

Subscription Memory Service

Here are outlined two examples of the flow for a web application in the lines of “The solution to the Yahoo problem” post by Dave Winer (I’ve called it subscription memory service).

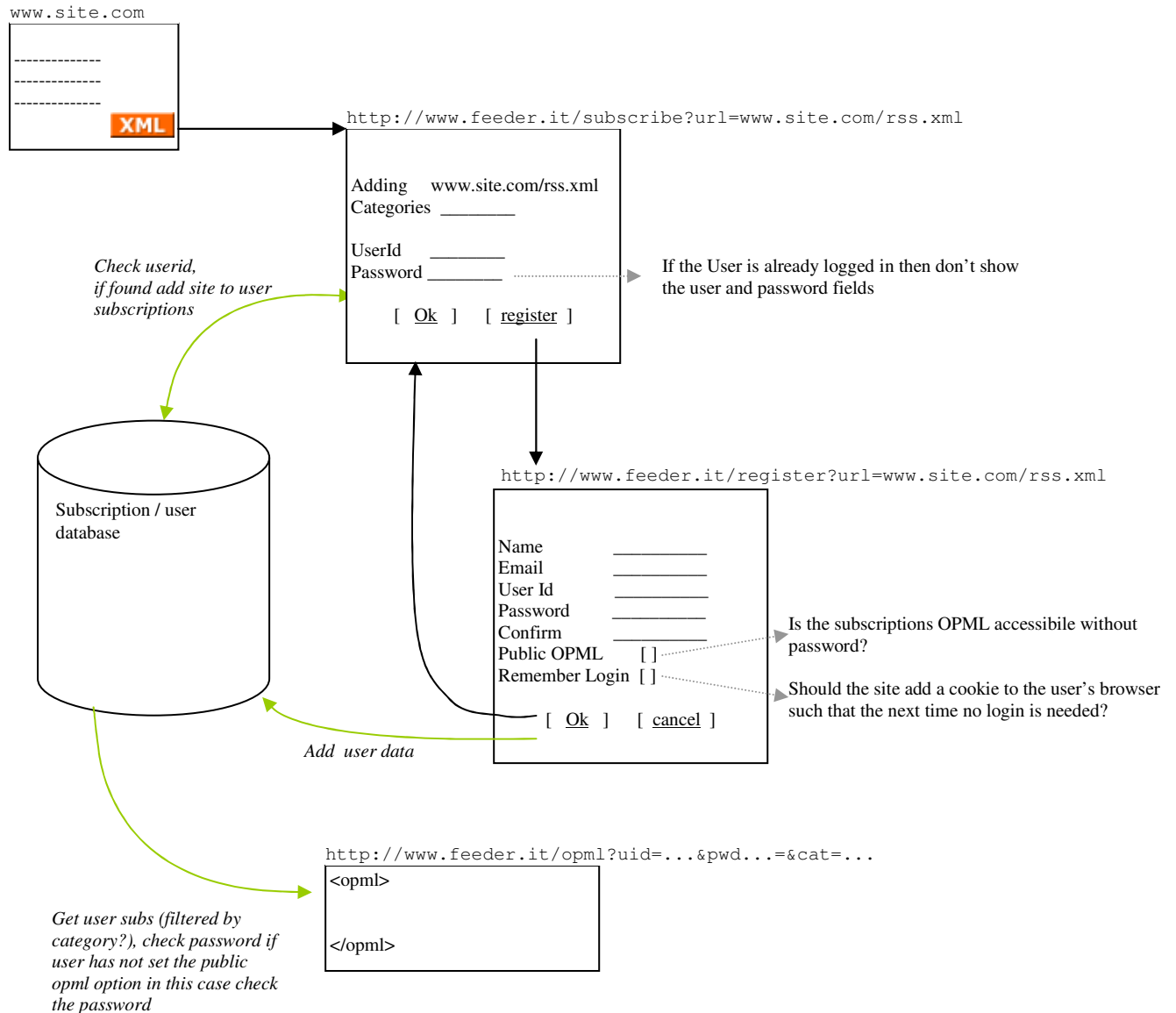
I have made a couple of additions to the initial Winer’s proposal:

- A check is added to the registration page asking the user if he/she wants to make the subscriptions OPML publicly accessible. If it’s not then anyone accessing the opml has to provide both the userid and the password.
- In the confirmation page I have added the ability to specify one or more category / keywords / tags for the feed being subscribed (which should be optional). This way the user’s subscriptions could be retrieved filtering for a specific category and the data base of tagged subscriptions would be a good start to identify the relevance of a feed about a specific topic (it would be a kind of *peer reviewed topic system*).

