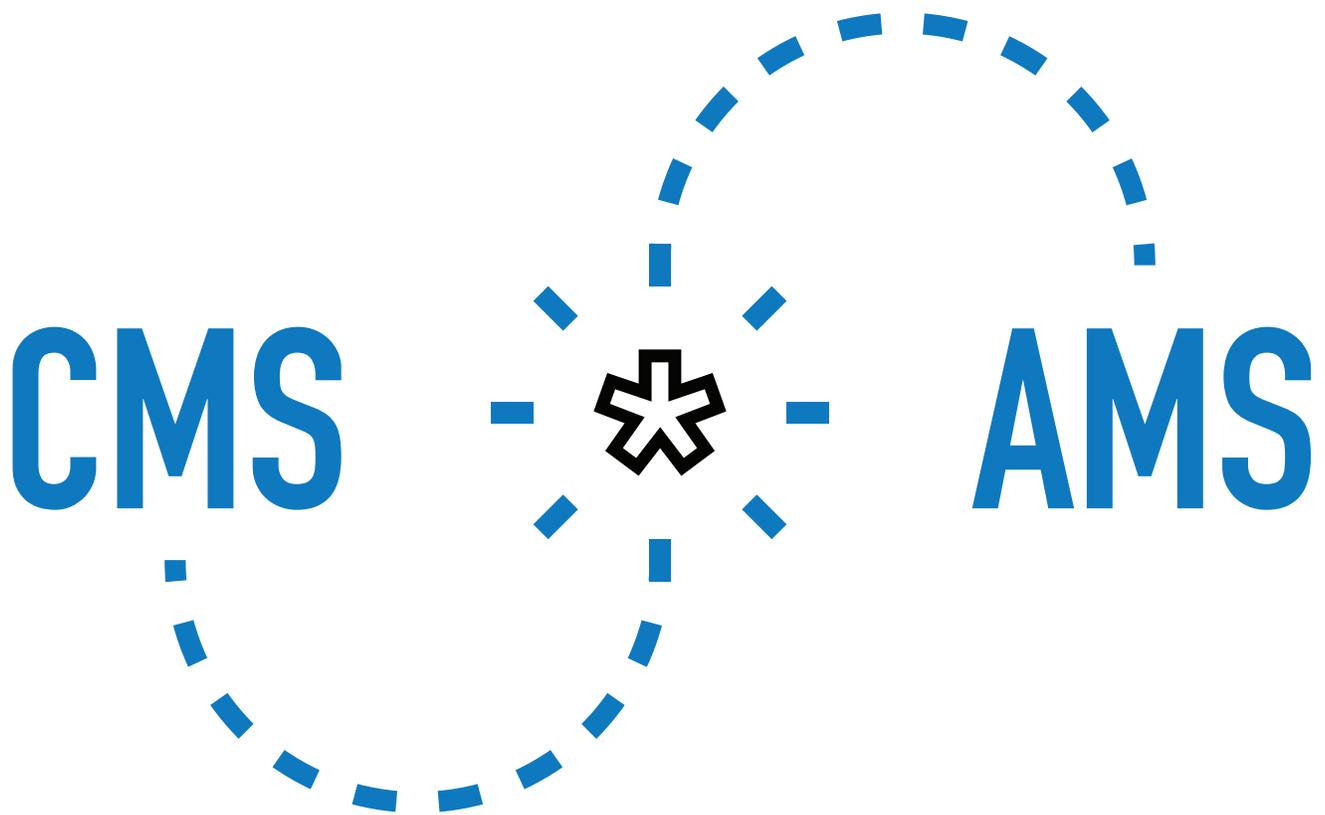


Successful CMS & AMS Integrations



Successful CMS & AMS Integrations

Associations rely on their websites and software tools to effectively serve their members needs. These needs include everything from providing member exclusive content to handling event registration and payment. Some associations will choose an Association Management System (AMS) to handle all of their online member needs. Others will make use of 'single purpose' tools such as a Content Management System (CMS) to manage their website.

Successful integrations are possible between nearly any combination of AMS/CMS tools, such as netFORUM/Wordpress or Drupal/ACGI. This paper covers some of the key planning considerations for an integration project. It will discuss the challenges, common features and advantages of these projects.

Standalone Content Management System (CMS)

Using a best in class web content management system for your website has a number of advantages, including:

1. **Rich Features** – General purpose web CMS solutions have a wide variety of modules and refined interfaces that allow staff to easily create content. Many common association website needs can be solved through simple configuration of existing features, including calendars, content rotators, content cross-promotion and implicit personalization.
2. **Staff Experience** – Popular CMS solutions are used by many organizations, both commercial and nonprofit with large communities of developers and users. This makes it easier to find staff and/or agencies that can design, build and run your website.
3. **Flexible Platforms** – Open source CMS solutions have modular architectures and flexible application program interfaces (APIs) that allow for developers/designers to create unique and experiences for your members.

