


# Front-end - Research - CMS - CMS for 2017+

## Decision Made

Craft CMS was chosen! See [Third-Party Software - Craft CMS - Frontend](#).

## Useful Resource

There is a great website for researching static CMS website generators: <https://www.staticgen.com/>. This was found  10 Sep 2018 , and would have **definitely** streamlined the research, but only **maybe** could have changed the decision to use Craft CMS.

## Requirements

### Must Have

- CMS
  - WYSIWYG
  - page management
  - blog functionality
  - user management
- support (for developer and user)
  - **must not** be a one dev team
- static content only
  - **either** static site generator
  - **or** discipline to ensure content is static
- multilingual
  - **either** markdown
  - **or** language files
  - **avoid** markup
- WYSIWYG
  - Markdown is **desirable**
    - because translation will be easier
  - HTML is **acceptable**
- versioning / rollback
  - database
  - version control
- user acceptance
  - authors
  - marketers

### Should Have

- static files
  - **either** CMS has ability to export static website
  - **or** CMS by design has static content directory
- flexibility for design/dev
- year(s) of history
- automated translation
  - developer work is expected

### Could Have

- plugin support
- integrations
- large community
- designer flexibility
  - ability to **use** patterns (UI modules)
  - ability to **use** templates

## Ignored

- user install of plugins
  - this sounds dangerous; there should be a process
- zero-dev design
  - ability to **create/edit** patterns (UI modules)
  - ability to **create/edit** templates

## Feature Matrix

Key					
Excellent	Acceptable	Possible	Unacceptable	Unclear	Unknown

		Wordpress	Siteleaf	Grav	Pagekit	Craft
Basics	Pages	Yes	Yes	Yes	Yes	Yes
	Blogs	Yes	Yes	Yes <sup>1</sup>	Yes	Yes
	Plugins	Yes !	Developer <sup>1</sup>	Yes	Yes	Developer
Support	Docs	Yes	Yes	Yes	Yes	Yes
	Community	Yes	No	Yes	Developer	Yes
	Proprietary	Yes	Blog	Yes	Yes	Yes
I18n	Multilingual Website	Plugin <sup>1</sup>	<ul style="list-style-type: none"> <li>• Developer <sup>2</sup></li> <li>• Flatfiles <sup>2</sup></li> <li>• Plugin <sup>2</sup></li> </ul>	<ul style="list-style-type: none"> <li>• CMS</li> <li>• Flatfiles !</li> </ul>	Future <sup>1</sup>	CMS <sup>1</sup> PHP <sup>1</sup>
	Outsourced Translation	Plugin <sup>1</sup>	Developer	Developer	Not Yet	<ul style="list-style-type: none"> <li>• Developer <sup>1</sup></li> <li>• Translator Login <sup>1</sup></li> </ul>
Content	WYSIWYG	<ul style="list-style-type: none"> <li>• HTML <sup>2</sup></li> <li>• Markdown <sup>2</sup></li> </ul>	Markdown	Markdown !	<ul style="list-style-type: none"> <li>• HTML</li> <li>• Markdown</li> </ul>	• HTML <sup>2</sup> !
	SEO <sup>1</sup>	Yes Plugin !	Plugin <sup>1</sup>	Basics <sup>2</sup>	Basic	Yes Plugin !
	Revisions	Yes	Github	Github	<ul style="list-style-type: none"> <li>• Github</li> <li>• Future <sup>1</sup></li> </ul>	Yes
Static	Content	Plugin	N/A	Discipline	Discipline	Plugin <sup>3</sup>
	Files	Plugin <sup>3</sup>	Yes <sup>3</sup>	No	No	No
	AJAX Forms	Plugin	Developer	Plugin <sup>3</sup> Developer <sup>3</sup>	Plugin ! Developer	Tutorial Developer
Versioning	Data	Database	Database	Files	Database	Database
Rollback		Files	Files		Files	Files
Backup	Process	<ul style="list-style-type: none"> <li>• SQL / VCS</li> <li>• Plugin</li> </ul>	Manual <sup>3</sup>	<ul style="list-style-type: none"> <li>• VCS</li> <li>• CLI</li> </ul>	Manual	<ul style="list-style-type: none"> <li>• Native</li> <li>• Plugin</li> </ul>
Design	Use Templates	Copy Page	Copy Page	Copy Page	Copy Page	Yes !
	Use Modules	Plugin	Developer	Yes	Yes <sup>2</sup>	Yes !
	Set Layout / Theme	Hack	Yes	Yes	Yes <sup>3</sup>	Yes !
	Custom CSS	Plugin	Yes	Plugin	<ul style="list-style-type: none"> <li>• Gross</li> <li>• Future <sup>1</sup></li> </ul>	Plugin
	Custom Fields	Plugin	Yes	Technical	Future <sup>1</sup>	Yes !
Users	User Roles	Plugin	Yes	CLI	Yes !	Yes !

