FATWIRE INTERACT 2010

GLOBAL USER CONFERENCE

FatWire® SOFTWARE

www.interact.fatwire.com

FATWIRE INTERACT 2010
GLOBAL USER CONFERENCE

Hit the Ground Running

Designing and Building Your Site with the GST Site Foundation

Tony Field
Global Solutions Architect

Agenda

- What is the GST Site Foundation?
- Why is it useful?
- The Budget Chief's Perspective
- The Architect's Perspective
- The Developer's Perspective
- Demo
- Questions

What is the GST Site Foundation?

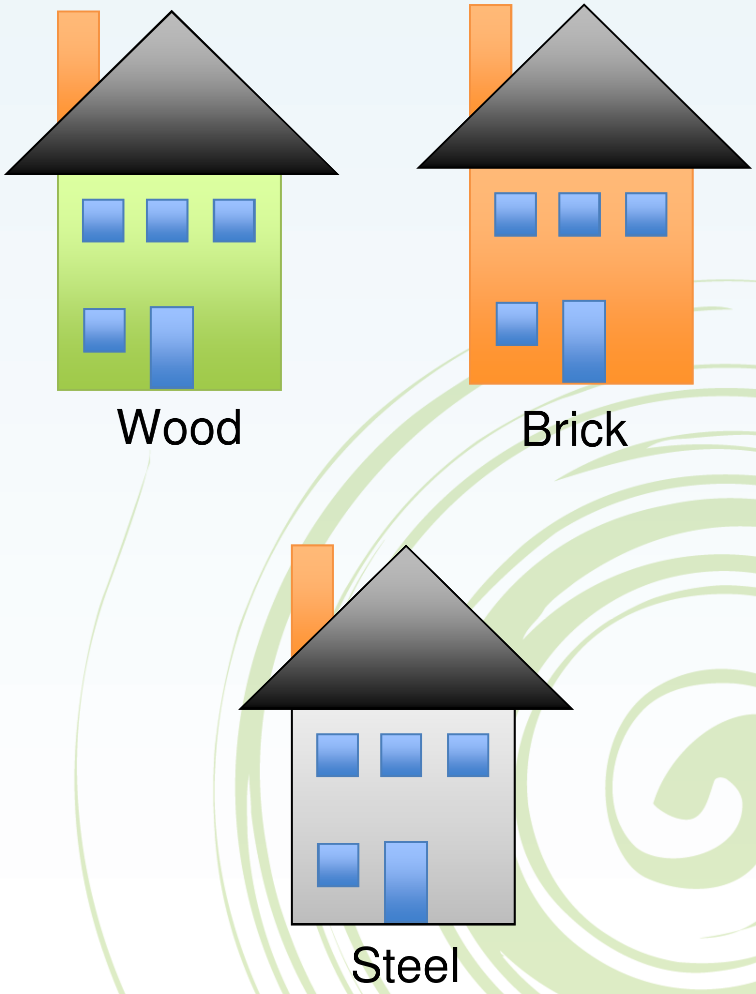
The GSF is a lightweight architecture specification for common patterns that arise in Content Server sites

Why is the GST Site Foundation Useful?

- **Most sites share common characteristics:**
  - Structured, mostly hierarchical navigation
  - Custom URLs
  - Multiple page wire-frames
  - Custom business logic
  - SEO optimizations
  - etc.

Why is the GST Site Foundation Useful?

- **Yet most sites use slightly varying architectures:**
  - Navigation
  - URL specification and calculation
  - Controllers, dispatchers & wrappers
  - Selecting the display style for an asset
  - Common attribute names
- **Simple tasks get in the way of domain-specific problems**



Why is the GST Site Foundation Useful?

- **The GSF takes the drudgery out of site design**
  - Defines architectural standards for common tasks
  - Standardizes processes that affect business users
  - Simplifies development & enables code sharing

- **In short**
  - It saves money
  - It enables an improved the end user experience
  - It allows for saved time to be re-invested

Introducing the GST Site Foundation

The Budget Chief's Perspective

The Budget Chief's Perspective

- **Efficient Development**
  - Saves time at each stage of the project
- **Reduced risk**
  - Tested design & high quality code
- **Flexible**
  - Can be enhanced to support project-specific initiatives
- **Improved return on investment for website**
  - Less time spent on plumbing means more time available for higher-ROI initiatives