Successful CMS & AMS Integrations

Challenges of Integration

While a standalone CMS offers compelling features to associations, sometimes the challenges of having two different tools that don't always play together nicely can be overwhelming. A poor integration between two otherwise excellent tools can lead to a poor member experience and reduced non-dues revenue. Poor integrations are obvious to members, including:

1. Members have to log in multiple times while trying to accomplish a single task leading to frustration.
2. Signing-in doesn't work or members can't get access to content they should, leading to frequent calls to your associations support line.
3. Online transactions such as dues renewal or event registration are overly complicated or are confusing leading to decreased sales.

On the other hand, a well-executed integration is invisible to members, since things *just work*. Getting to a seamless member experience takes careful planning, deep knowledge of both systems and indepth coordination between association staff, the AMS implementor and the web team.

Advantages of Open Source CMS

Open source CMS solutions such as WordPress, Drupal, and BrowserCMS are all excellent choices for association websites. One of the major advantages that open source tools have over commercial choices is their flexible nature. Developers are more able to change or tweak these tools in response to the very specific needs of the association. This is especially important when creating deep integrations between the CMS and AMS.

In many cases, these tools will have prebuilt modules to integrate with the AMS. For example, Drupal has an available netFORUM module (www.netforumondrupal.com) which can be configured for a given projects.

Successful CMS & AMS Integrations

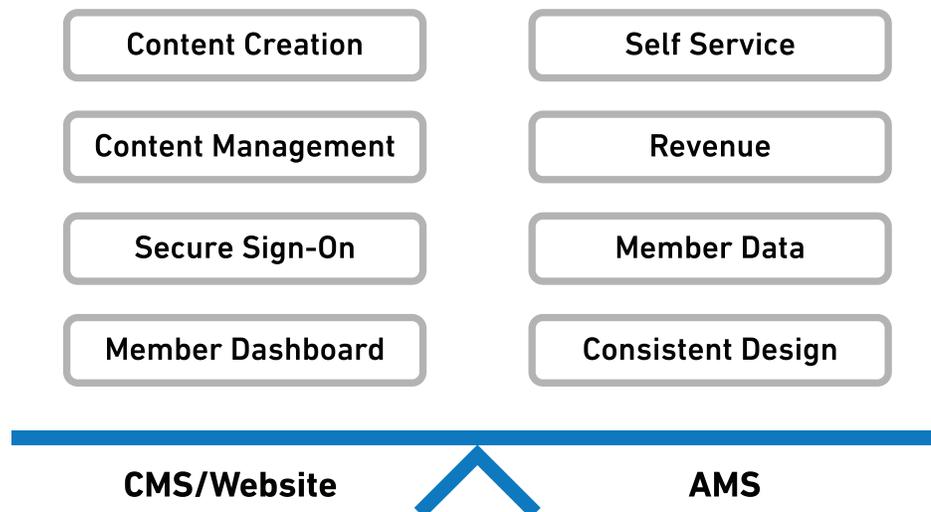
Benefits

A well-implemented integration uses the strengths of both tools to provide superior member experiences. After 10 years of developing successful AMS/CMS integrations for our clients, BrowserMedia has seen associations have the best success with the following division of responsibilities:

AMS Responsibilities

- **Self Service:** Members use the AMS member pages to handle self-service functionality like event registration and paying dues.
- **Revenue:** All revenue related features such as event registration pages or store products are handled via an AMS staff-only interface.
- **Authoritative source of member data:** Accounts (username/password) are managed in the member database along with membership levels.
- **Consistent design:** Public facing member pages are styled to match the overall website design.

One of the key requirements for a successful integration is that the AMS must provide robust APIs. Associations should talk with their AMS vendor (or prospective vendors) to ensure web service APIs (SOAP, JSON, etc.) are available for operations like authentication and member information.



Successful CMS & AMS Integrations

Website Responsibilities

- Efficient content creation: Web content and design assets (CSS/Javascript) are managed via the CMS user interface.
- Single sign-on: Members login on the website that authorizes them against the member database APIs.
- Member dashboard: Provide members with tailored/personalized links to key content/functionality for member experience.

Beyond these basic responsibilities, deeper integrations are possible through the use of the AMS APIs. These features could range from creating detailed social member profiles to 'Find a Member' functionality.

Basic Integration

A standard AMS/CMS integration will have at least these key pieces:

1. Single Sign-On (SSO) - Ensures members only need to log in once, regardless of how many systems are connected.
2. Authorization - Validating what members are allowed to do once they have been authenticated.
3. Consistent Design - Ensures all web pages visited by members use the same design styles, whether served by the CMS or the AMS.

