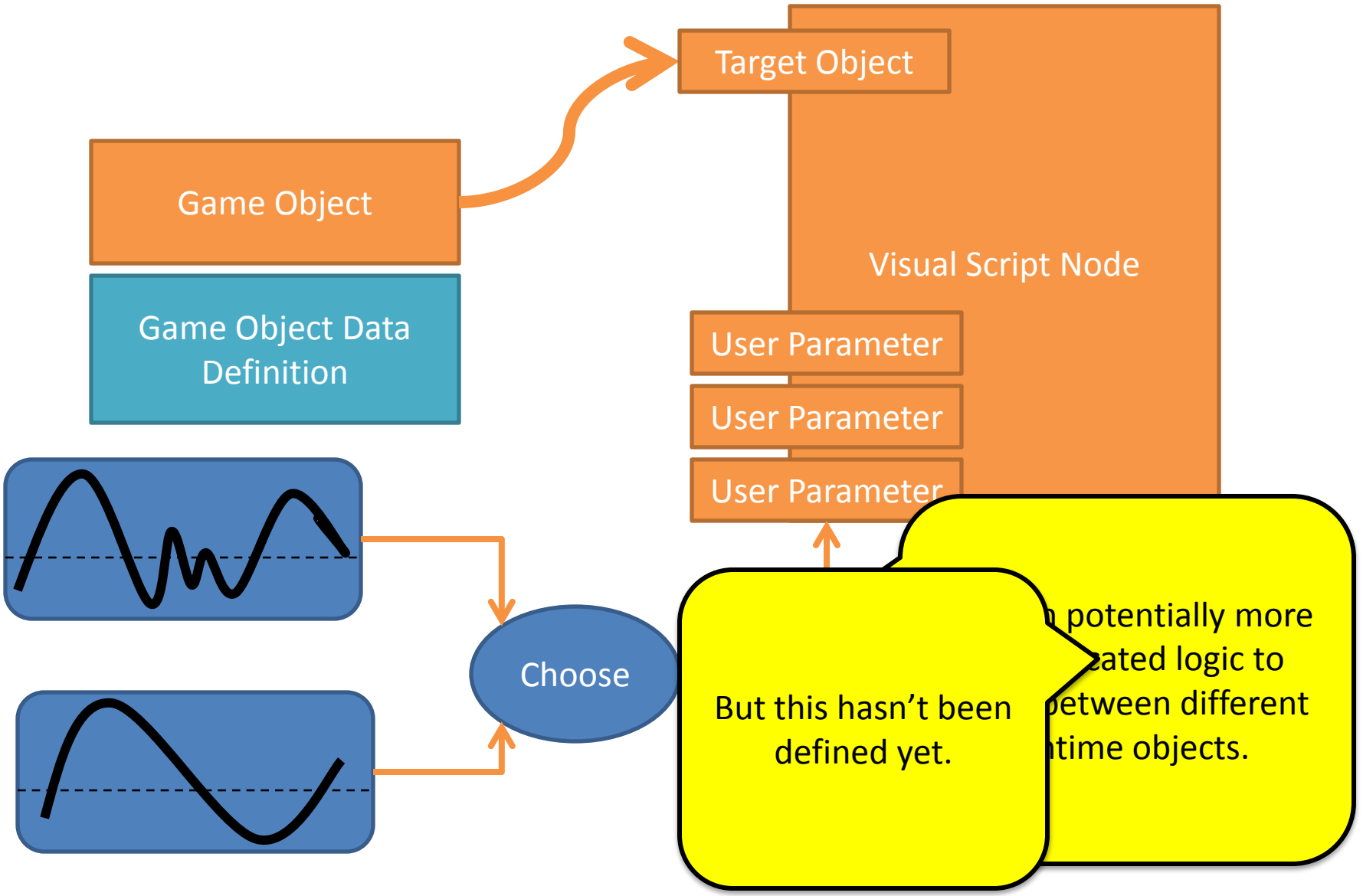

Target Object
Game Object
Visual Script Node
Game Object Data Definition
User Parameter
User Parameter
User Parameter

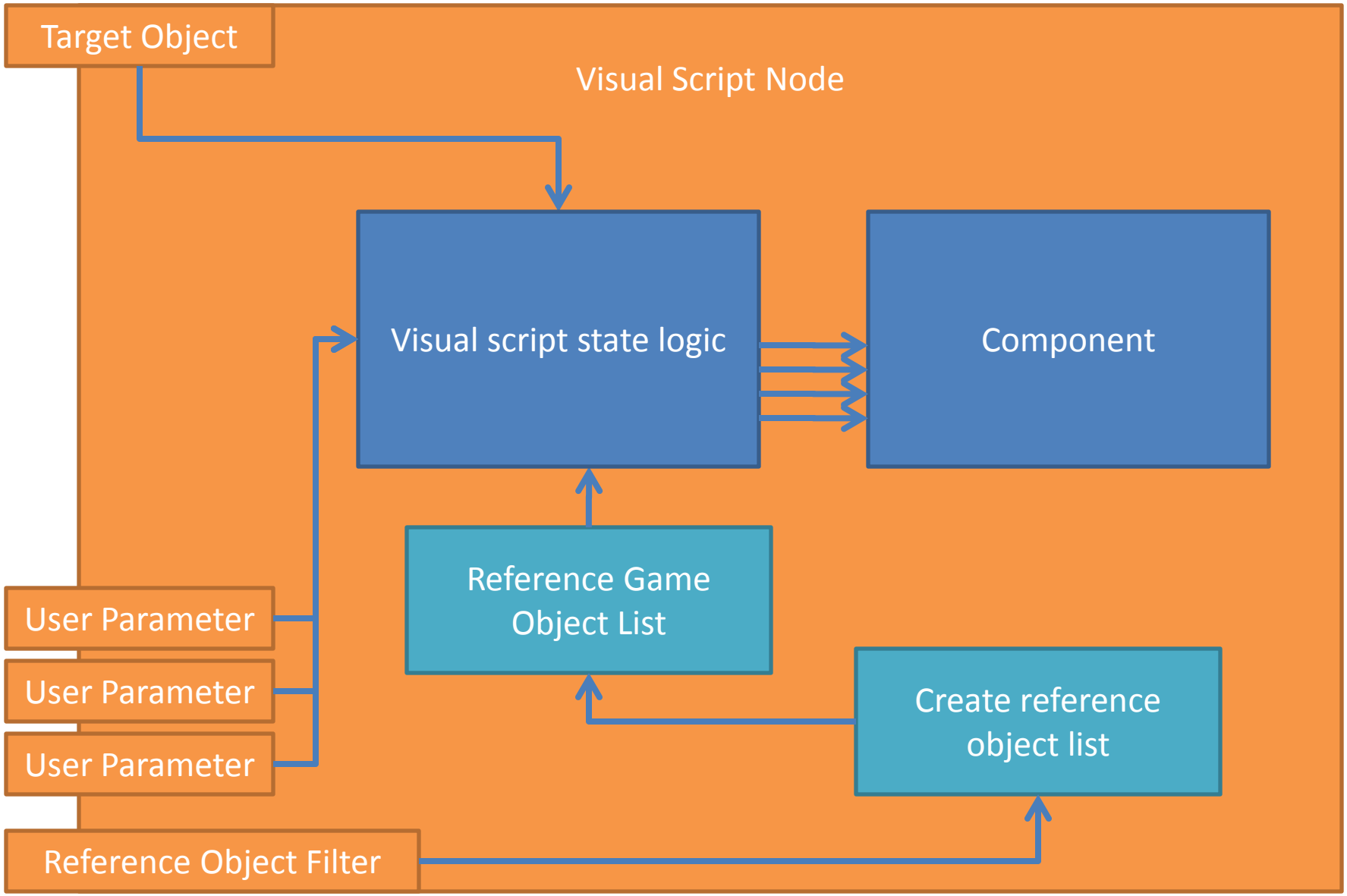

Target Object
Game Object
Visual Script Node
Gam
A visual script node can also expose parameters to the user. Those parameters are unique to the visual script node and the UI for those parameters is unique to the type of parameter.
User Parameter
User Parameter
User Parameter

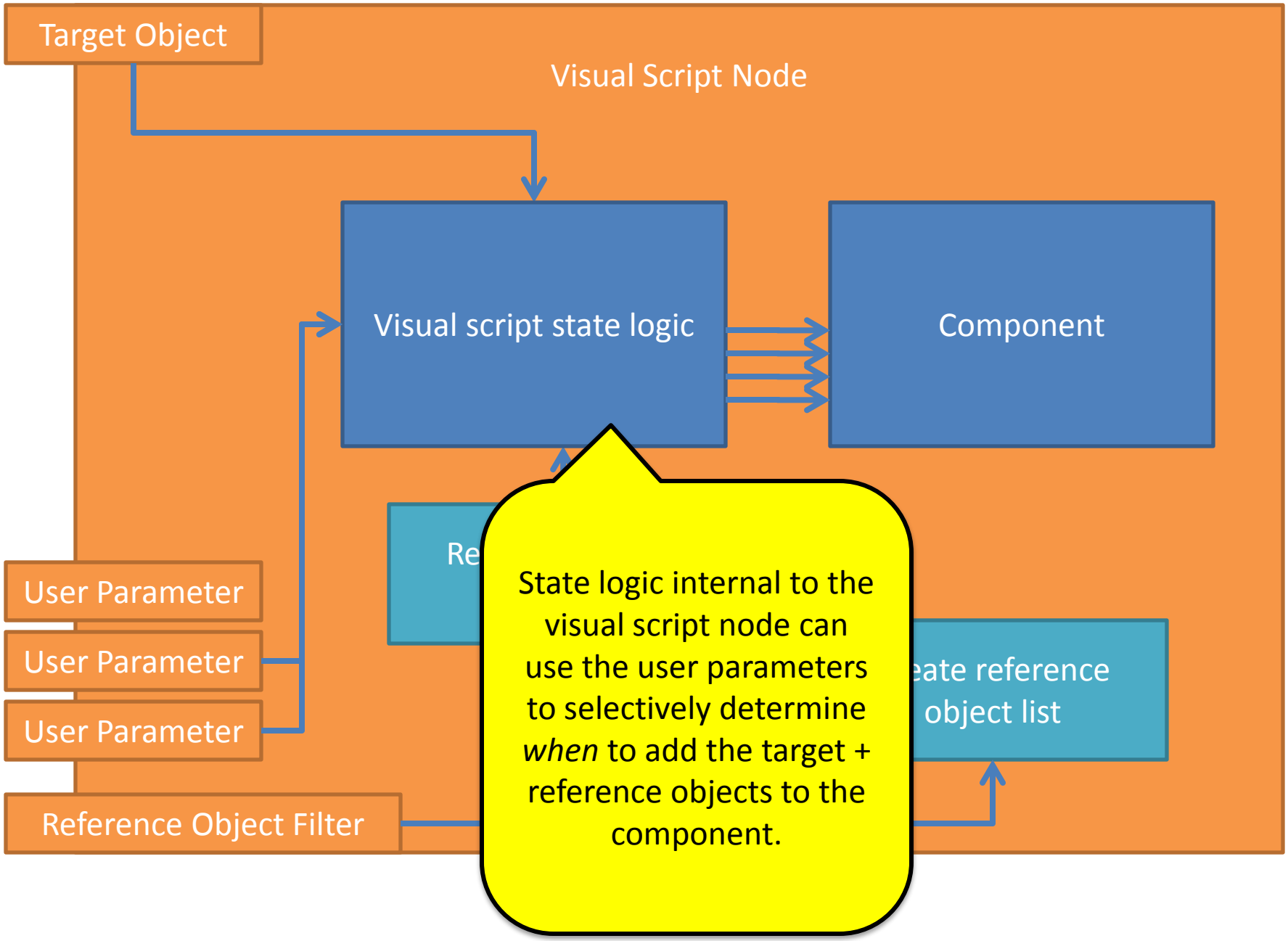

Target Object
Game Object
Visual Script Node
A visual script node can
to the
parameter
the visua
and the UI
parameters
the type of
e.g. A parameter may be a target position and orientation of the target object, which would be defined in the 3D space of the scene.

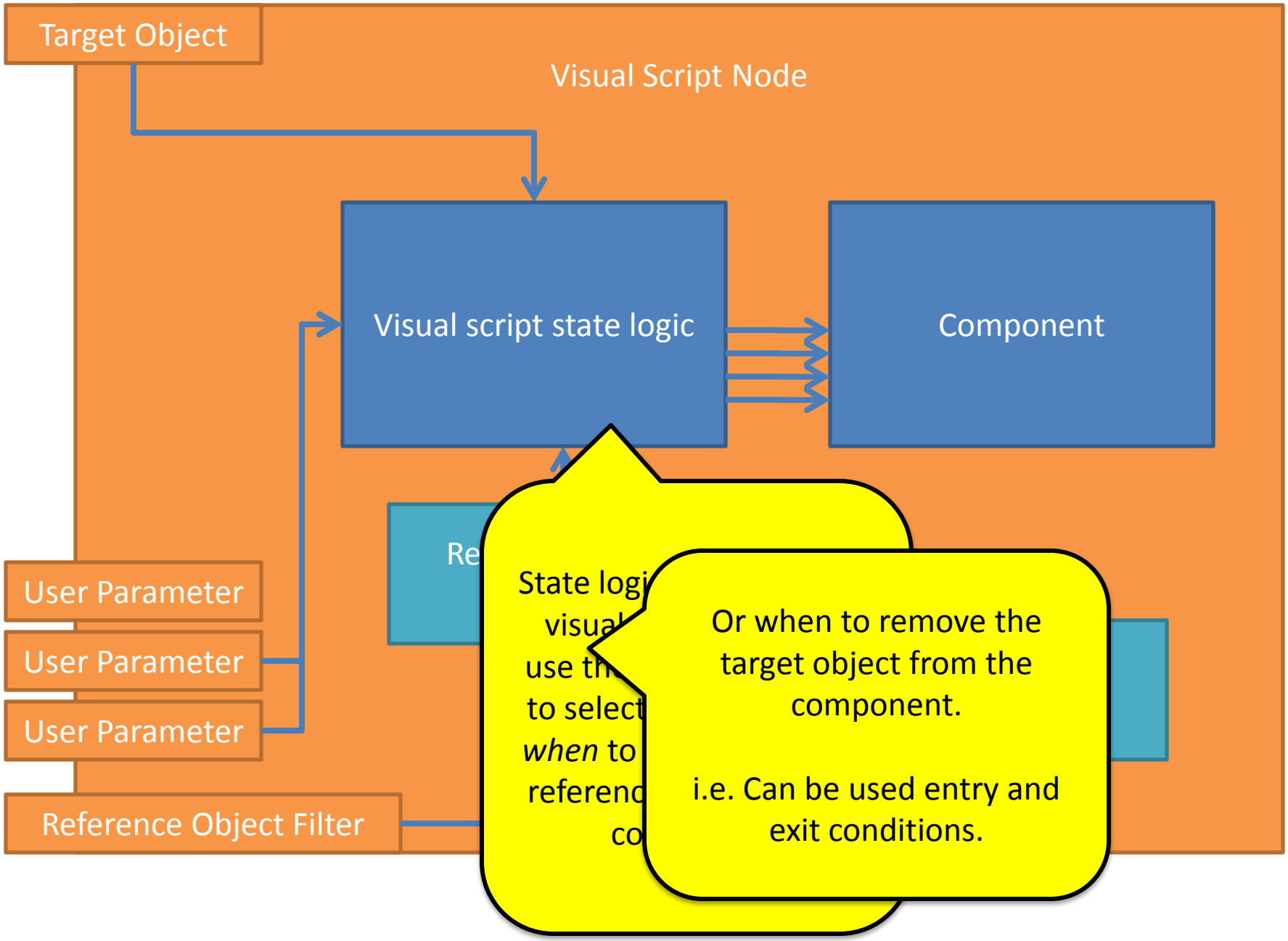

Target Object
Game Object
Visual Script Node
Game Object Data Definition
User Parameter
User Parameter
User Parameter
Ultimately, in alignment with our realtime goals, I'd like to see those values themselves be modifiable at runtime. E.g. by attaching a curve.

