# PicoContainer Web

Published version: 2.0

Built on Pico Container

### Pico Web Applications



### Web Frameworks



### Axis



### Hosted by

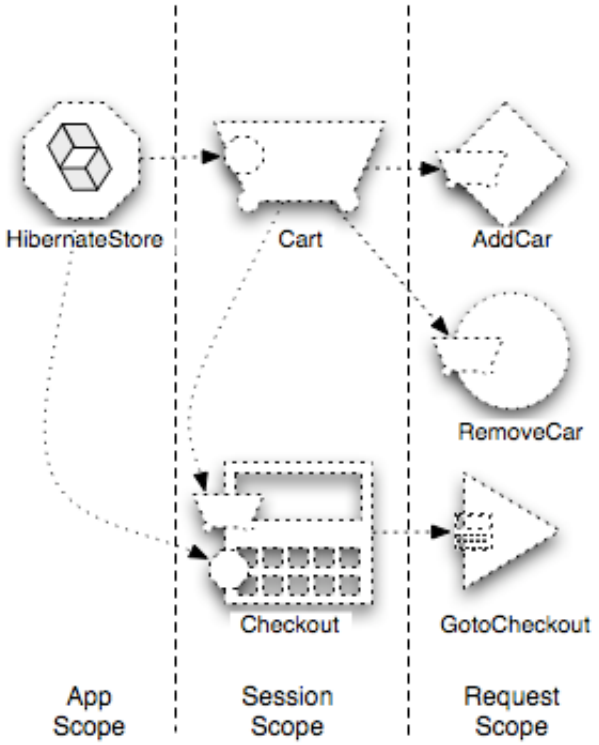
### Tools

## Scoping Web Components

Choosing which scope for you components requires you to understand understand how scopes work. Consider a contrived stateful web application that is all about buying cars. A subset of the functions it must perform are 'add car to cart', 'remove car from cart' and 'go to checkout'

The Components we are going to need for this minimalistic web application are 'ShoppingCart' and 'Store'. The actual functions above are also going to be transient components 'AddToCart', 'RemoveFromCart' and 'GotoCheckout'.

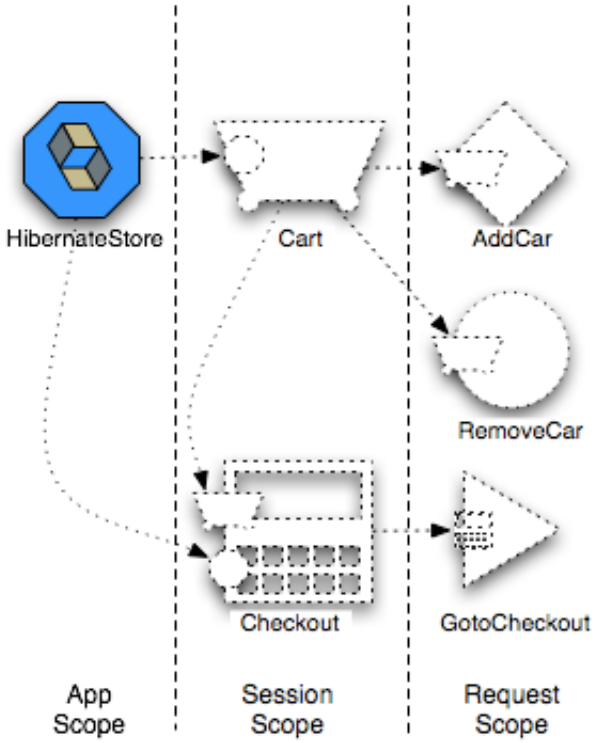
Assuming a composition script for these components, the initial state of the stack is one where all the components are in place, but nothing has been instantiated yet.




The components we've described above are

shapes, and the dependencies describes with arrowed lines. For these diagrams it does not matter which type of Dependency Injection is being used.