SSO makes use of web services as well as connects to the public member pages to handle login. For a netFORUM integration, this means using the SOAP-based xWeb APIs as well as eWeb member pages. For an ACGI integration, this would mean using the Oracle Stored Procedure or HTTP Relay APIs, depending on which features are needed.

Successful CMS & AMS Integrations

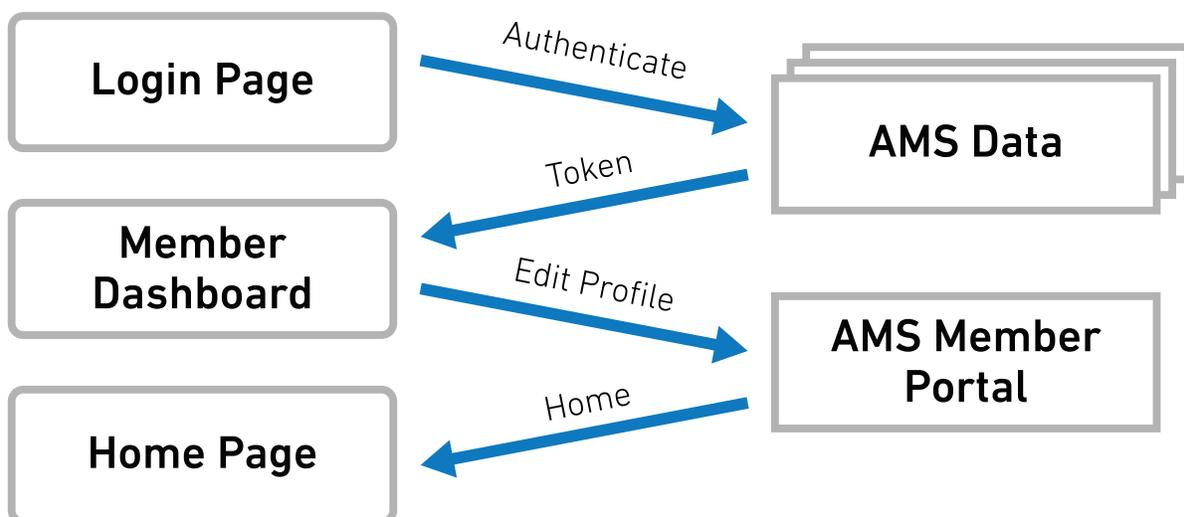
1. Single Sign On

Here is the basic outline of how SSO works.

1. A member comes to the website (served by CMS), and login using their member username/password.
2. The CMS calls API to authenticate the user, passing the user account and any API authentication keys.
3. If successful, the CMS marks in the member as authenticated, and redirects the member to the member pages. Typically the API will provide a token that needs to be passed to the member pages to complete authentication.
4. At this point, the user has authenticated sessions with both the CMS and AMS and can travel between pages on either system.

It's important that the AMS be configured to have the website handle the login, otherwise users can have inconsistent experiences. For example, suppose a visitor is browsing the store pages served by the AMS and then tries to checkout. When they log in order to get the 'Member' price, if the CMS doesn't handle the login they will find themselves just logged into member pages but not the CMS.

Pro Tip: Use Chrome's 'Incognito Window' browsing feature to test authentication. Since no cookies are retained when you close the window, this mode makes it easy to consistently verify logins.



Successful CMS & AMS Integrations

2. Authorization

Many associations will use their AMS tool's Customer Relation Management (CRM) module to track members at different levels of membership (Student through Corporate), nonmembers or prospective members. It's probable that not all of these individuals should have the same access to content in the CMS.

Once a user has been authenticated, the CMS will need to determine which 'group' that member belongs to for the purposes of accessing content. Associations should discuss how members are segmented with their website implementor, whether its just 'members vs. nonmembers' or more something more complex (e.g. U.S. citizens eligible to donate to political action committees). As a general rule, start with the minimum possible groups and then determine what each group is allowed to do.

Working against the APIs, the CMS should query for more individual information. It can then assign members to the necessary group so that users can see the content they are allowed to. The CMS will be configured to determine what the different groups can see.