	User Permissions	Plugin	Yes	No	Yes !	Yes !
	Login as User	Plugin	No	No	No	Yes
	Comments (3rd Party)	Plugin	Developer	Plugin	Developer	<ul style="list-style-type: none"> <li>Developer</li> <li>Tutorial</li> </ul>
Analytics	Internal	No	No	Yes Plugin	Plugin	No
	3rd Party <sup>2</sup>	Plugin	Developer	Plugin (Google)	Plugin (Google)	<ul style="list-style-type: none"> <li>Plugin (Google)</li> <li>Plugin (Piwiks)</li> <li>Cool Plugin (Google)</li> </ul>
Technical Requirements	Database	<ul style="list-style-type: none"> <li>MySQL 5.6+</li> <li>MariaDB 10.0+</li> </ul>	N/A	N/A	<ul style="list-style-type: none"> <li>MySQL 5.1+</li> <li>SQLite 3+</li> </ul>	MySQL 5.1.0+
	Language	PHP 5.6+	Ruby 1.9.3+	PHP 5.5.9+	PHP 5.5.9+	PHP 5.3.0+
	Hosting	Any	<ul style="list-style-type: none"> <li>Github</li> <li>Amazon S3</li> <li>Rackspace</li> <li>FTP / SFTP</li> <li>via API</li> </ul>	Any	<ul style="list-style-type: none"> <li>Apache 2.2+</li> <li>nginx</li> </ul>	<ul style="list-style-type: none"> <li>Apache</li> <li>Nginx</li> <li>IIS</li> </ul>
Developer	CLI	3rd Party	Yes	Yes	Yes	3rd Party
	API	3rd Party	Yes	No	No	Not Yet
	Statistics	<ul style="list-style-type: none"> <li>Plugin</li> <li>3rd Party</li> </ul>	3rd Party	No	Yes	No
	Cache	Plugin	Hosting-Dependent	Plugin	Yes	Plugin (Beta)
Pricing	Recurring	N/A	\$0 / \$7 / \$30 / \$90 / \$300+	N/A	N/A	N/A
	One-Time	See Plugins	N/A	N/A	N/A	\$0 / \$200 / \$300
Notes			Import CMS content CMS is a service		Security	

## Analysis

### Service Recommendation

#### WordPress

**WordPress** has everything; but users have more power to make bigger mistakes. **Craft CMS** has everything; but users and developers will have a learning curve. Mistakes (with rollback) are more desirable than a learning curve. **Siteleaf** is too technical.

#### Known Defects

##### CrowdIn WordPress Integration

1. Pre-integration source language content is available on translated site.
2. Unable to switch language of admin of translated site to source language.
3. ~~Text fields added by Advanced Custom Fields plugin are not translatable.~~
  - The cause of the problem was user error.
4. Image fields added by Advanced Custom Fields plugin are broken, missing, or incorrect.

☒ Awaiting e-mail response from CrowdIn (bug reports sent by Unknown User (wesleyb)).

##### Polylang WordPress Plugin

1. Media fields are translated, but do not show up on front-end.

## Service Suggestions

### WordPress - Grandfather CMS

WordPress meets all requirements, but it does so mostly with plugins. The inherent independence of plugins offer a feature done well; but can introduce plugin or version conflicts and inconsistent admin interface. Design through the CMS is supported through a powerful and popular plugin. WordPress does not support Markdown, so users can easily ruin the HTML. Translation process has excellent plugin support. WordPress is free, as are most of the plugins. Users can install plugins—a process and discipline must exist.

#### Pros

- Familiar software
- Plugin(s) for automated translation process
  - *Designer empowerment content is not translated or copied—and configuration is unreliable.*
- Designer empowerment (via plugin)

#### Cons

- Plugin-based feature set
- HTML can be ruined by users

#### Mixed

- Plugin install by users

### Craft CMS - CMS For Designers

Craft CMS meets all requirements, and it does so mostly with native features. Design through the CMS is powerful, but there is a great learning curve for designers and developers. Director of Design introduced this CMS. Craft CMS does not support Markdown, but uses an HTML "purifier"; custom scripting is required to automate the translation process. Craft CMS has a one-time price tag. Plugins must be installed by developers.