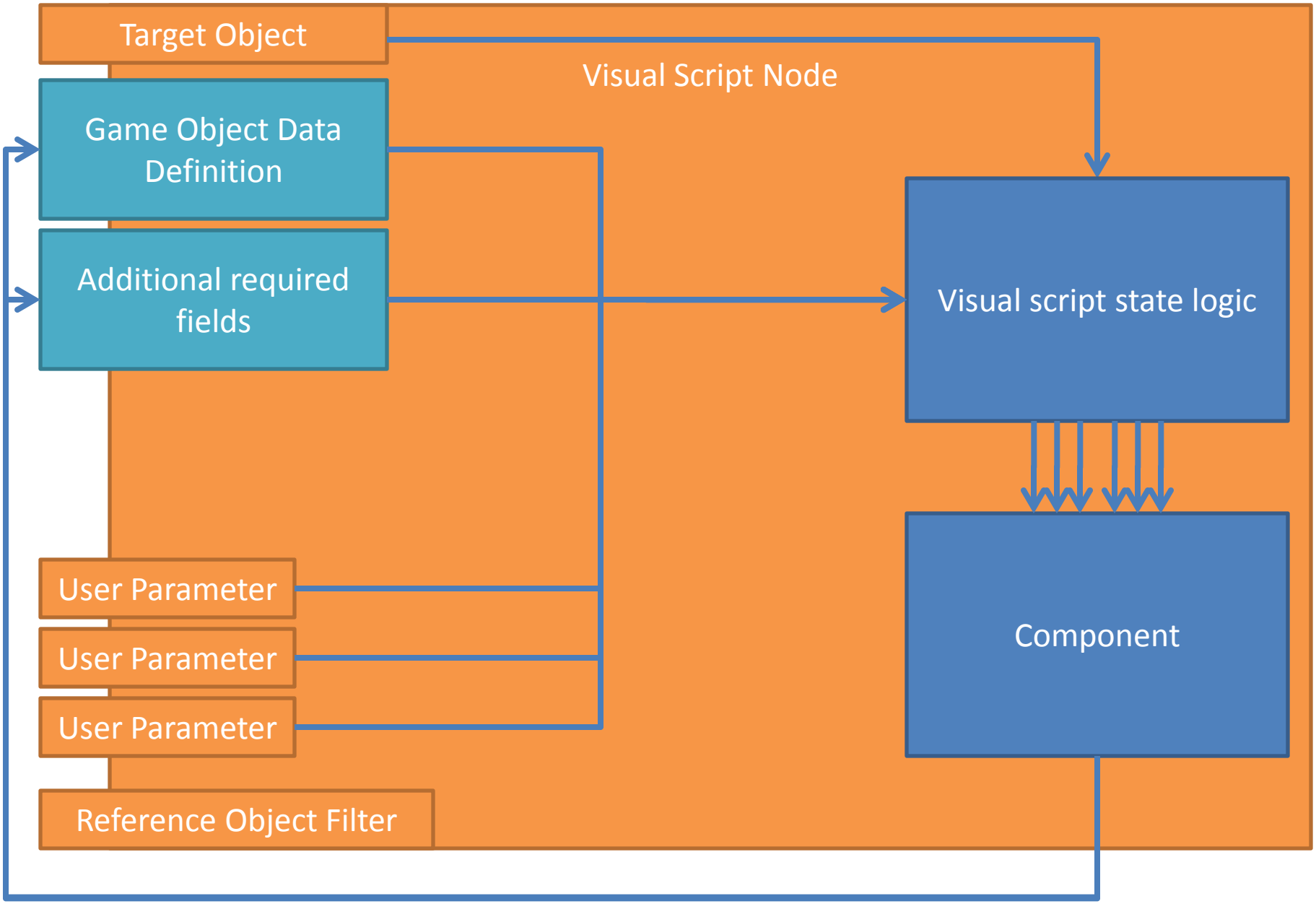

Target Object
Game Object
Visual Script Node
Game Object Data Definition
User Parameter
User Parameter
User Parameter
Choose
Or even potentially more sophisticated logic to select between different runtime objects.

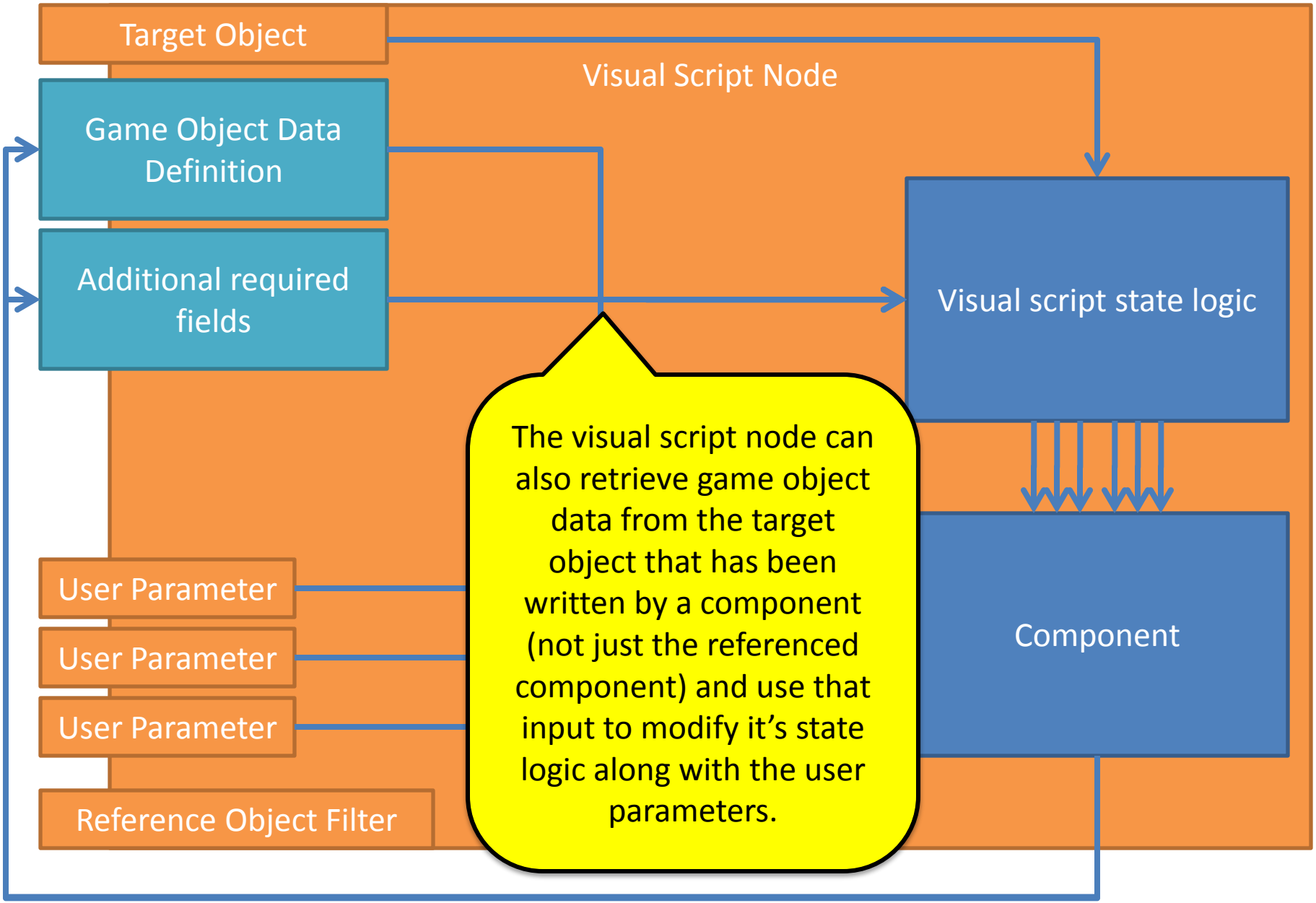

Target Object
Game Object
Visual Script Node
Game Object Data Definition
User Parameter
User Parameter
User Parameter
Choose
But this hasn't been defined yet.
potentially more
cated logic to
between different
time objects.

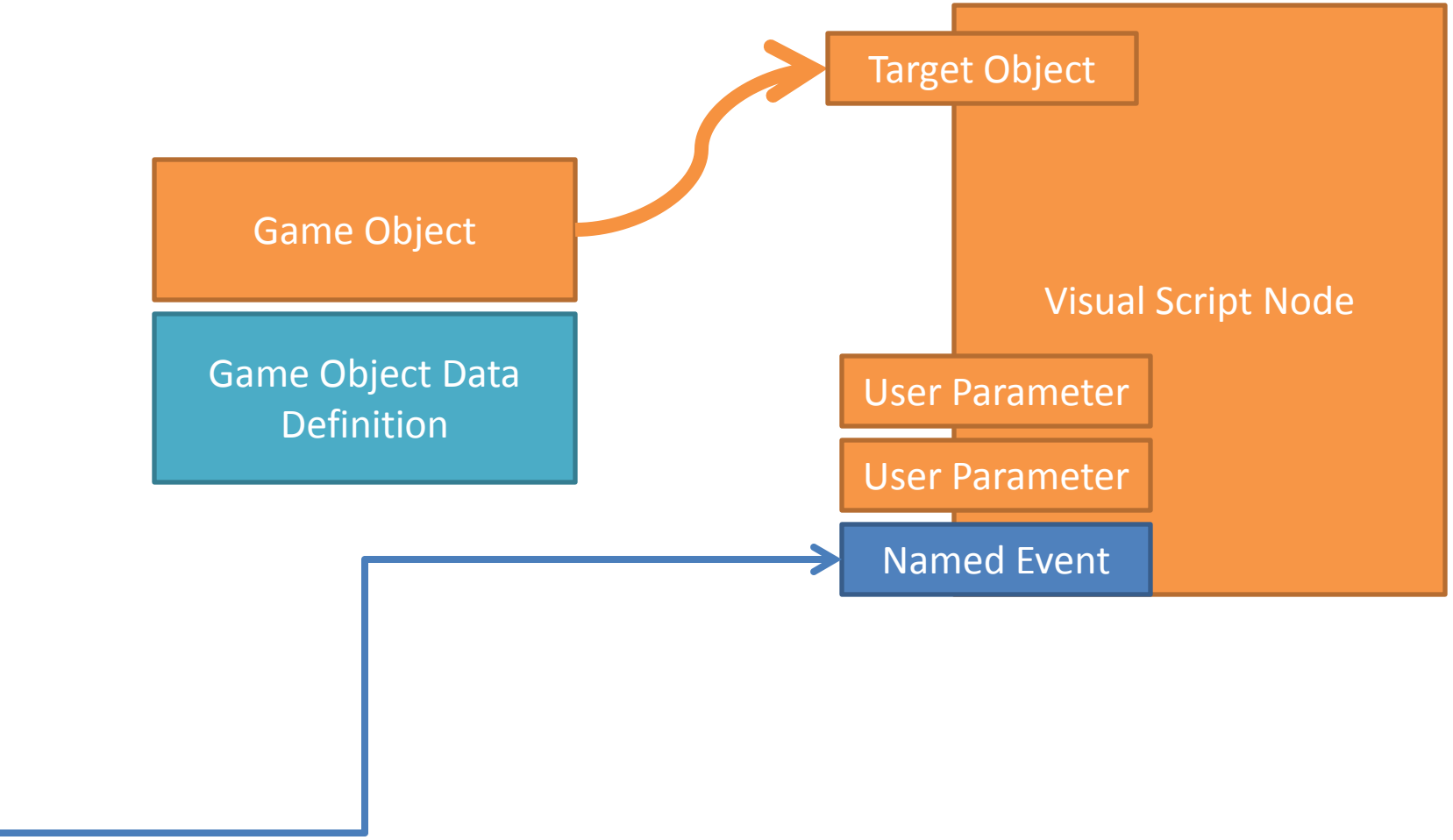

Target Object
Visual Script Node
Visual script state logic
Component
Reference Game Object List
User Parameter
User Parameter
User Parameter
Create reference object list
Reference Object Filter

