When Changing the Server-side Code is Not an Option

Depending on your existing web application, modifying the existing server-side code to account for the mobile experience may not be an option. Some web applications contain a great deal of complexity and were built by teams that have since moved on to other projects. This situation sometimes makes it very difficult for the team maintaining a web application to make changes. If the application doesn’t have a consistent, well-thought-out architecture, or doesn’t contain unit tests, changes can be even more risky. When you couple this [unfortunately all-too-common] situation with a business justification that mandates an aggressive mobile strategy that “has to be available in weeks”, attempting to add a mobile experience to the existing implementation is a frightful proposition. Fortunately, there are solutions available to meet the business requirements, but as the engineering team, don’t expect to be thrilled with it.

## Objectives

* Describe the overall concepts of the approach
* Describe the advantages and disadvantages (should be mostly self-evident)
* Describe the challenges and limitations with taking this approach
* List some of the organizations who can help using a similar approach

## Notes

There are a number of very successful companies (with an impressive clientele) that have based their business model on these types of implementations. So this topic should not be disparaging. Try to avoid terms like hack, munge, screen-scrape, etc or put their use in context ... “proxy” is likely too ambiguous.

An engineering team would likely never favor this type of implementation since it permits the problem with the existing server-side implementation to remain, but strictly from a business justification perspective, this is an extremely viable option.

**Challenges**

* Potentially a LOT of regex
* JavaScript either isn’t used or requires much more work
  + Depends on the existence of a web API
* Images require [new] multiple sizes or a resizing service such as src.sencha.io