#### Pros

- Native feature set
- Markdown is available (native and extensions via plugin)
  - *Also, the HTML, if used, is "purified".*
- Designer empowerment (HTML stays clean)

#### Cons

- Learning curve
- Scripting required for automated translation process

#### Mixed

- Plugin install by developers

### Siteleaf - CMS As A Service

Siteleaf meets all requirements, but requires developer effort. Siteleaf uses flatfiles and generates a static file website; the CMS is proprietary. Usage and creation of modules requires developers to reinvent wheels that other CMSs have built. Siteleaf supports Markdown, which makes translation easier; a translation plugin is available, but untested. Siteleaf has a recurring price tag. Plugins must be installed by developers.

#### Pros

- Markdown is supported
- Flatfiles; no database
- Static file website by design

## Cons

- Developer-powered feature set
- Plugin install is very technical

## Mixed

- Scripting required for automated translation process, but it is minimal

## Other Services

### Grav - Flatfile CMS

Grav meets most requirements, but requires developer effort. User management is very limited. Grav has no database and uses flatfiles. Design through the CMS is possible but not intuitive. Grav supports Markdown, which makes translation easier; custom scripting is required to automate the translation process. Grav is free. Users can install plugins—plugins are few and simple, but installation, setup, and use is clean and reliable.

### Pagekit - Future Wordpress Killer

Pagekit has phenomenal potential, but it does not yet meet a major requirement—multilingual support. Developers should monitor its progress; one day it will meet all the requirements.

## Incomplete Evaluation

- **Kirby** - "A flat-file CMS that's still fully dynamic". Static file generation is not the goal.
  - Feature set looks very good.
  - Nicole had trouble setting up the CMS.
  - Wesley has not tried yet.
- **Perch** - Retrofit a CMS onto an existing static website. Static file generation is not offered. Trial [purchase](#) (refund available) to test fully.
  - [Perch: Buy](#)
    - *"If you buy a license and Perch or Runway isn't the right fit for you, just drop us a line within 7 days and we'll refund you."*
  - [Perch: Forum: Perch SSG \(Static Site Generator\)](#)
  - [Perch: Forum: Static Site Generation](#)

## Evaluation Considered

- **Statamic**
- **Pyro**
- **CloudCannon**
- **Sitecake**

## Not Evaluated

- **Lektor** - Too simple.
- **PieCrust** - Support team is just one guy.
- **Pico** - Support team is small. History is brief. It's a glorified file editor.

## Footnotes

### Generic

! Any feature cell entry followed by " ! " means that the feature is done very well.

Some CMSs or CMS plugins store manual translations in a database. A combination of scripts and plugins could:

1. Extract English content from database.
2. Convert English content from CMS format to translation format.
3. Send/Receive content to/from translation service in translation format.
4. Convert language content from translation format to CMS format.
5. Import language content into database.

## Features

<sup>1</sup> SEO content:

- Page/post metadata and slugs.
- Site metadata and media.

<sup>2</sup> Third-party analytics:

- Usually a third-party requires a certain Javascript snippet be added to the site.
- A value " Yes " means that Analytics is supported in this way through the CMS.

## WordPress

<sup>1</sup> Multilingual plugins:

- The [The WordPress Multilingual Plugin](#) was **purchased** and is installed (*but not in use*) on the DataFoundry blog.
  - **Untested.**
- The [Crowdin WordPress Integration](#) plugin **delegates** translation to [Crowdin](#).
  - This plugin is in Beta.
    - *Revealed in response to e-mail inquiry sent by [Unknown User \(wesleyb\)](#).*
  - This plugin would couple us to Crowdin.
- The [Polylang](#) plugin ([how to article/video](#)) requires **manual** translation, but this can be **delegated** or **automated**:
  - Through Polylang, there is a [Lingotek WordPress integration](#):
    - [Lingotek](#), through their [Lingotek WordPress plugin](#), offers language editing delegation to:
      - employees
      - machines
      - community
        - this option allows us to use Speaklike or One-Hour for translation.
      - their own professional translation service
        - this option allows us to use Lingotek for translation
    - There are [some 5-star reviews](#), but the following problems have been reported.
      - [Unexpected Usability Issues](#)
      - [Plugin Conflicts & HTML Translation Bugs](#)
      - [minority of various publicly unresolved support issues](#)
    - *In-house scripting required to automate this. See **Footnotes Generic Language**.*
      - *Supposedly, [Bulk Translation with Polylang](#) is made easier with PHP (review [Polylang: Function Reference](#)).*
- The [WordPress Multilingual Plugin](#) tackles both multilingual management and the translation process itself.
  - *WPML is the granddaddy of WordPress multilingual plugins.*
  - *Polylang offers conversion from WPML to Polylang.*
- For more options and critiques, see [Translation Management and Collaboration Platforms for WordPress Developers](#).