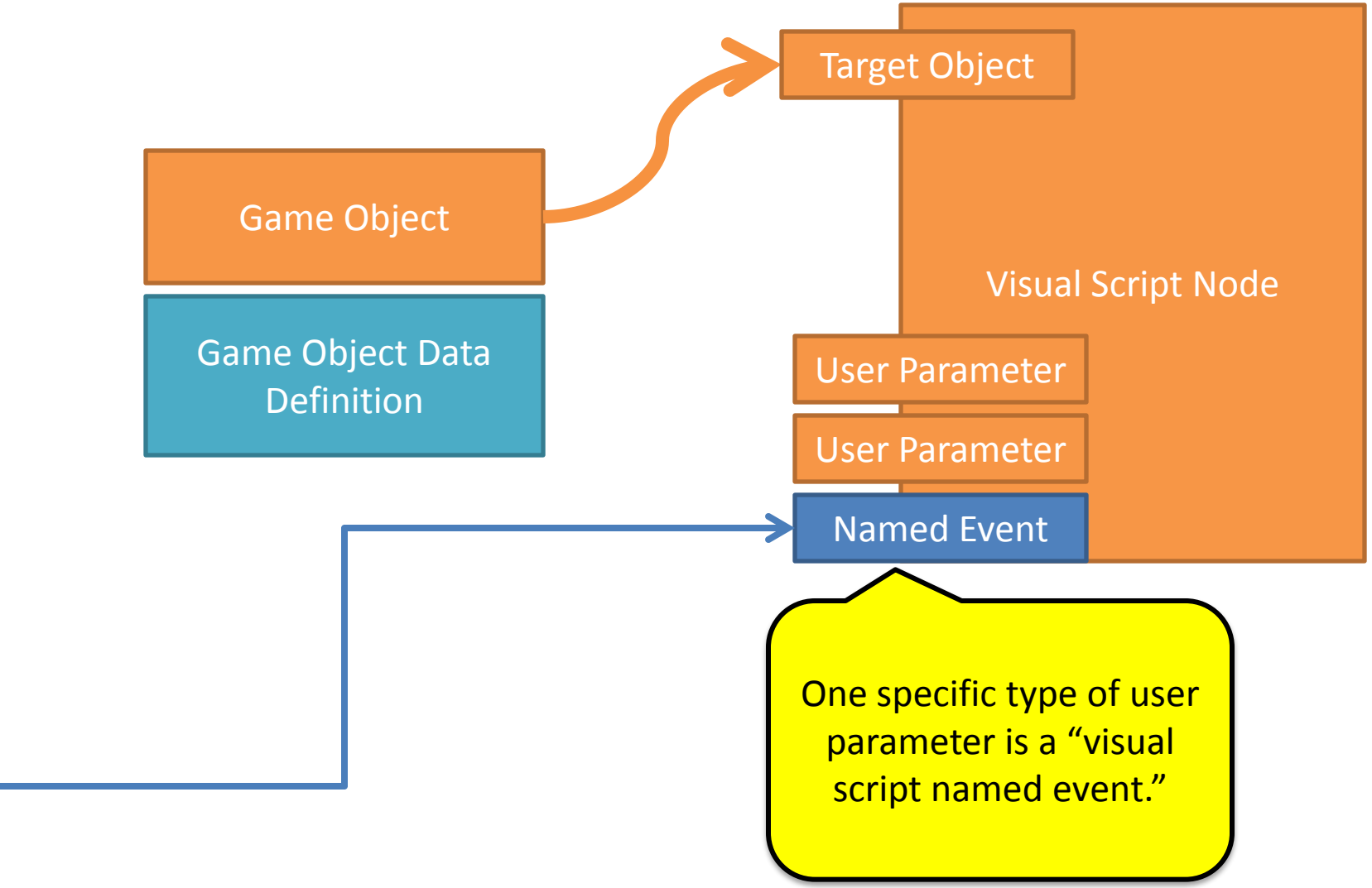

Target Object
Visual Script Node
Visual script state logic
Component
User Parameter
User Parameter
User Parameter
Reference Object Filter
State logic internal to the visual script node can use the user parameters to selectively determine *when* to add the target + reference objects to the component.

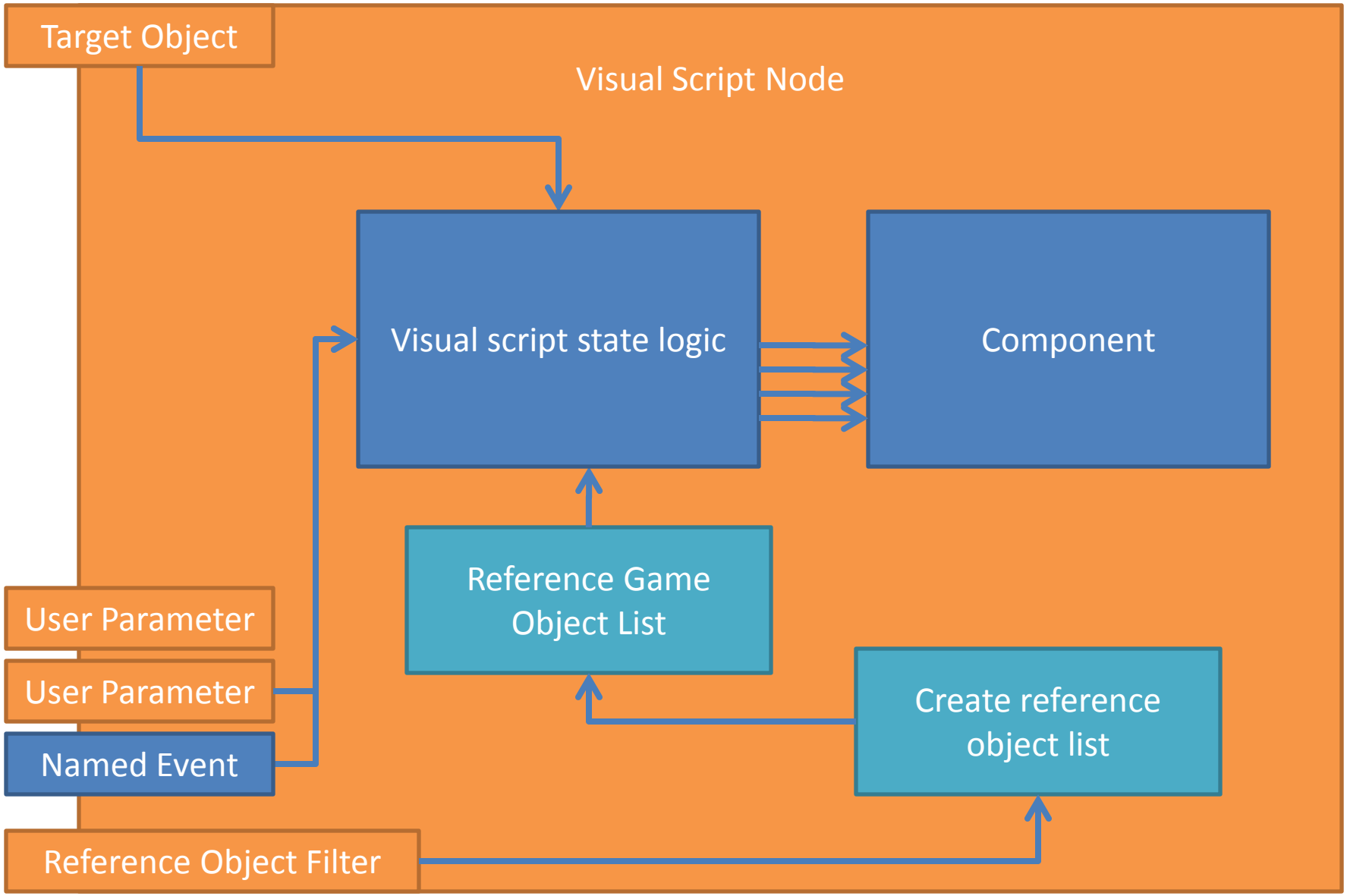

Target Object
Visual Script Node
Visual script state logic
Component
Re
User Parameter
User Parameter
User Parameter
Reference Object Filter
State logi
visua
use th
to select
when to
referenc
co
Or when to remove the target object from the component.
i.e. Can be used entry and exit conditions.

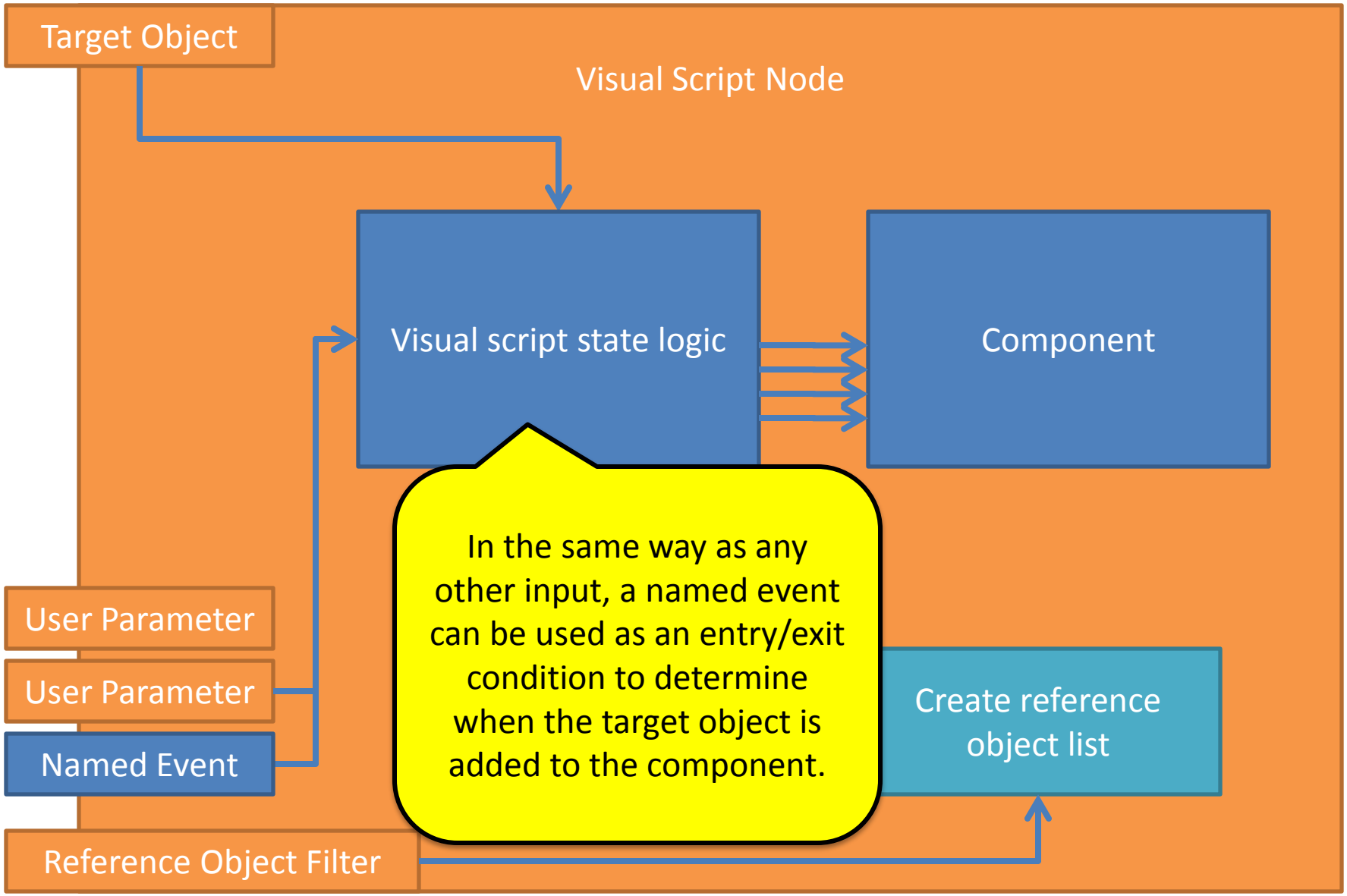

Target Object
Visual Script Node
Game Object Data Definition
Additional required fields
Visual script state logic
User Parameter
User Parameter
User Parameter
Component
Reference Object Filter

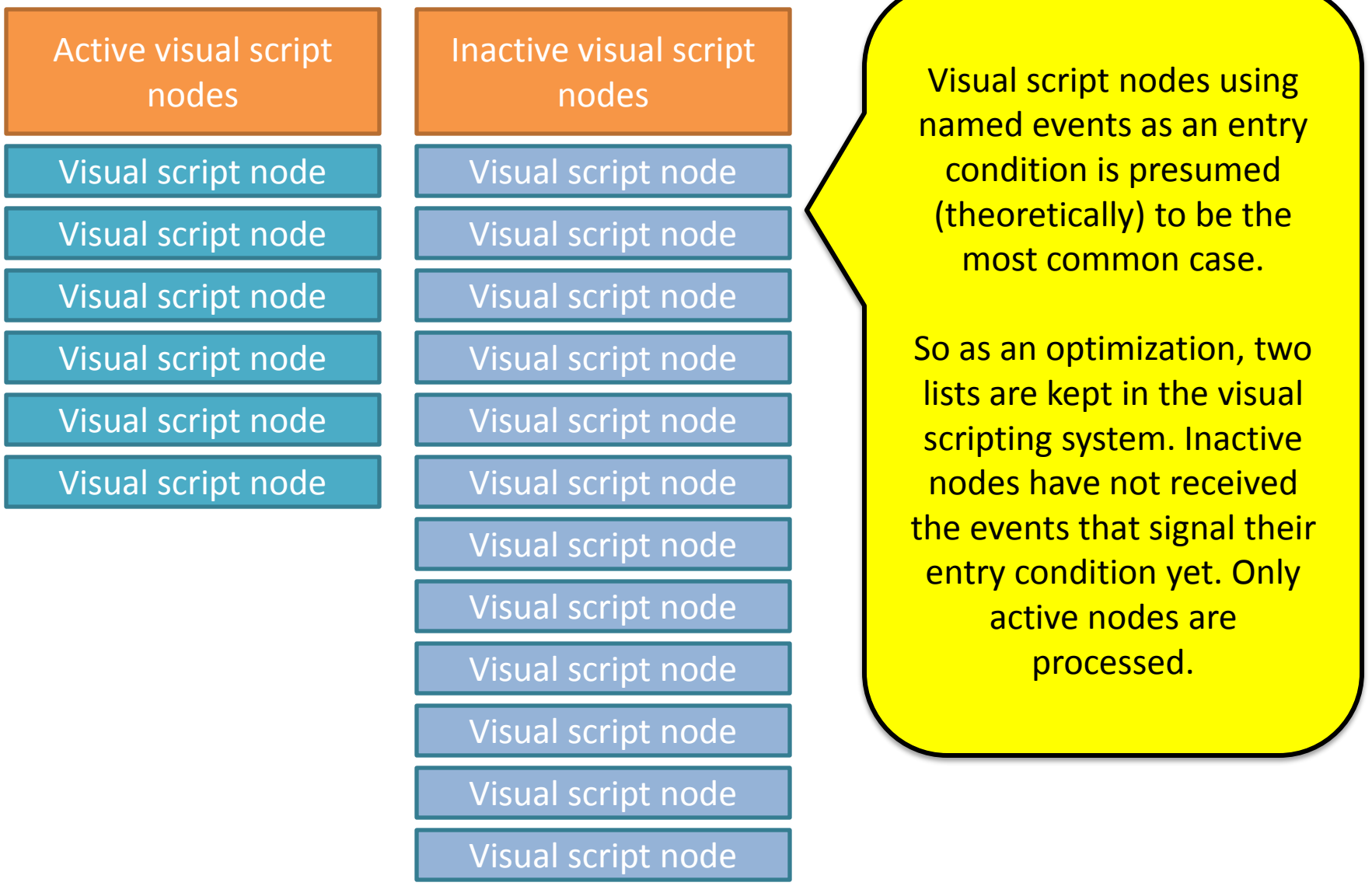

Target Object
Visual Script Node
Game Object Data Definition
Additional required fields
Visual script state logic
The visual script node can also retrieve game object data from the target object that has been written by a component (not just the referenced component) and use that input to modify it's state logic along with the user parameters.
User Parameter
User Parameter
User Parameter
Component
Reference Object Filter