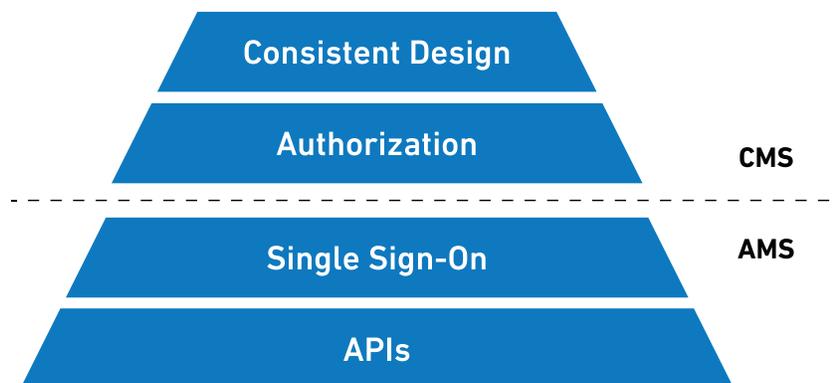
If the authorization logic is particularly complex (requiring multiple API queries), associations should consider creating a custom API. Not all AMS solutions will support this; for example, netFORUM Enterprise supports custom web services but netFORUM Pro (currently) does not.

3. Consistent Design

The attractive design for the association website should carry through to the member pages as much as possible. The HTML & CSS assets should be applied in the member pages interface to style the key member pages. In most cases, it will be necessary to make tweaks to accommodate web elements present on the member pages.

This does result in some duplication of styles as they are copied into the AMS, but is less costly than handling member self service through the APIs. AMS solutions tend to differ greatly in how they handle their member pages, using everything from built in CMS-lite tools to providing iframe-based content.

Successful CMS & AMS Integrations



User Management

Each CMS will handle user authentication/authorization in slightly different manner, though the general integration principles are the similar regardless of the CMS or AMS involved.

Drupal

Drupal has several AMS modules available for some systems, but for other AMS solutions it may be necessary to create a custom module. Typically, a Drupal AMS module will use external user authentication, which creates a record in the CMS database for each user. This creates some amount of duplication between the CMS and AMS, but allows for additional 'website profile' information to be stored in Drupal. For instance, if an association wants to take advantage of Drupal's community/social features, additional profile (i.e. Image Galleries, Member Bio, etc.) data stored in Drupal can be linked to the user record. Member information can be 'cached' in Drupal for greater redundancy in case the AMS server is unavailable.

Keep in mind that there may be constraints for given AMS modules. For example, with the Drupal/netFORUM module, one constraint is that both eWeb and Drupal need to be on the same domain (e.g. netforum.example.com and www.example.com) since the solution uses cookies for SSO.

BrowserCMS

BrowserCMS AMS modules typically use temporary users that exist for the duration of the session. This avoids the need to duplicate information in the CMS database, but means there is greater dependency on the availability of the AMS for member login.

Successful CMS & AMS Integrations

BrowserCMS handles the redirects between the CMS and member pages, so SSO does not rely on both sites being on the same domain. Logged-in members will not appear in the BrowserCMS member interface. Group permissions (i.e. which members can do what) are configured through the user interface.

Planning/Budgeting Considerations

Getting an integration right is tricky due to the complexity and required communication between teams. Here are some things to keep in mind in order to ensure a smooth project.

1. **Coordination Time:** Integrations always take longer than teams might expect due to the back and forth conversation between 2-3 teams. Start early, and allow enough calendar time to iterate toward the completed integration.
2. **Design Integration:** Ensure budget/responsibility for applying the website design to member pages is accounted for. Ensure it's clear if the web CMS team should also be styling these pages, if the AMS implementor is handling that, or if both teams will need to work on it.
3. **Groups:** What member roles do you track in the AMS and what areas of the CMS should they have access to?
4. **Testing Environment:** Ensure testing AMS and CMS servers are available during the integration time so that the CMS can be tested safely without affecting 'production' data.
5. **Licensing:** Many AMS solutions will require additional licensing to implement SSO, often for access to the APIs. For example, netFORUM customers need to have at least netFORUM Pro or Enterprise versions, since Team doesn't support xWeb, as well as license for xWeb.
6. **Timeouts:** Ensure that session timeouts on both the CMS and the AMS are the same. If the CMS has a longer timeout, a member could experience problems accessing AMS member pages.

Successful CMS & AMS Integrations

Conclusion

Associations need effective websites to serve their membership. A well-implemented CMS/AMS integration helps them do that, while a poor one creates extra work for staff and/or bad experiences for members. Integrations are challenging even for experience practitioners. This whitepaper has endeavored to set forth the key considerations for a successful AMS/CMS integration, with the goal of helping associations better serve their members.

About BrowserMedia

BrowserMedia is a full service creative design agency and has been building websites and AMS integrations for over 12 years. We are an official Auctra Industry Solutions Partner and have successful integrations with multiple AMS tools, including iMIS, netFORUM, Personify, ACGI and Euclid. BrowserMedia understands the day-to-day challenges that associations face.

We provide innovative solutions to associations through seamless integrations to help them better serve their members.

See our portfolio at BrowserMedia.com/work and do contact us at sales@browsermedia.com or 877-468-3769.