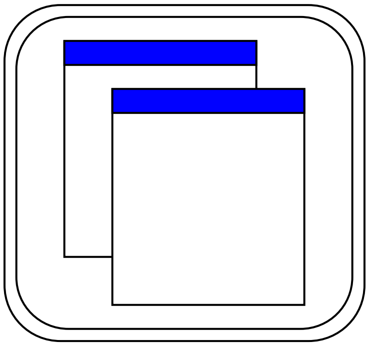
Sessions in Browsers



When multiple browser windows are open, browsers send the same cookie for all windows. The result is that a single session is used for all interactions with a single user.

The advantage of this approach is that these two windows "share" information – for example, if the user has signed on in one, the second window is also authenticated.

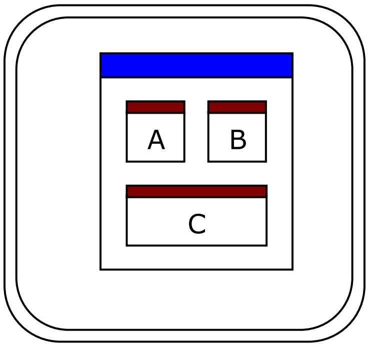
The disadvantage of this approach is that if the Web application uses sessions, then the user can't carry out to disjoint actions. For example, if both windows have a hotel reservation application, then you can't book two different hotels in parallel (if the hotel information is stored in a session).

Consequently, good Web sites take great care to try to be stateless as much as possible through the process (also for scalability), but even when they become stateless, they store only essential pieces of information in the session (mainly for functionality reasons).

Sessions in WSIA/WSRP

Basic Scenario

The challenge in WSIA/WSRP, is that the two cases are needed: sometimes applications (portlets) need to share data, and sometimes not.



In the illustration on the right hand side, A and B may need to share information, but may also represent two identical portlets that need to be isolated to enable to user to view/use separate data.

Conceptually, if A and B need to **share information**, then they can share the same session (more on that below).

When A and B need to be **isolated**, there are two implementation options for the Producer:

1. Use a single session for both portlets, A and B are responsible for segregating their information in the Session object (e.g., by using a key, which we call the entity).
2. Use a separate session for A and for B.

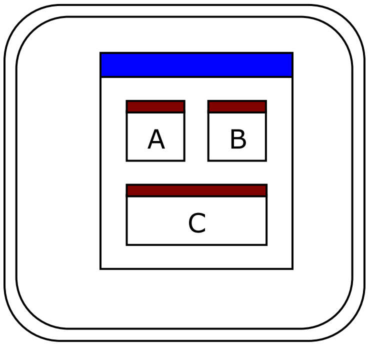
The advantage of approach (1) is that fewer sessions are in place.

The advantage of approach (2) is that it's more natural for the developer of the portlets (it doesn't need to use any key to store information in the session). It also alleviates the developer from the need to implement a custom expiration mechanism for the information stored within the session.

It is likely that developers will use approach (1) when the typical scenario is sharing of data between portlets, and use approach (2) when the portlets are designed to be segregated.

Conclusion 1: This decision how to use sessions is an implementation decision that is up to the portlet developer. WSIA/WSRP should not enforce one model or another.

Sharing between Group Portlets

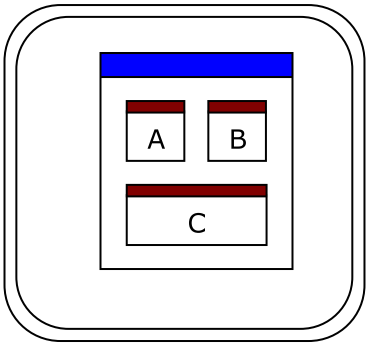


In some cases, portlets may want to share information regardless of the user session. For example, a cached record set may need to be shared.

In the illustration on the right hand side, A and B may be group portlets that may still need to share transient information.

We can ignore this problem as part of WSRP/WSIA. Namely, the portlets should use a custom-developed mechanism that is of no interest to us. However, portlets A and B may still need to know that they run in the same context.

Sharing between Group and User Portlets



Another scenario is that A and B are user portlets, but C is a group portlet, and information still needs to be shared. (E.g., a cached record set).

Conclusion 2: Data sharing between portlets seems to be orthogonal to the question of sessions. It should be possible without sessions.