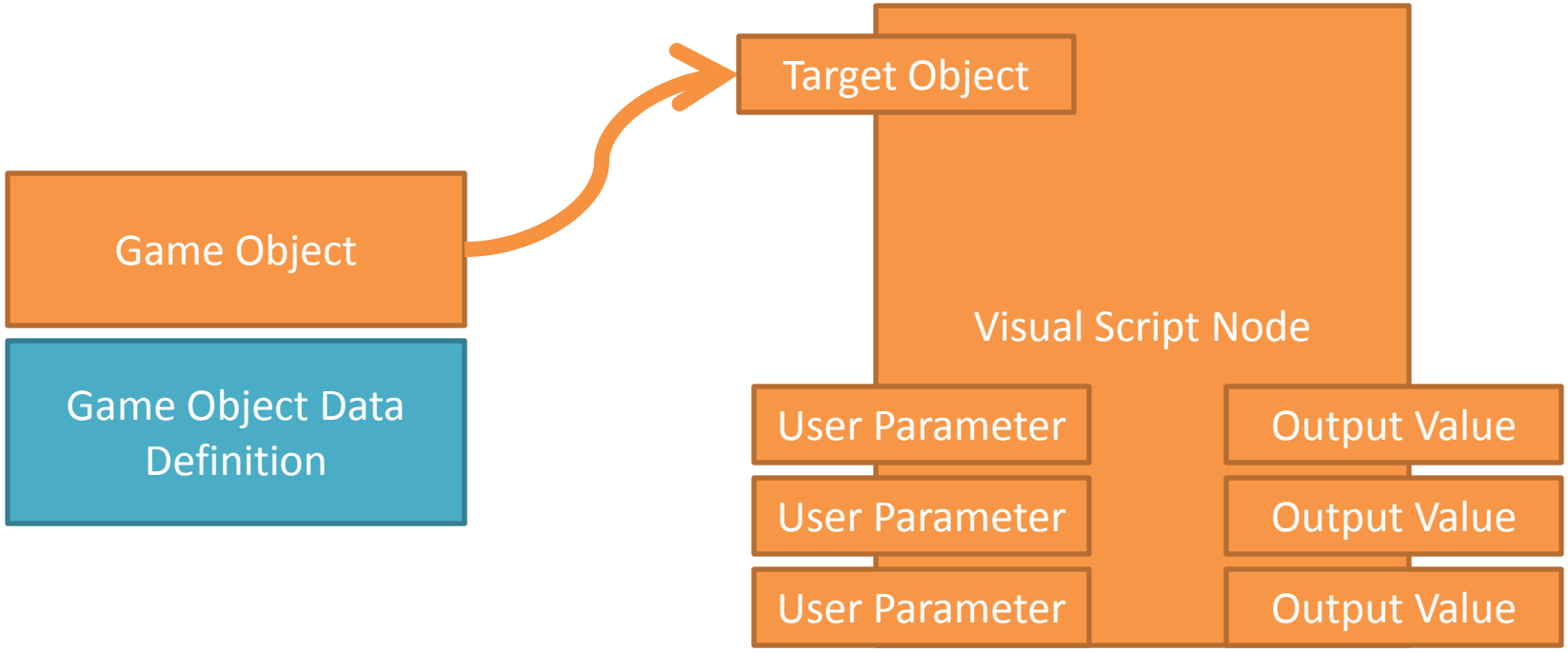

Target Object
Game Object
Visual Script Node
Game Object Data Definition
User Parameter
User Parameter
Named Event

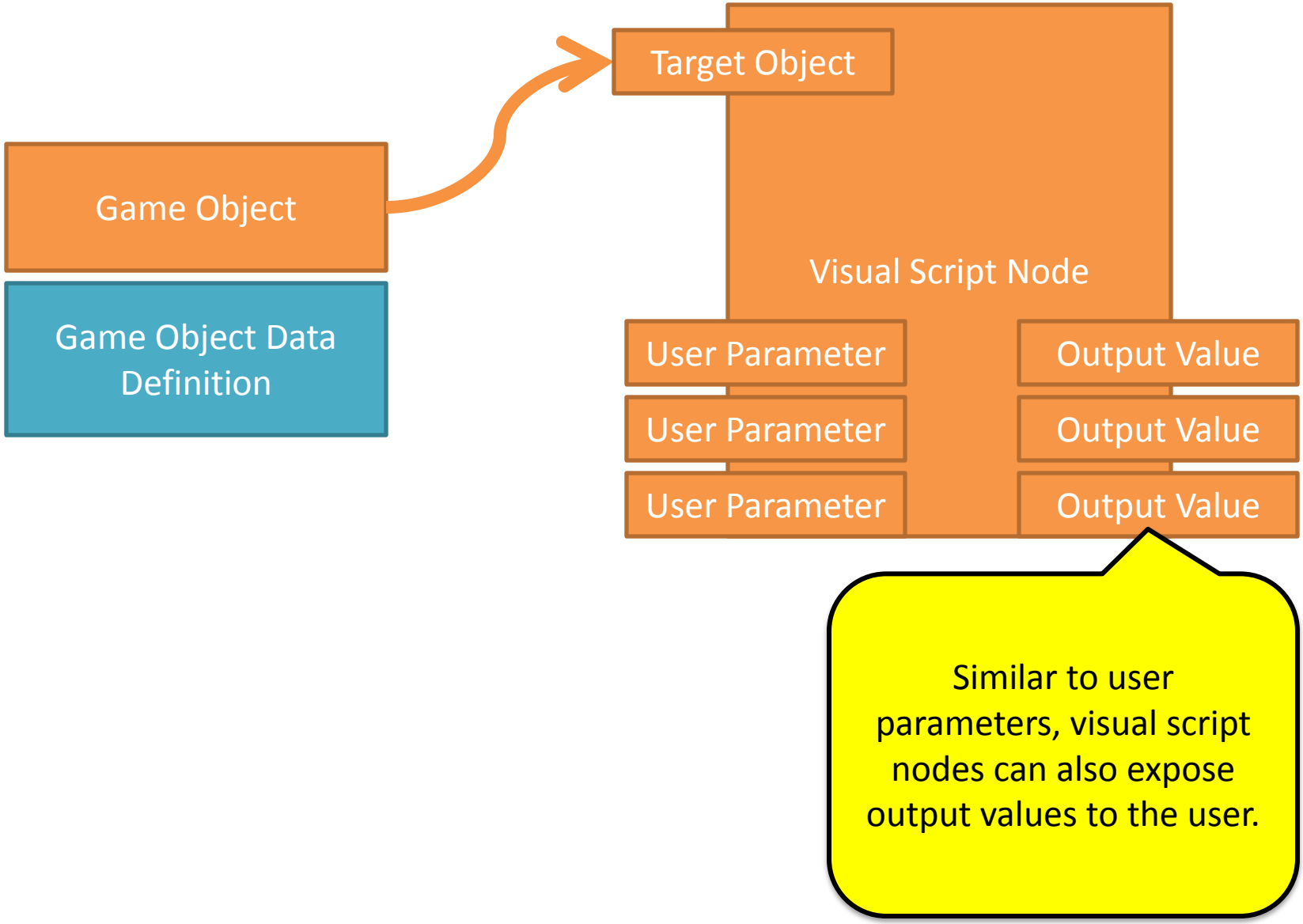

Target Object
Game Object
Visual Script Node
Game Object Data Definition
User Parameter
User Parameter
Named Event
One specific type of user parameter is a “visual script named event.”

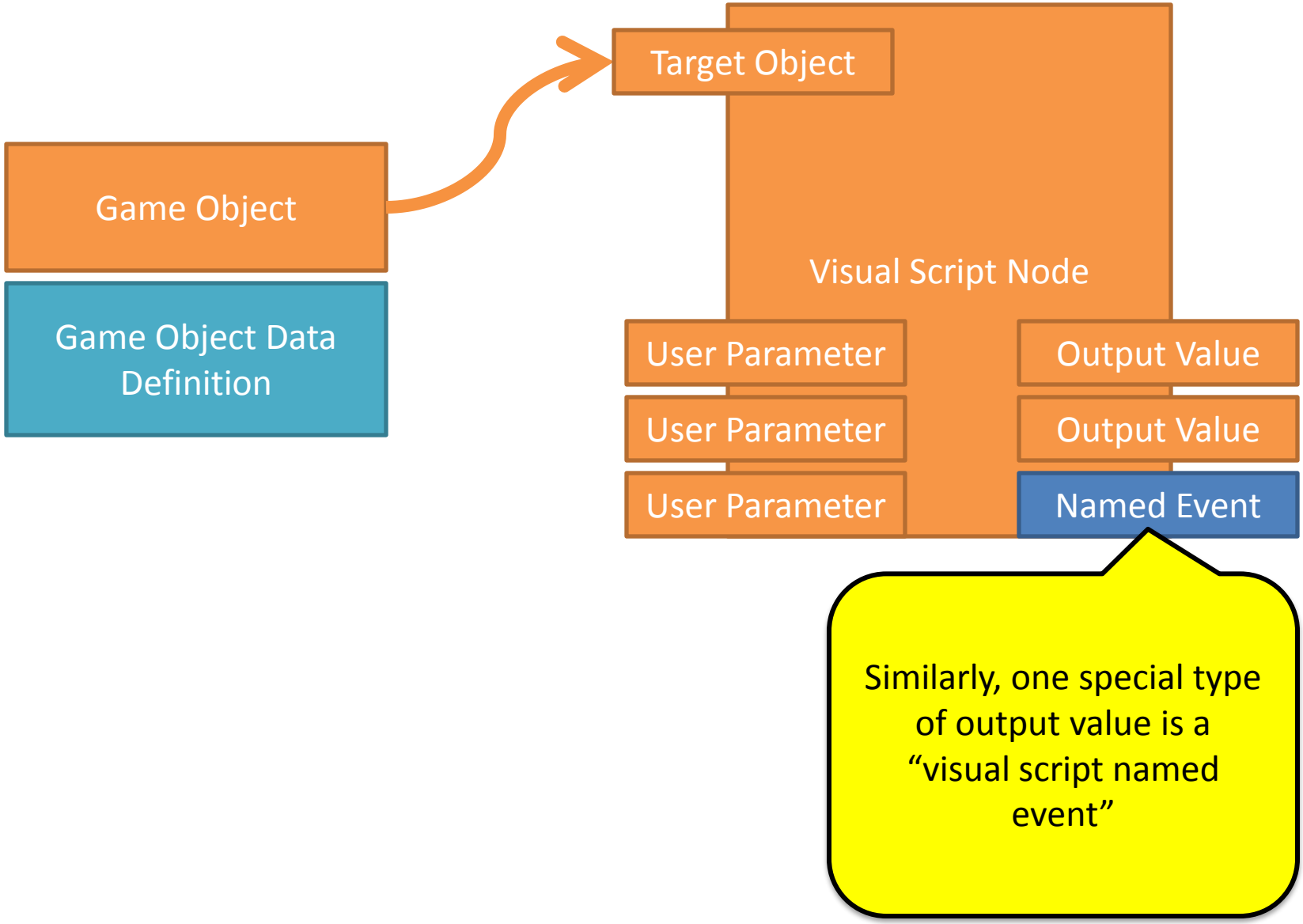

Target Object
Visual Script Node
Visual script state logic
Component
Reference Game Object List
User Parameter
User Parameter
Named Event
Create reference object list
Reference Object Filter

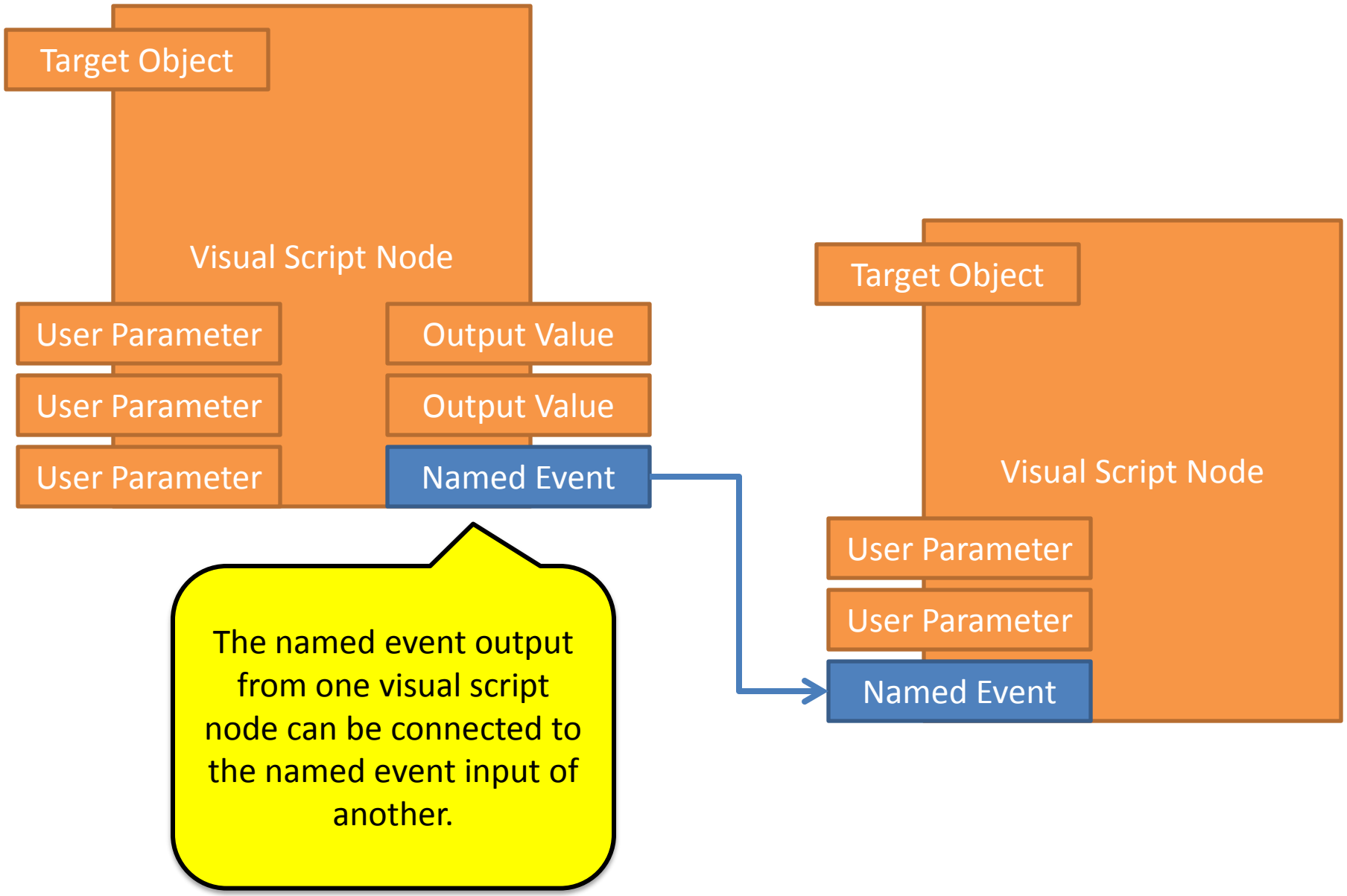

Target Object
Visual Script Node
Visual script state logic
Component
In the same way as any other input, a named event can be used as an entry/exit condition to determine when the target object is added to the component.
User Parameter
User Parameter
Named Event
Create reference object list
Reference Object Filter

