WSRP - End User Identity

During the preceding email exchanges and the first security subcommittee conference call, 4 cases (levels) of passing the user information from the consumer to the producer emerged:

1. **Anonymous** - The portal does not send any information about the user to the portlet. This is useful when the content that is returned is not user-specific, and there are no issues of billing the end user by the consumer. A good example for this would be a news headlines producer.
2. **Identified** - The user's ID (and probably some elements of his profile) are sent to the portlet.
   One use case for this is a weather forecast producer that is personalized according to the user's zip code. The consumer could expose in its metadata that it requires the user's zip code, and the producer would pass that property to him. This requires us to define a group of standard profile properties.
   Two more use cases assume that the consumer and producer have the same notion of user identity (they use the same LDAP, Project Liberty service, etc., or the producer can identify the user from the profile). The first use case is that the producer (of any service) notes the ID of the user for billing purposes (how the billing vs. the user is actually done is outside the scope). The second use case is that the producer uses the user's ID to perform some action for him in a third system - for example ordering an item in some online store.
3. **Authenticated** - In addition to the user ID and profile information, a collection of {system name/user/password} is also sent. These passwords are used by the producer to authenticate the user to any other systems that it needs to get data from.
   A use case for this is when the producer is a servlet running over a back end application, and in order to get the data for the user, it needs to authenticate itself to the back end application with the user/password the user has in that application.
   A possible implementation for this is that during the bind phase the producer sends its public key to the consumer, and when getting the content the consumer uses that public key to encrypt the passwords. That ensures that only the intended producer can see what the passwords are.
4. **Authenticated with an intermediate** - The producer in this case is not the real producer; it is aggregating the results of another producer.
   A use case for this could be when the same servlet from 3 is exposed to a different consumer that is outside the firewall through an intermediate ASP in the DMZ that limits the kind of queries allowed.
   This raises security issues since the intermediate can see the data that was meant for the actual producer. Assuming this is only really a problem regarding passwords (after all the intermediate is still a trusted producer), the encryption method suggested in 3 still applies. The public key sent would then belong to the actual producer and not to the intermediate, so the intermediate cannot decrypt the passwords.