<sup>2</sup> Markdown plugins:

- The [Jetpack plugin Markdown](#) feature saves content as HTML.
  - Users can still input/edit Markdown.
  - Translations will be 100% in HTML.
- Because language plugin(s) would be used, it is strongly recommended to avoid excess user-generated HTML.
  - *Using "Modules & Custom Fields" can limit and organize user-generated HTML.*

<sup>3</sup> Static site plugins:

- The [Simply Static](#) plugin has good reviews, clear honest documentation, and one job.

User permissions:

- Consider the [Gravity Forms integration](#) which integrates with [User Role Editor](#) (WordPress plugin).



Research from [Unknown User \(nicolee\)](#).

I recommend using the following: [User Role Editor](#) (WordPress plugin)

We would need the Pro version, but comes with various license fees, and has granular support for user permissions.

- It is unclear whether User Role Editor offers the ability to login as another user.

Modules & Custom Fields plugins:

- The [WordPress Toolset plugin](#) is advertised by the [The WordPress Multilingual Plugin](#).
  - **Untested.**
- The [Advanced Custom Fields](#) plugin is highly recommended.
  - **Pay money.** Get the pro version to cover all use cases: [Advanced Custom Fields PRO](#)
    - [Unknown User \(wesleyb\)](#) found uses cases for Pro while preparing CMS demo.

☑ Implementation test from Unknown User (nicolee).

☑ Compatibility with language plugins; research/testing from Unknown User (wesleyb).  
[Polylang](#): **Tested**. Also, a reply to a WordPress plugin support forum post verifies compatibility.  
[WPML](#): **Untested**. An article by ACF "Multilingual Custom Fields" explains how to make ACF plugin work.  
[Crowdin](#): **Tested; has defects**. The plugin is still in beta.

Layouts feature:

- We could use a "Custom Field" as an identifier for a layout.

## Siteleaf

<sup>1</sup> Plugin installation:

- The [Siteleaf Jekyll Plugins page](#) explains that plugins are installed via [Jekyll](#).
- The [Jekyll plugin documentation](#) explains how to do this.
- Plugin installation requires knowledge of [Ruby gems](#).

<sup>2</sup> Multilingual support:

- A [Jekyll Multiple Languages Plugin](#) is available.
- Development and/or proprietary support is required. **Not** available through the CMS.
  - [Gist of How to Do This](#)
  - Official answers:
    - [via Twitter in January 2014](#)
    - [via Twitter in October 2014](#)
  - [Feature request for Jekyll](#) (Siteleaf is powered by Jekyll)

<sup>3</sup> Untested support for several technical features:

- See "**Hosting options**".
- [Versioning via Github](#)
- [Static Site Hosting](#)

Module feature hack:

- Developers can introduce and manage modules with partials.
- Developers could allow modules by combining custom fields and layout:
  - Page has custom fields prefixed with `module_`.
  - The layout chosen for the page knows how to display each `module_` custom field.
  - The order of the `module_` custom fields could be used to arrange their display.

Metadata support:

- Siteleaf handles extra content in an explicit and flexible manner. Any extra content for a page:
  - **may** be defined by an editor
    - via [smart field names](#), content will automatically support lists, media, text, data, etc
  - **may** be displayed by an editor (content)
  - **may** be displayed by a developer (template)

Plugin requirement:

- Plugins are [Ruby gems](#), which is a package manager for [Ruby](#).

Hosting options:

- A single site can be [hosted on the cloud](#) via:
  - Siteleaf
  - Github Pages
  - Amazon S3
  - Rackspace

- FTP / SFTP
- Duplicate sites are suggested to use the API.
- Cache should be managed by hosting solution.

 **Awaiting explanation from Unknown User (rafaelf) (2016-08-12)**

CLI feature:

- The CLI is a Ruby gem for API actions

## Grav

<sup>1</sup> Blog feature:

- Distinguishing blogs from pages requires discipline:
  - Add a folder (home page) for blogs.
  - Add a template for blogs.
  - Fill out "Options" section in Page Editor (publish, date, category, tags).