What I see as important as the reference software implementation is that the entry points (URLs, parameters) for the site be clearly defined in the specification for the service.

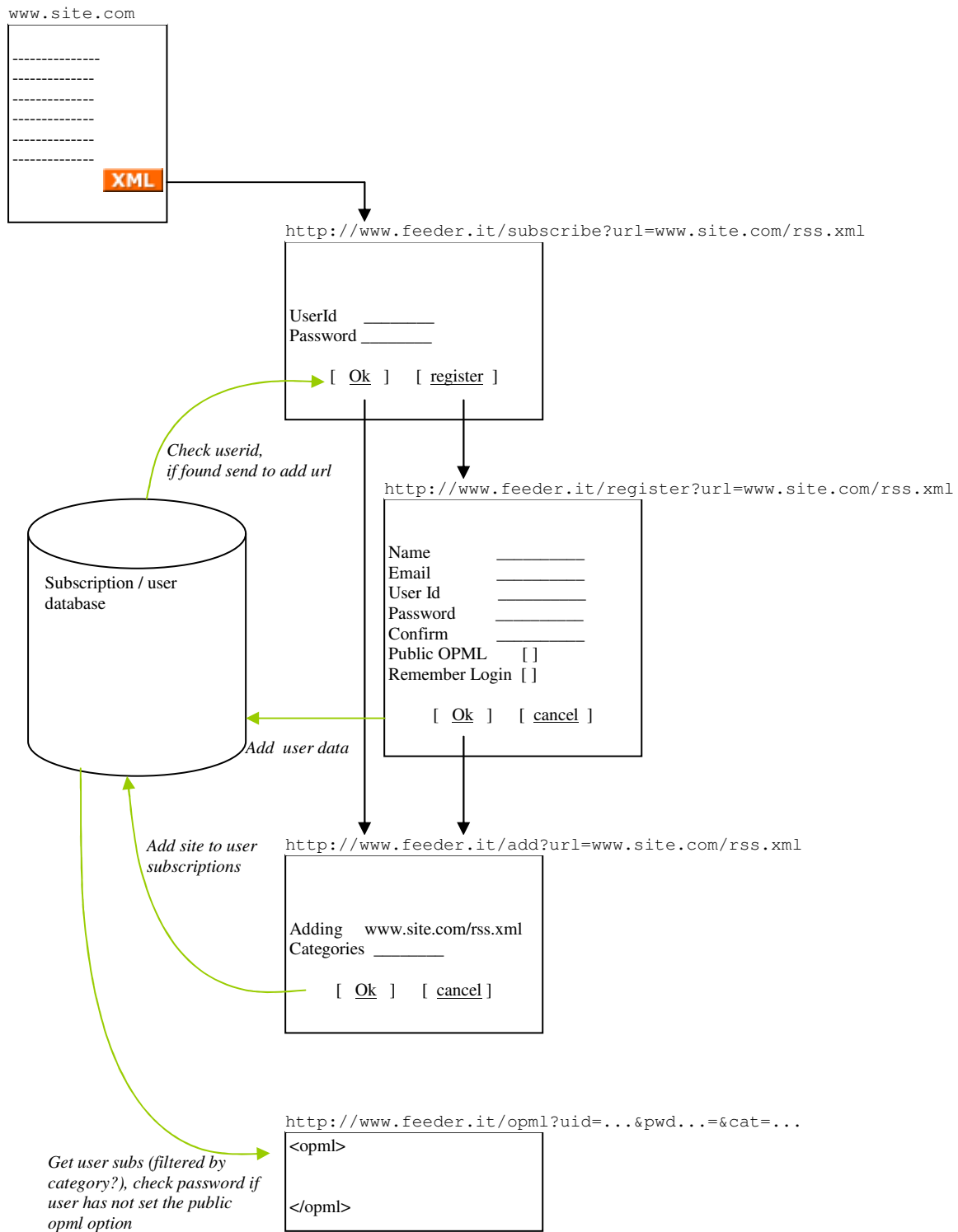
Note: below I have shown a database as the but it could be anything (if the category feature is not used it could even be a static repository of opml files updated by the subscription pages)

Example 1: just two pages to register / login / subscribe and one for the opml



Subscription Memory Service

Example 2: three pages for login, subscriptions confirm, registration plus a page for the opml



One important addition to such a service would be a pair of pages which give an explanation (for the not technical user) of what is rss, how aggregators can make reading easier, and even a series of links to the aggregators vendors. This infos should be directly accessible from the subscription page (which will be linked from the weblogs) such that a casual visitor may learn what this is all about.