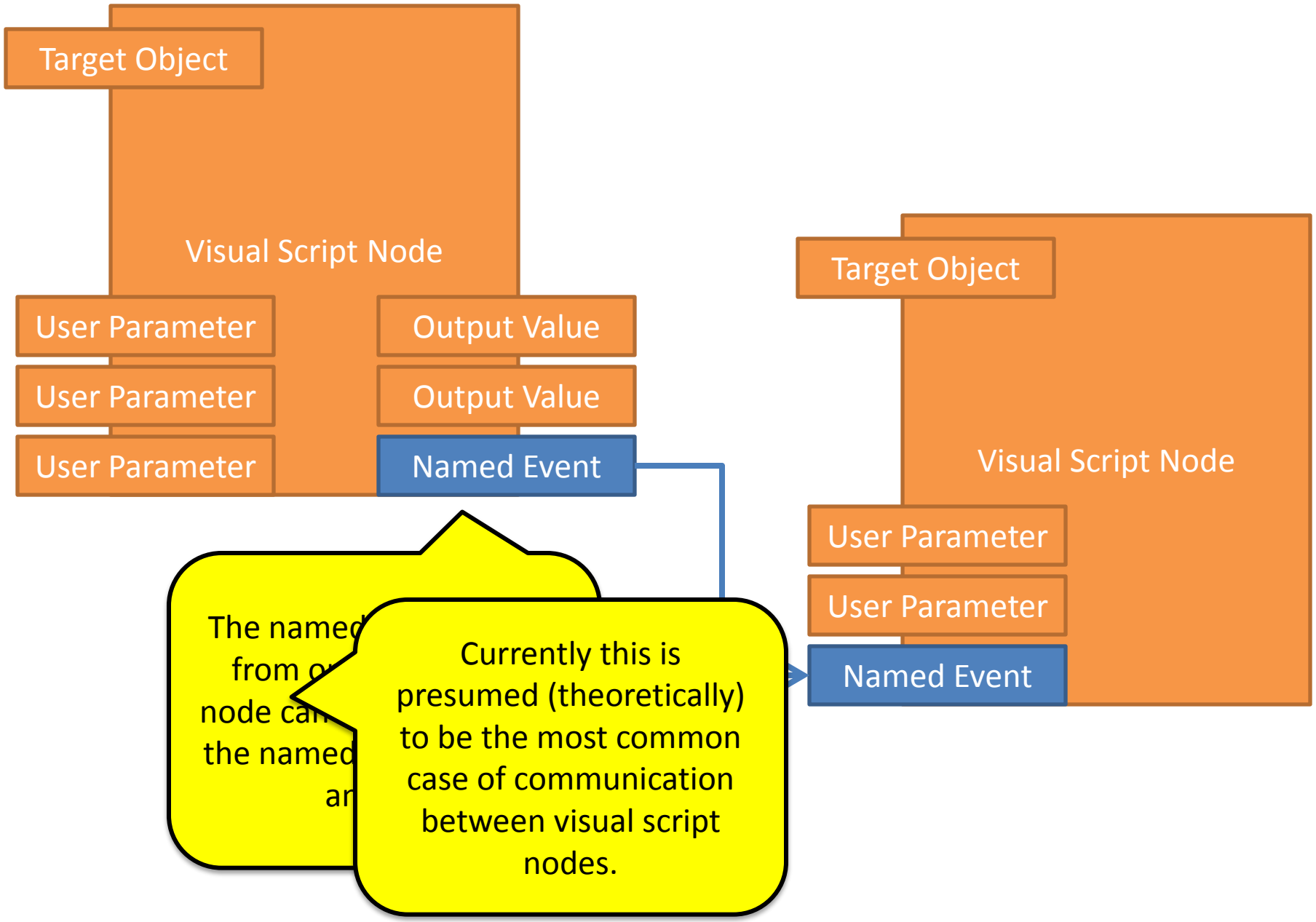

Active visual script nodes
Visual script node
Visual script node
Visual script node
Visual script node
Visual script node
Visual script node
Inactive visual script nodes
Visual script node
Visual script node
Visual script node
Visual script node
Visual script node
Visual script node
Visual script node
Visual script node
Visual script node
Visual script node
Visual script node
Visual script node
Visual script node
Visual script nodes using named events as an entry condition is presumed (theoretically) to be the most common case.
So as an optimization, two lists are kept in the visual scripting system. Inactive nodes have not received the events that signal their entry condition yet. Only active nodes are processed.

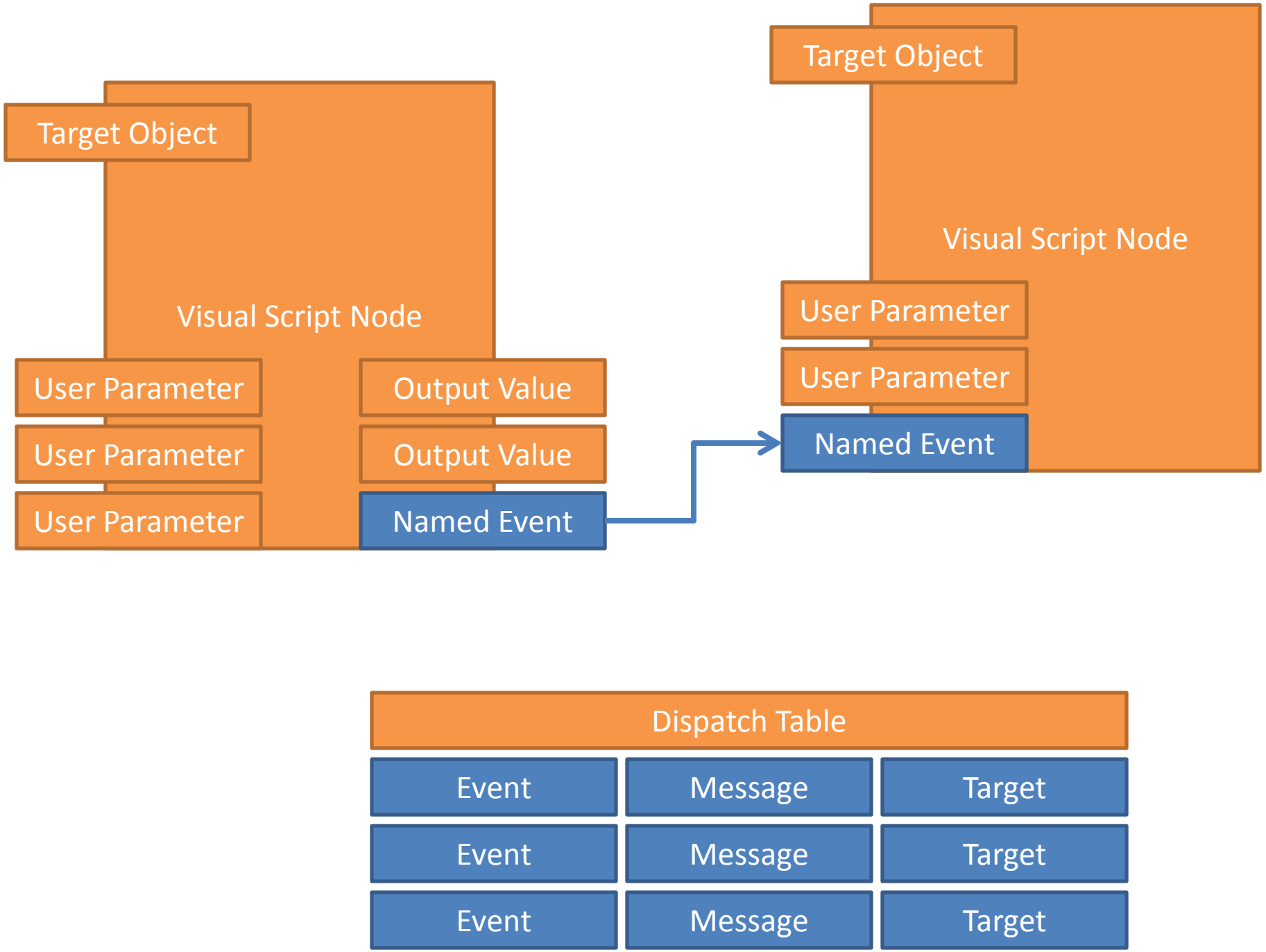

Target Object
Game Object
Visual Script Node
Game Object Data Definition
User Parameter
User Parameter
User Parameter
Output Value
Output Value
Output Value

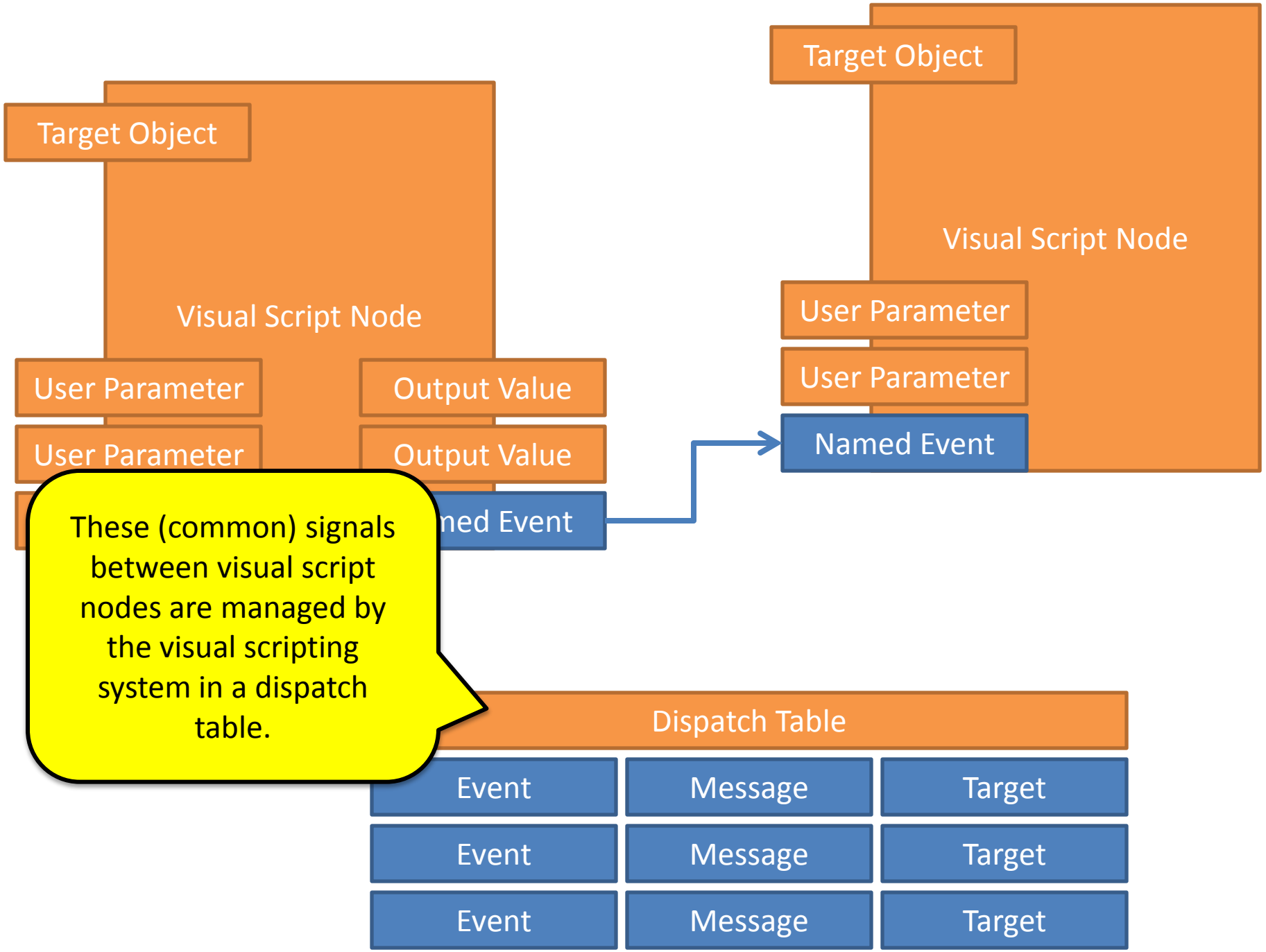

Target Object
Game Object
Game Object Data Definition
Visual Script Node
User Parameter
User Parameter
User Parameter
Output Value
Output Value
Output Value
Similar to user parameters, visual script nodes can also expose output values to the user.