<sup>2</sup> SEO features:

- Only the most basic SEO meta data may be entered through the CMS.
- Metadata names must be known:
  - Most CMS's let you enter common metadata without reference to the fact that it is metadata.
  - This CMS expects users to know what names the metadata values are assigned to.
    - We can solve this by creating a page template to be copied.

<sup>3</sup> Form support:

- The form plugin adds dev form settings to CMS.
- A form template can be made for form pages.
- Forms need to be created by developers.
- Basic editing of forms can be done by user, but should be done by developer.
- Forms suggest the use of YAML "Frontmatter" ("Expert" edit mode of page).

Design features:

- Templates: No specific support, but a special section of hidden pages can serve as templates:
  - One must redefine repeat modules and order for each page.
  - One can copy module parent pages to do this easily.
  - One can prepare module parent pages to serve as copiable "templates".
- Modules: A learning curve exists:
  - Pages containing module children need Page Template set to "Modular".
  - Module child pages need a name prefixed with underscore.
  - Module child pages need Page Template set to "..." (choose from list—see "Layout / Theme").
  - Value of "Page Template" changes based on whether page is or is not a module child.
- Layout / Theme:
  - See "Display Template" and "Body Classes" in [Page Editor \(advanced options\)](#).
  - Read into CMS as item in "Page Template" selection if template file exists.

## Pagekit

<sup>1</sup> Future features:

- Custom Fields feature is on the [PageKit Roadmap](#) as #119.
- Revisions feature is on the [PageKit Roadmap](#) as #4.
- Multilingual feature is on the [PageKit Roadmap](#) as #173.



<sup>2</sup> Modules feature:

- Modules can be added above and below content in developer-defined locations.
- Modules can **not** be used **within** content.

<sup>3</sup> Layouts feature:

- We could use the "HTML Class" as an identifier for a layout.

Custom CSS entry:

- The editor can add this with HTML in the content area via `<style>...</style>`.

Developer features:

- Statistics are available by checking "Enable debug toolbar" in "System" "Info" section.
- Cache options are available in "System" "Settings" "Cache" section.

## Craft

<sup>1</sup> Multilingual support:

- Multilingual support takes some work to setup properly:
  - [Craft CMS: Setting Up a Localized Site](#)
  - [Why we use Craft CMS: easy multi-language support](#)
- Automating the translation process is possible:
  - **Automated & Secure Solutions**
    - Database content export/import via scripts and plugins and translation service (see ).
      - [Front-end - Craft CMS - Research - Translation - Via CMS Interface](#)
    - Code strings (PHP array) via scripts and plugins and translation service (go to Hell).
      - [Craft CMS: Setting Up a Localized Site: Step 7: Update your templates](#)
      - [Craft CMS: Translating Static Text](#)
  - **Manual & Insecure Solutions**
    - Allow third-party translator to login with limited access back-end account.
      - Create language-specific user groups for translators that can edit certain sections (groups of pages).
      - The [Locale Sync plugin](#) allows updating content in other locales after updates in source locale.

<sup>2</sup> Content editing:

- **HTML**
  - An [HTML Purifier](#) is integrated into the content save. It can be turned off, but it shouldn't. It does a good job.
- **Markdown (no WYSIWIG)**
  - There is a native [Markdown template Twig filter](#) `{{ ... | markdown }}`.
  - There is a [MultiMarkdown plugin](#): "Smartdown" that brings [Markdown Extra](#) and [SmartyPants](#) via [Twig filter](#) `{{ ... | smartdown }}`.
  - There is a [Markdown plugin](#): "Doxter" that brings [Markdown](#) via [Field Type](#).

<sup>3</sup> Static content:

- There are three options for caching Craft CMS content:
  - [Fastly](#) (plugin integration to paid service)
  - [Varnish](#) (plugin integration to open source tool)
  - [HTML Static Cache](#) (beta plugin)

Form tutorials:

- See the [Entry Form Example from Craft CMS](#); below it are more form examples.

Design features:

- Creation of custom fields, layouts, templates, modules, etc are all possible through the CMS—developers, technically, are **not** required.
- There is a **steep** learning curve for implementation and a usage.

Requirements:

- See [Craft CMS Requirements](#).
- Content **must** be hosted in the root directory.

## References

### General

Reviews:

- [Choosing The Best CMS \(Blog Post\)](#)

## CMS Specifics

### Craft

Articles:

- [Wrapping Your Head Around Craft Sections and Entry Types](#)

### Grav

Plugins:

- Language Switcher
- Grav CDN
- Staticfilecache