



Nicolas Hafner
@Shinmera

<https://everything.shinmera.com>

Web Frameworks

- HTTP Server
- Database
- Handling of common problems
- Template rendering
- User accounts, authentication

Web Frameworks

- HTTP Server
- Database
- Handling of common problems
- Template rendering
- User accounts, authentication
- Potentially everything in computing too

About Seven Years Ago



About Seven Years Ago

- Comics with comments
- Forums for discussions
- How not to require users to have multiple logins?

About Seven Years Ago

- Comics with comments
- Forums for discussions
- How not to require users to have multiple logins?
- Write my own software

About Seven Years Ago

- Comics with comments
- Forums for discussions
- How not to require users to have multiple logins?
- Write my own software ... in PHP
- Evolved over time, eventually I got to Lisp

Radiance Goals

- “Do it right this time”
- Run multiple applications side-by-side
- Share resources between applications
- Make as many parts optional as possible
- Configurable by administrator
- Documented well and easy to use

Interfaces

- Standardise access to common features
- Load interface implementation on demand
- Weak contract to ensure consistency

Interface Definition

```
(define-interface cache
  (defun invalidate (name)
    "Causes the cached value to be re-computed.")

  (defmacro with-caching (name test &body body)
    "Caches the return value if TEST is non-NIL."))
```

Interface Implementation

```
(defvar cache::*caches* (make-hash-table))

(defun cache:invalidate (name)
  (remhash name *caches*))

(defmacro cache:with-caching (name test &body body)
  (once-only (name)
    `(or (and (not ,test) (gethash ,name *caches*))
        (setf (gethash ,name *caches*)
              (progn ,@body)))))
```

Interface Usage

```
(asdf:deftsystem another-todo-app
  ...
  :depends-on ([:interface :cache]))

...

(in-package :another-todo-app)

(defun render-todo-list (user)
  (cache:with-caching (user (changes-made-p user))
    (render-template (template "todo-list.html")
      (fetch-todo-items user))))
```

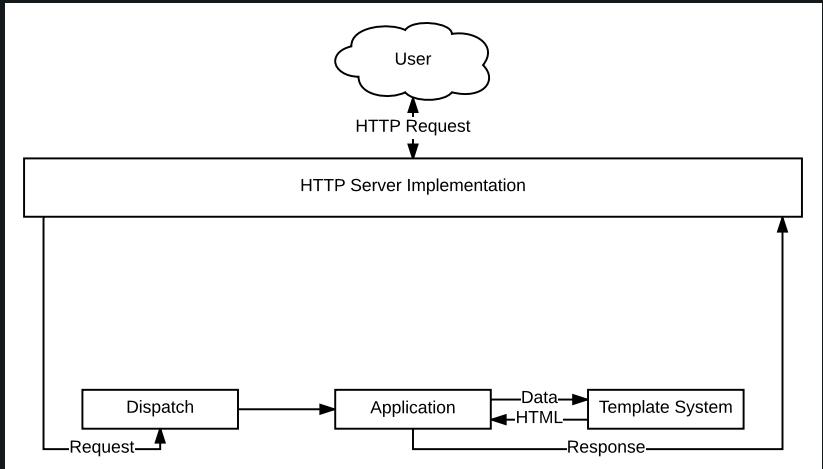
Interface Advantages

- No overhead as everything is direct calls
- Supports all kinds of definitions
- Allows framework to grow or shrink as needed
- Implementation can be exchanged

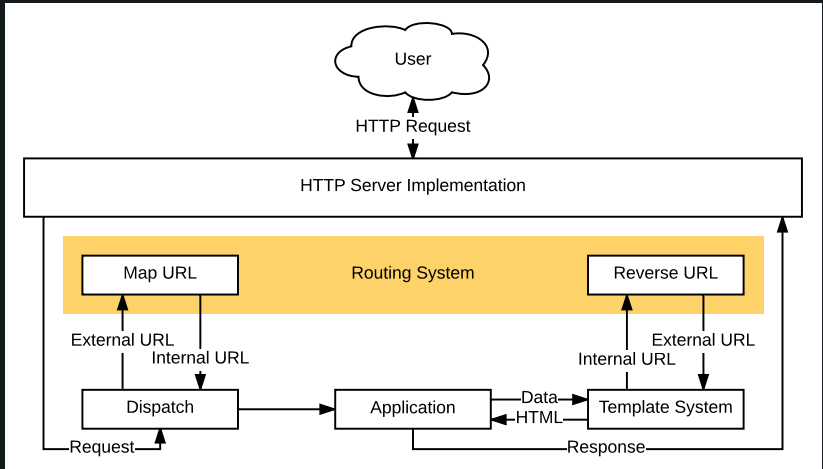
Routing

- Find proper content to deliver on an address
- Share the address space between applications
- Make it easy to visualise and understand

Routing Life-Cycle



Routing Life-Cycle



Routing Definition

Routing Example

Routing Advantages

Environments

Environment Access

Environment Advantages

Putting it All Together

Existing Applications

Resources

Acknowledgements