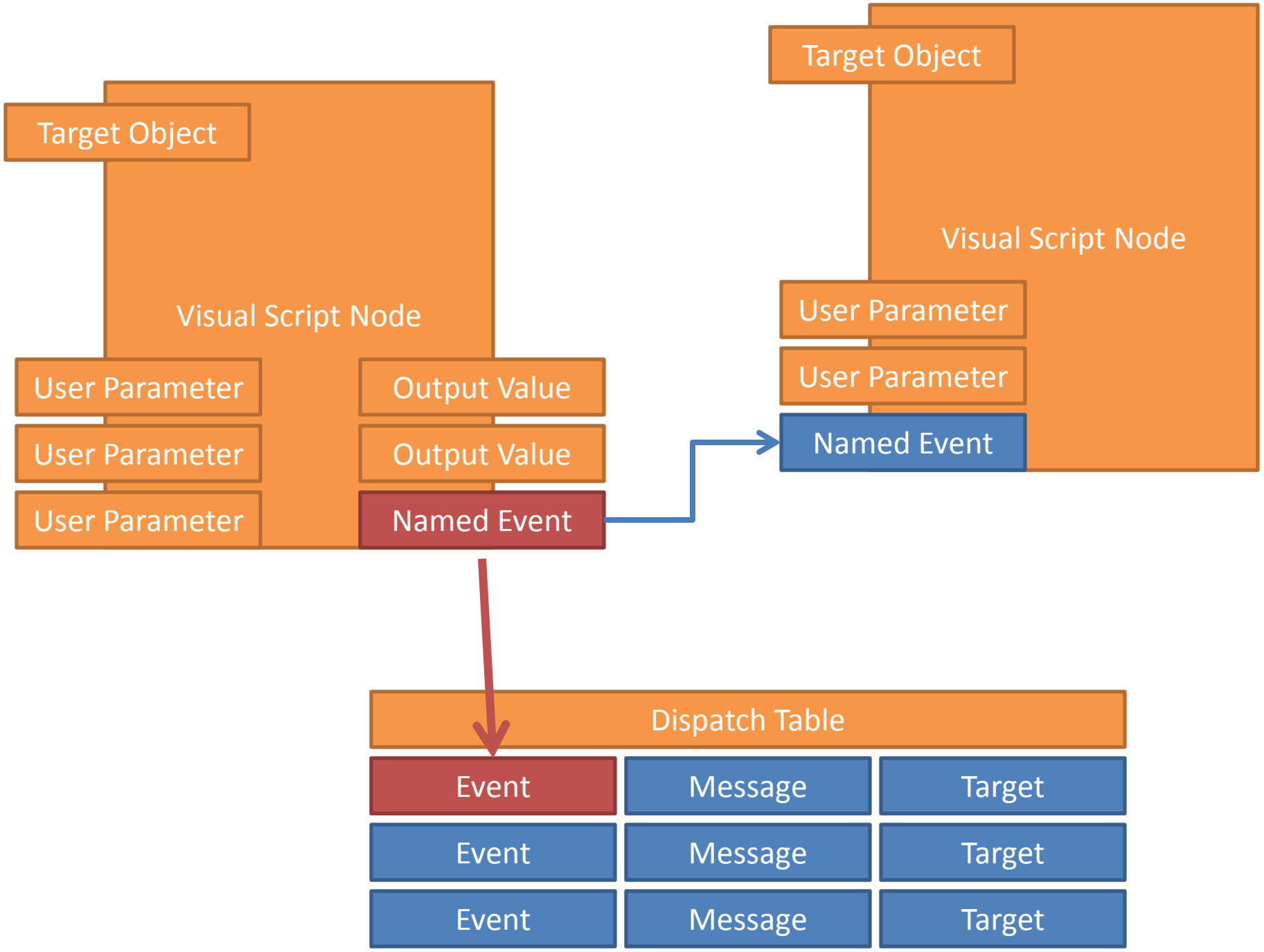

Target Object
Game Object
Game Object Data Definition
Visual Script Node
User Parameter
User Parameter
User Parameter
Output Value
Output Value
Named Event
Similarly, one special type of output value is a “visual script named event”

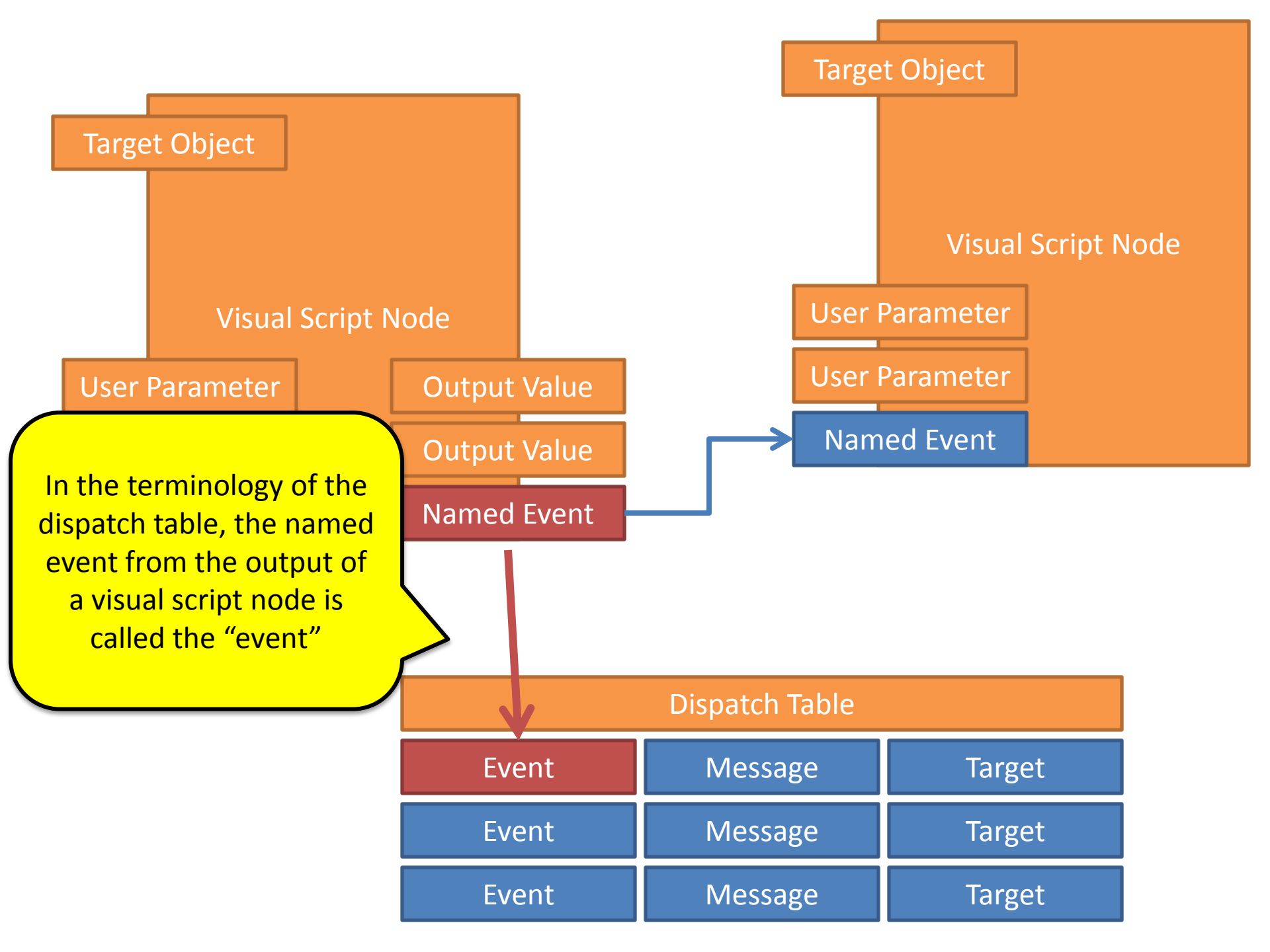

Target Object
Visual Script Node
User Parameter
User Parameter
User Parameter
Output Value
Output Value
Named Event
The named event output from one visual script node can be connected to the named event input of another.
Target Object
Visual Script Node
User Parameter
User Parameter
Named Event

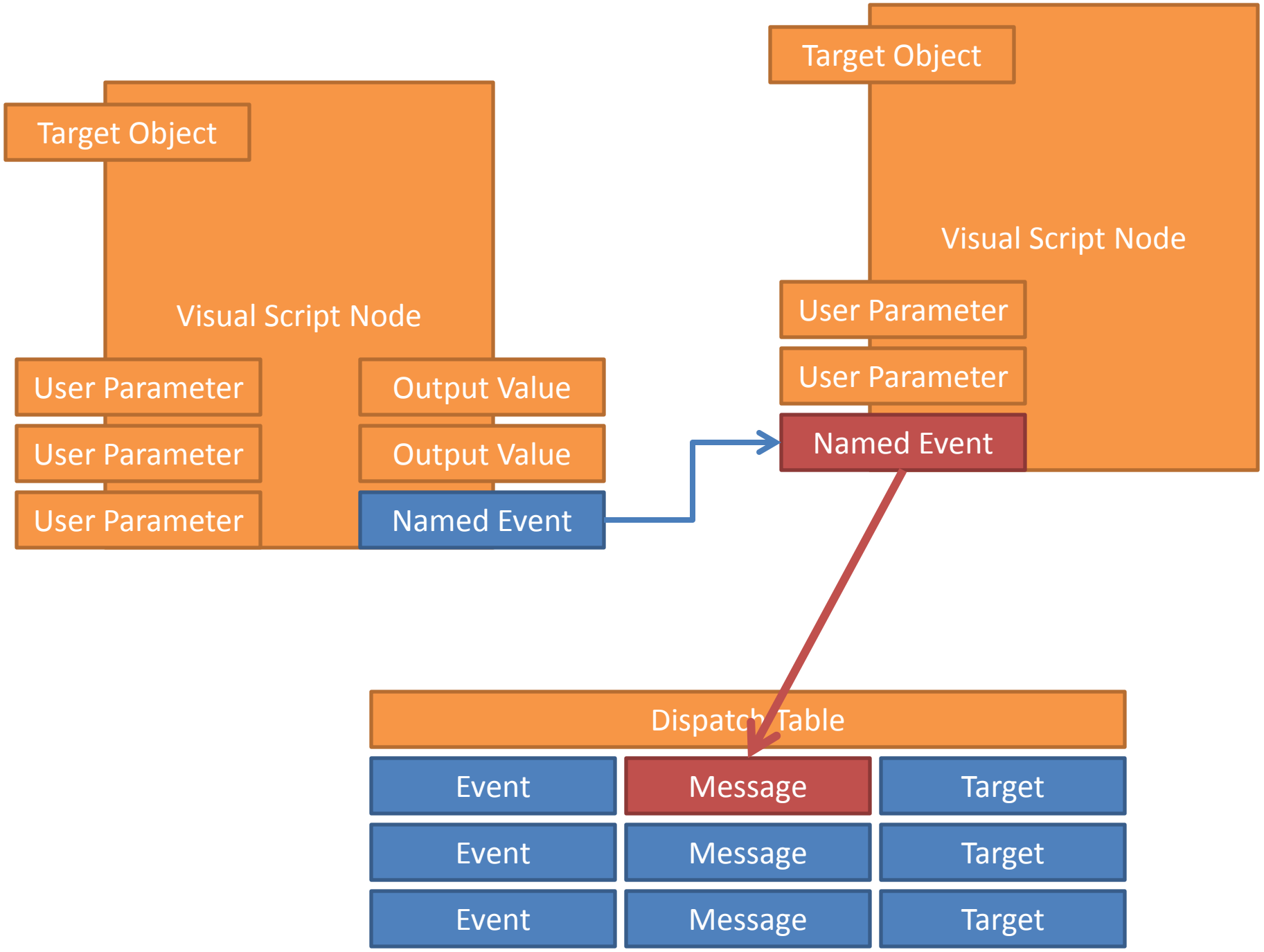

Target Object
Visual Script Node
User Parameter
User Parameter
User Parameter
Output Value
Output Value
Named Event
Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
The named
from ou
node can
the named
ar
Currently this is presumed (theoretically) to be the most common case of communication between visual script nodes.

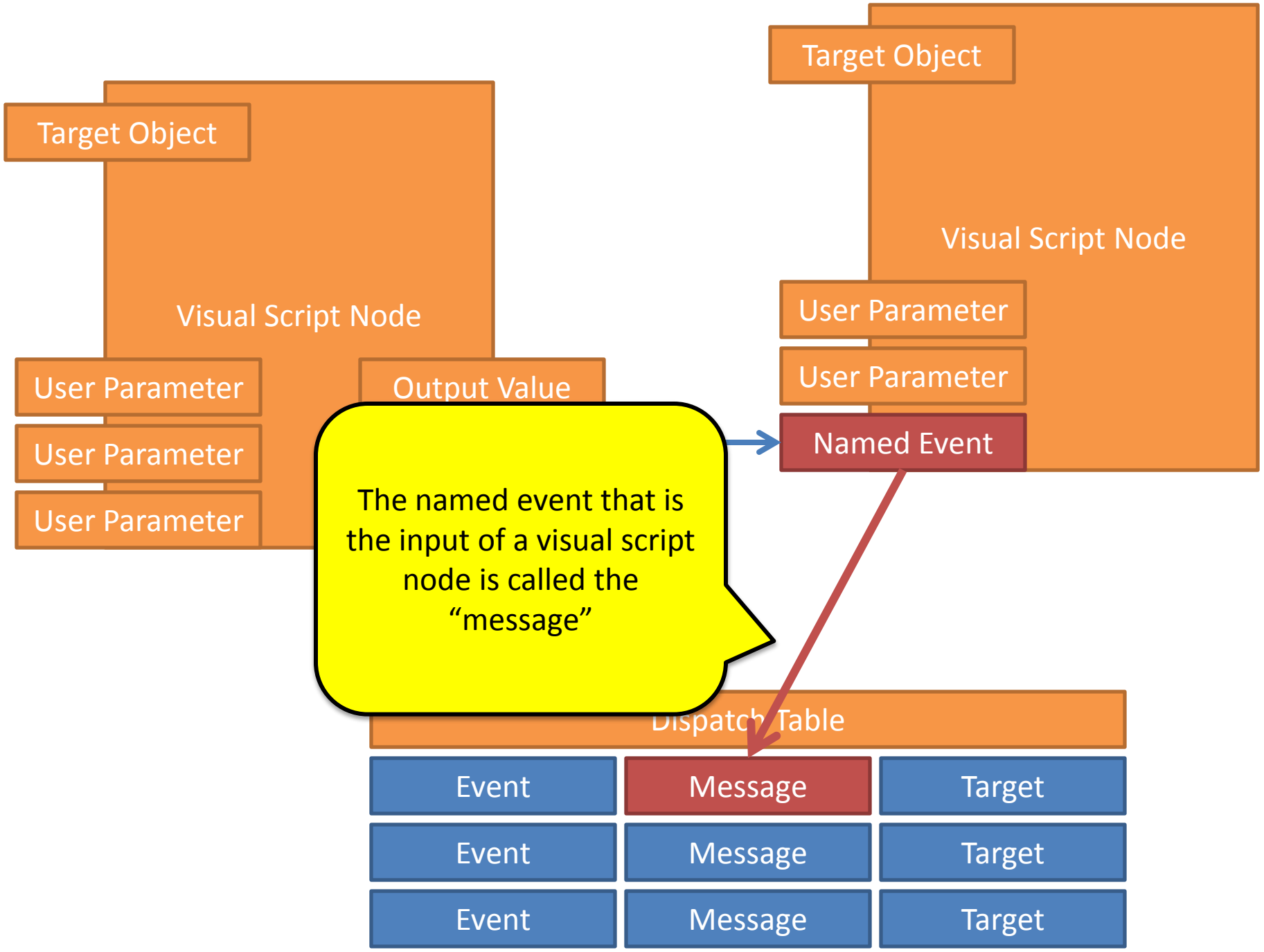

Target Object
Visual Script Node
User Parameter
User Parameter
User Parameter
Output Value
Output Value
Named Event
Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target