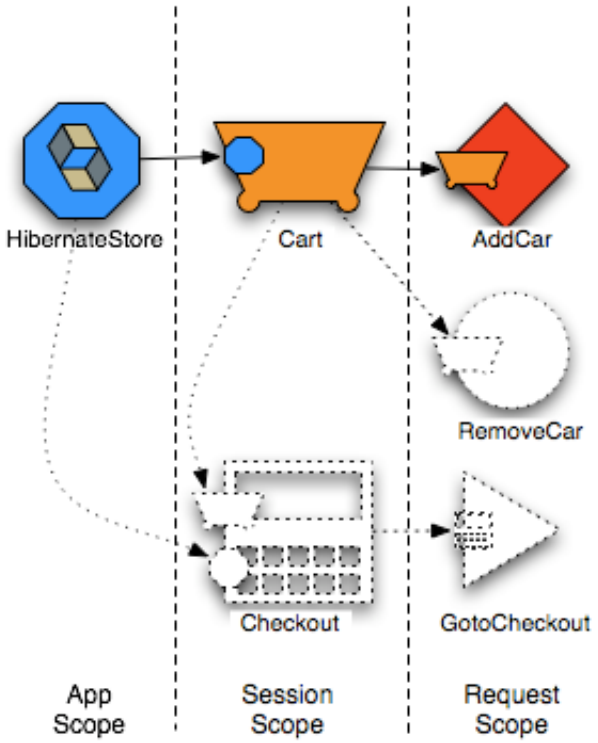
Next, consider the web container starting up, or where there are no active sessions:



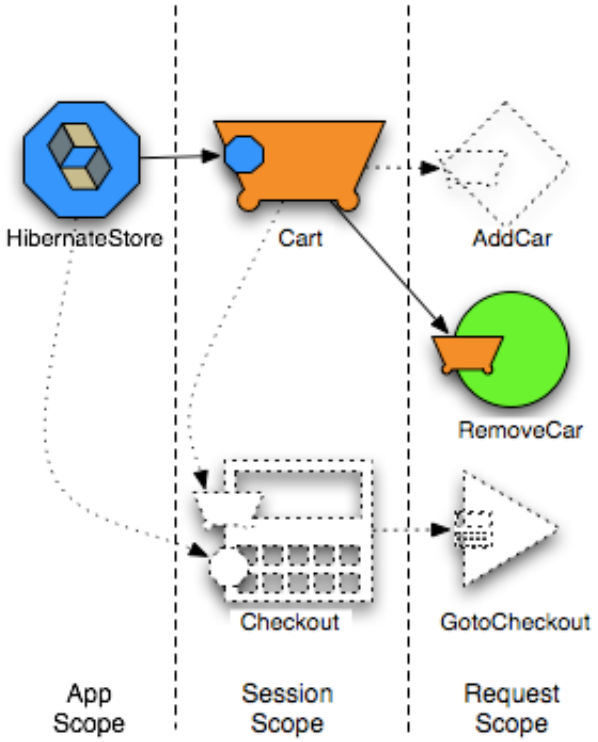

Colored shapes are instantiated, whereas dotted shapes are not (for that session). Solid arrows are actual injections, whereas dotted arrows are ones that could injections were the corresponding components to be instantiated.

Then after the first web request from a new session, a session scoped component is instantiated, and an action that will modify it based on input:

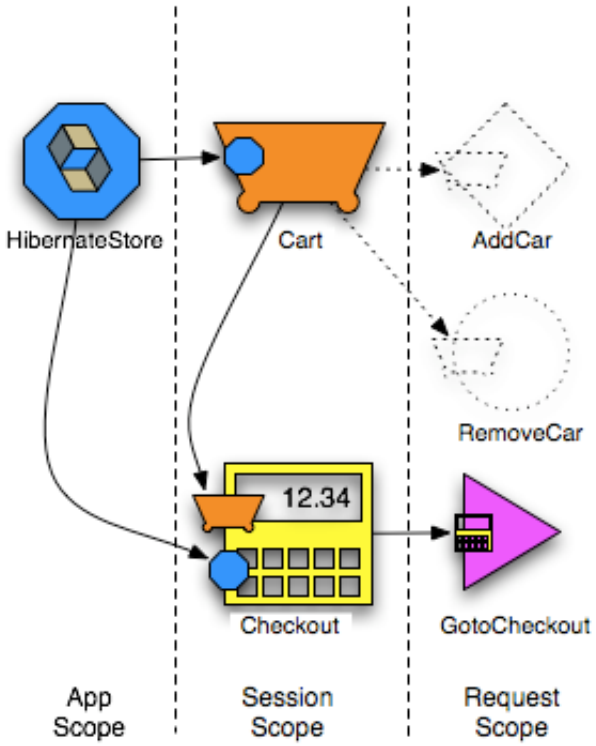
Cart is stateful at the session level, because it contains a list of items to potentially purchase.




Alternatively, a car could be removed from the cart. In this case the AddToCart component that had previously been instantiated, is has already been garbage collected for this session...




Finally, the user could go to the checkout with the contents of the cart ...




Checkout is stateful at the session level because it not only refers to the contents of a cart, but also contains new details like a payment mechanism for the pending purchase.