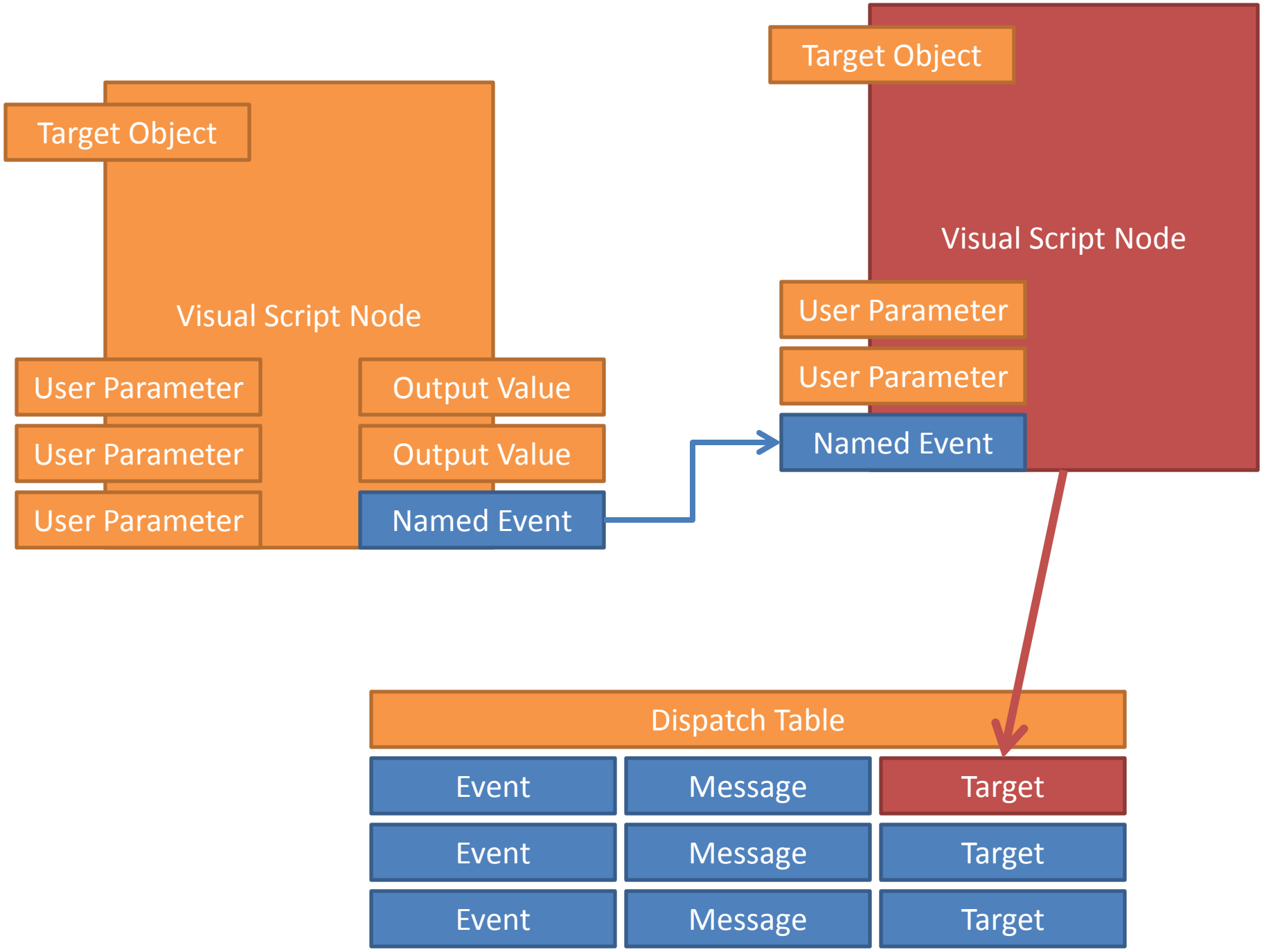

Target Object
Visual Script Node
User Parameter
User Parameter
Output Value
Output Value
ned Event
Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
These (common) signals between visual script nodes are managed by the visual scripting system in a dispatch table.
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target

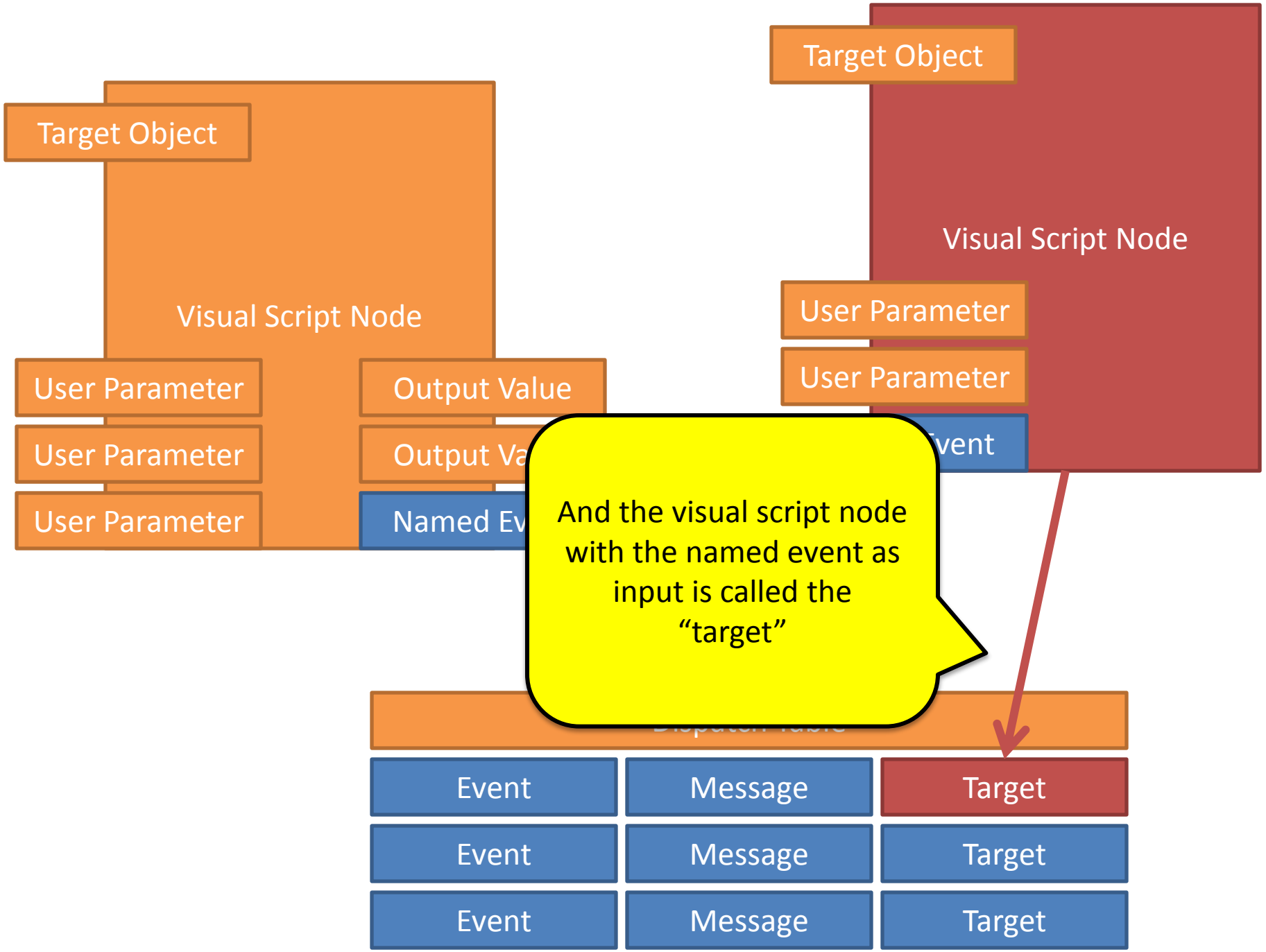

Target Object
Visual Script Node
User Parameter
User Parameter
User Parameter
Output Value
Output Value
Named Event
Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target

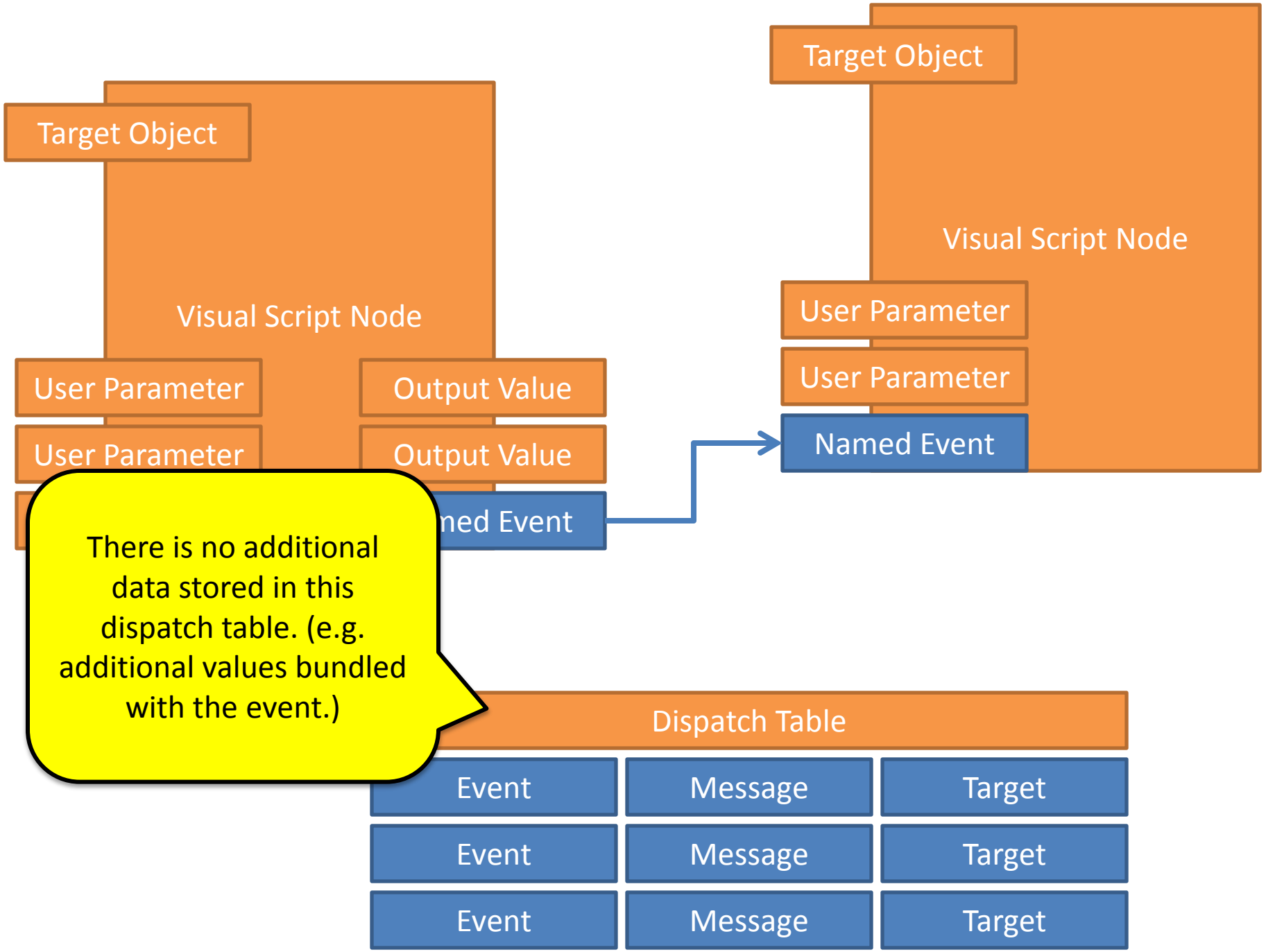

Target Object
Visual Script Node
User Parameter
Output Value
Output Value
Named Event
Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
In the terminology of the dispatch table, the named event from the output of a visual script node is called the “event”
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target

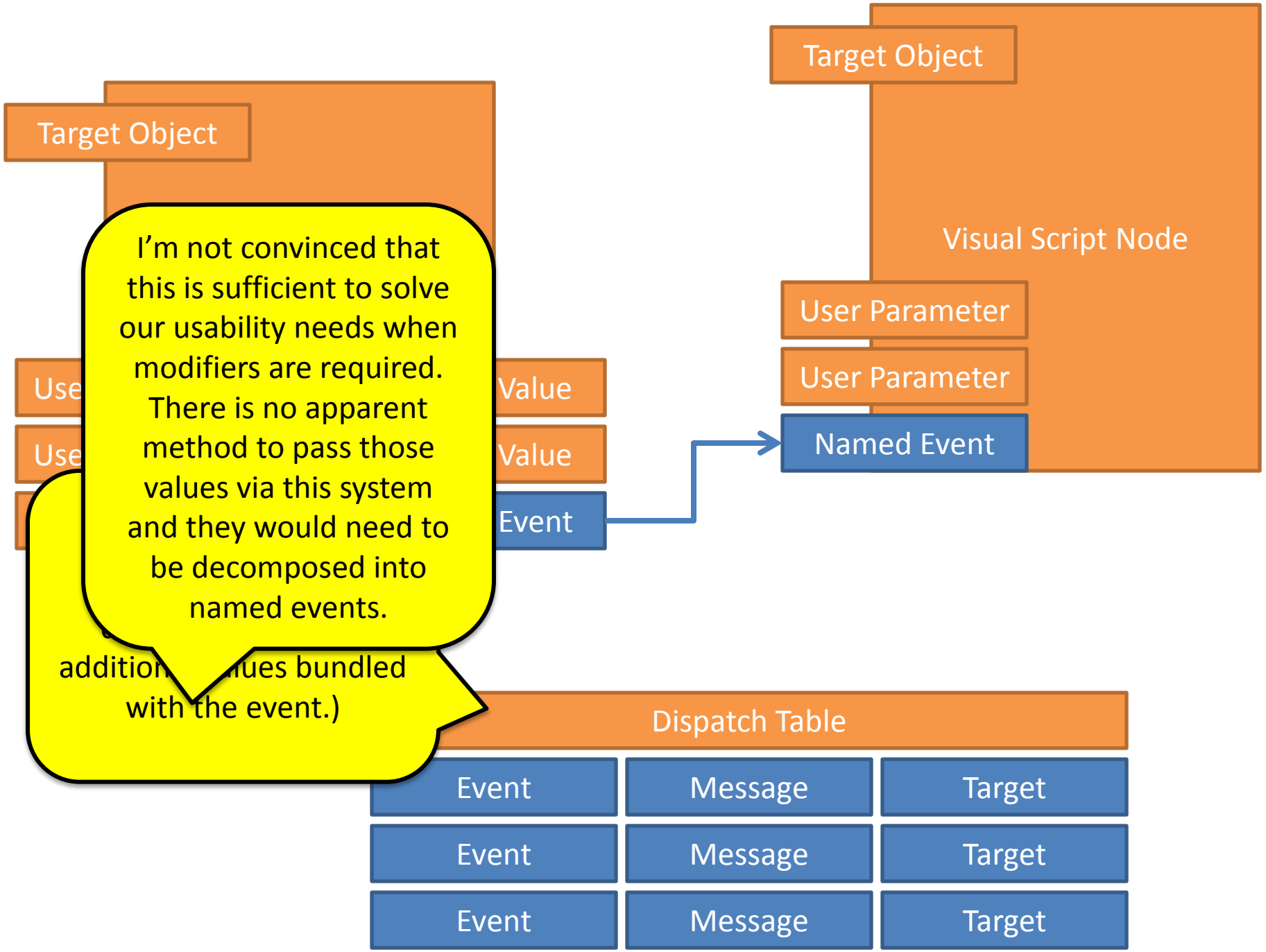

Target Object
Visual Script Node
User Parameter
User Parameter
User Parameter
Output Value
Output Value
Named Event
Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target

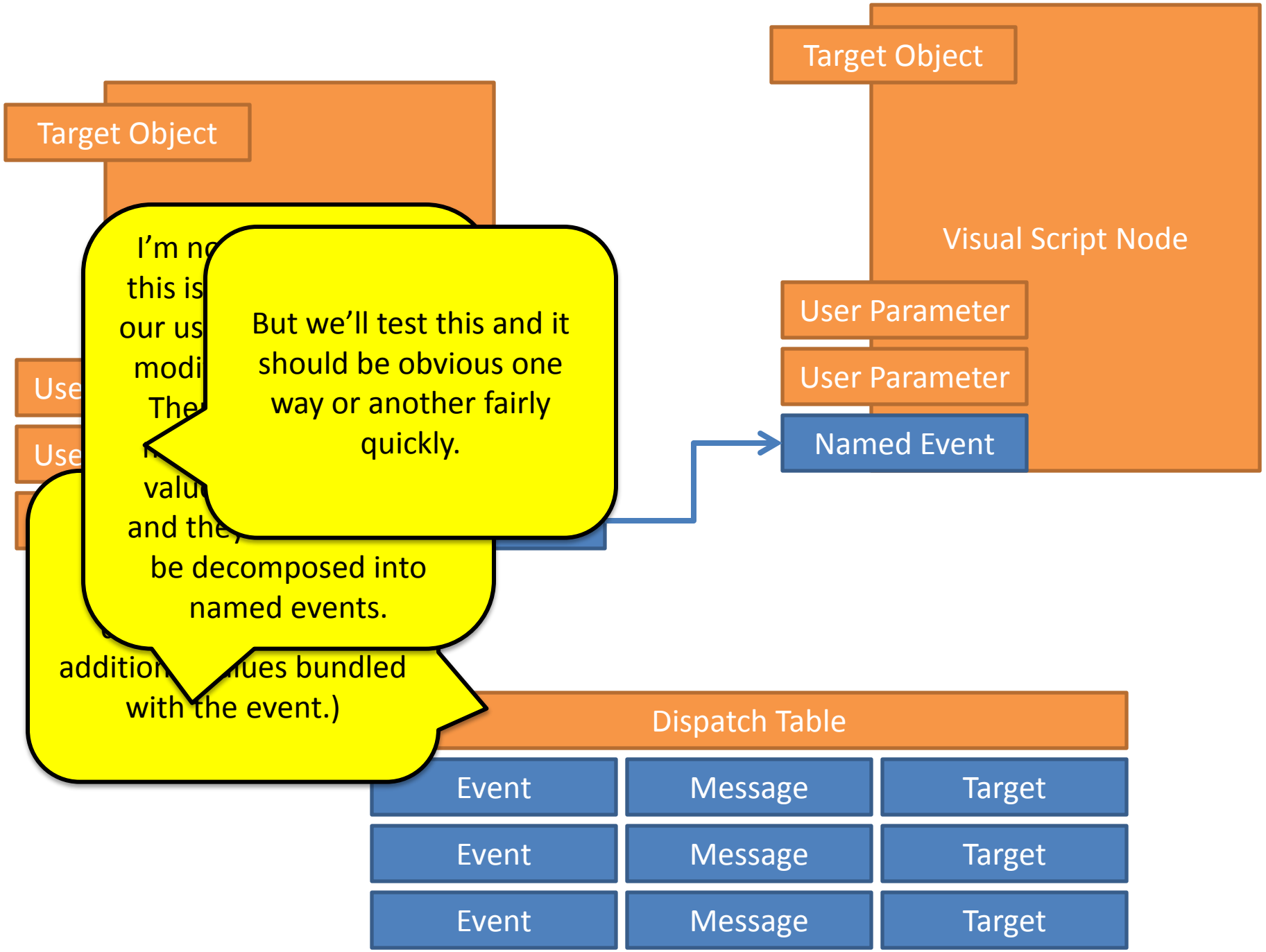

Target Object
Visual Script Node
User Parameter
User Parameter
User Parameter
Output Value
Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
The named event that is the input of a visual script node is called the "message"
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target

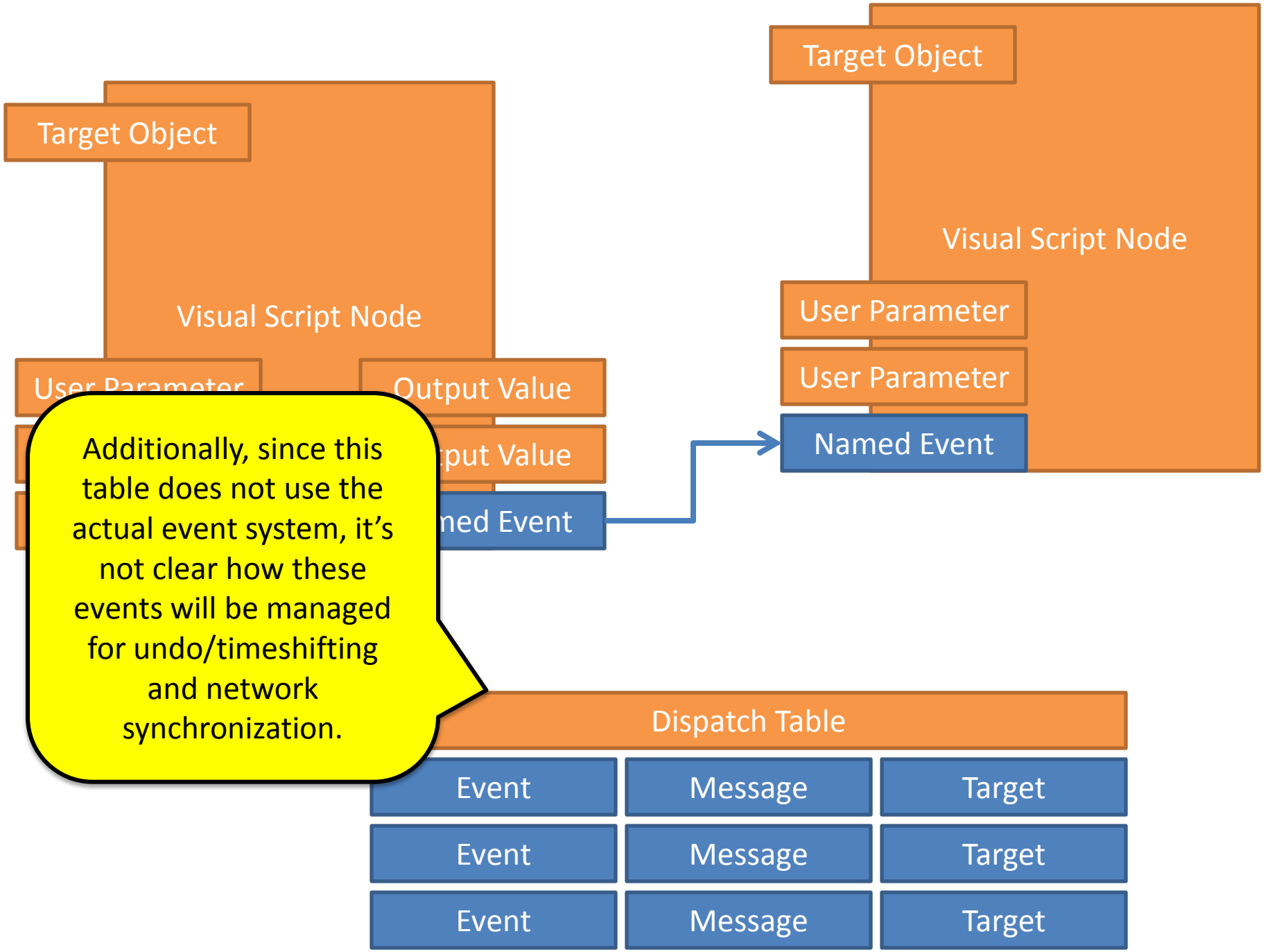

Target Object
Visual Script Node
User Parameter
User Parameter
User Parameter
Output Value
Output Value
Named Event
Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target

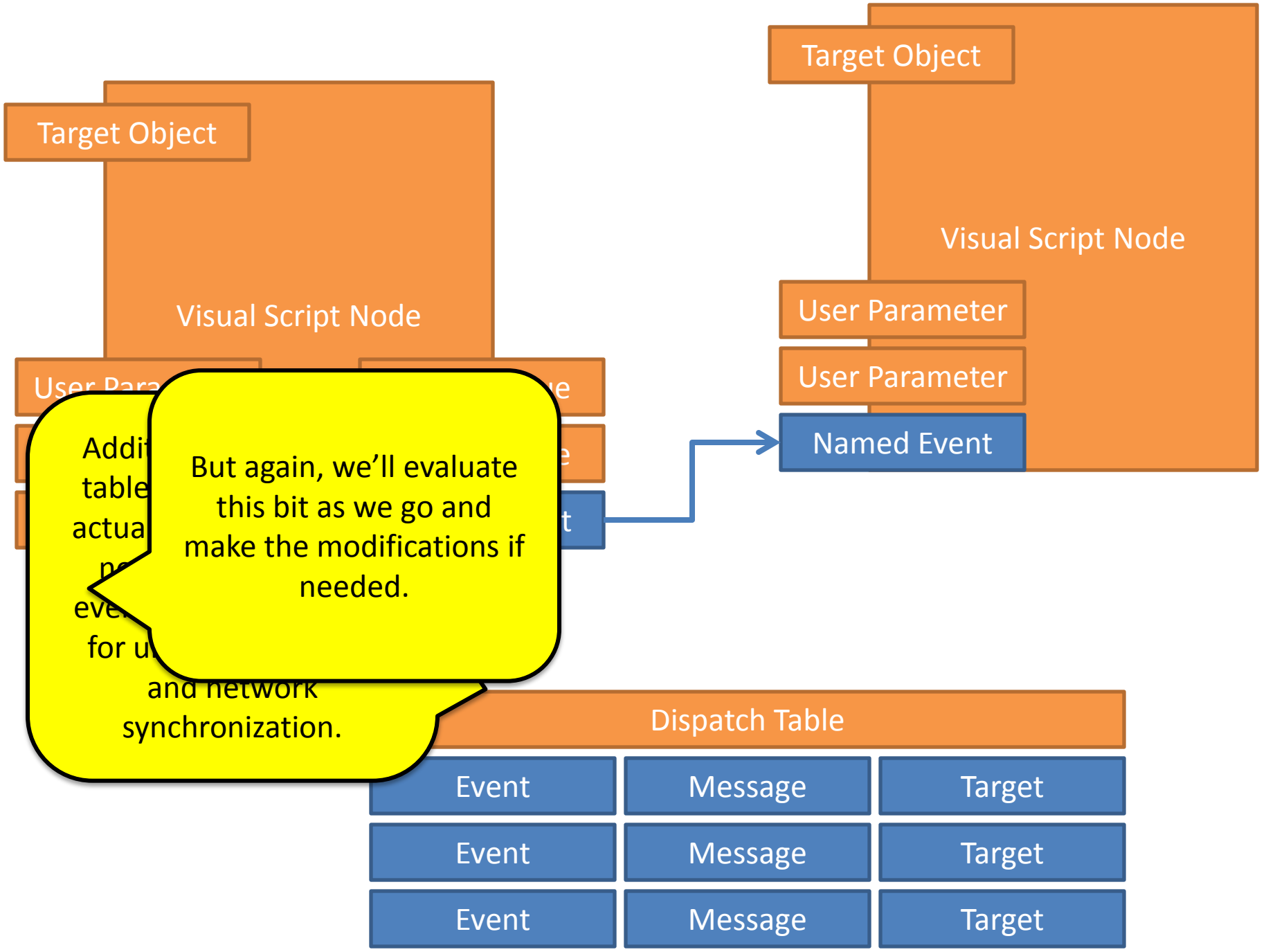

Target Object
Visual Script Node
User Parameter
User Parameter
User Parameter
Output Value
Output Va
Named Ev
Target Object
Visual Script Node
User Parameter
User Parameter
vent
And the visual script node with the named event as input is called the "target"
Event
Message
Target
Event
Message
Target
Event
Message
Target


Target Object
Visual Script Node
User Parameter
User Parameter
Output Value
Output Value
ned Event
Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
There is no additional data stored in this dispatch table. (e.g. additional values bundled with the event.)
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target


Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
Target Object
Use
Use
Value
Value
Event
I'm not convinced that this is sufficient to solve our usability needs when modifiers are required. There is no apparent method to pass those values via this system and they would need to be decomposed into named events.
addition ues bundled with the event.)
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target


Target Object
Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
But we'll test this and it should be obvious one way or another fairly quickly.
be decomposed into named events.
with the event.)
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target


Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
Target Object
Visual Script Node
Output Value
Additionally, since this table does not use the actual event system, it's not clear how these events will be managed for undo/timeshifting and network synchronization.
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target


Target Object
Visual Script Node
User Parameter
User Parameter
Named Event
Target Object
Visual Script Node
But again, we'll evaluate this bit as we go and make the modifications if needed.
and network synchronization.
Dispatch Table
Event
Message
Target
Event
Message
Target
Event
Message
Target