Hard Numbers

- **Discovery Savings**
  - Customers gravitate to out-of-the-box features rather than wide open options (that comes later)
    - Estimated savings: 1 day
- **Design Savings**
  - URL generation: 1 day
  - Controller: 1.5 days
  - Navigation: 1 day
  - WRA conception & design: 2 days
  - Tagging: 1 day
  - Linking: 1 day
- **Development Savings**
  - URL: 5 days
  - Controller: 4 days
  - Navigation: 2 days
  - WRA DAOs: 2 days
  - Tagging: 3.5 days
  - Linking: 2 days
- **QA Savings**
  - Your team is not the first to test this framework: 2 days
- **Deployment Savings**
  - Virtual Webroot and mod_rewrite configuration is simplified: 2 days

The Bottom Line

Project Phase	Potential Days Saved
Discovery	1.0
Design	7.5
Development	18.5
QA	2.0
Deployment	2.0
Total	**31.0**

Using the GSF means that over a month worth of predicted labor costs can be re-invested into differentiating features and new initiatives

Introducing the GST Site Foundation

The Architect's Perspective

FatWire SOFTWARE

The Architect's Perspective

- **GSF is a pleasure to use!**
- **For consultants**
  - No more repetition
  - Fewer difficult answers
- **For customers**
  - No more ambiguity
  - More time to solve your own business' problems
- **Shorter design documents**
  - Simply refer to the GSF spec doc

FatWire SOFTWARE

The Architect’s Toolkit

- **GST Site Foundation contains several types of elements for architects**
  - Design patterns
  - Naming conventions
  - Utilities
  - Referenceable design specification

- **Simplify site design**
- **Organize complex operations**
- **Rapid site design**

GSF Design Patterns

- Web Referenceable Assets
- Aliases & Links
- Site Plan-Based Navigation
- Vanity URLs
- Controller
- Web Server Configuration

Web Referenceable Assets (WRAs)

- **You have been using these for years: these assets can be specified directly on the url (typically using c/cid)**

- **WRAs:**
  - Have ***only one*** URL: a vanity URL
  - Can be placed into nav bars
  - Have a template specified, depicting their display style
  - Have standard attributes
  - Can be either normal pages or landing pages

- **Standard Attributes**
  - metatitle
  - metadescription
  - metakeyword
  - h1title
  - linktext
  - template
  - path (i.e. URL)

- **SEO is built into the design**
  - Single URL
  - Standard Attributes

- **Can be either Basic or Flex**

Aliases & Links

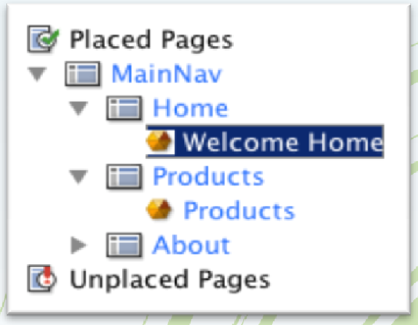
- **Every site requires the ability to create a link asset of some sort**
- **These have been created using names such as**
  - Link
  - Shortcut
  - ExternalLink
  - Reference
- **The new GSTAlias asset replaces all of these**
- **The common functionality has been abstracted into a base definition:**
  - Target asset (association)
  - Target URL (field)
  - Other WRA fields
- **Add your own fields**
  - The GSTAlias definition is only a partial spec. Add your own attributes as required.

SitePlan Tree-Based Navigation

- **Page assets and the SitePlan are typically overloaded**
  - Define navigational structure
  - Define URL hierarchy
  - Specify page composition
- **This is too complex and not flexible enough**
- **In the GSF, Pages and the SitePlan are used ONLY for navigation**
- **Page assets are not specified on the URL anymore**

SitePlan Tree-Based Navigation

- **Two new subtypes are introduced**
  - GSTNavName
  - GSTNavLink
- **GSTNavName**
  - Root nodes of trees represent navigation data structures (e.g. main nav, bottom nav)
- **GSTNavLink**
  - Child nodes represent navigation entries
  - WRAs are attached to NavLinks to create a linkable nav bar



Placed Pages
MainNav
Home
Welcome Home
Products
Products
About
Unplaced Pages

Vanity URLs

- Most customers develop their own URL Assembler
- Most customers want full control over URL design
- Programmatically deriving URLs can lead to accidental broken customer bookmarks when assets are moved

Solution:

Each web-referenceable asset

***defines** its own URL*

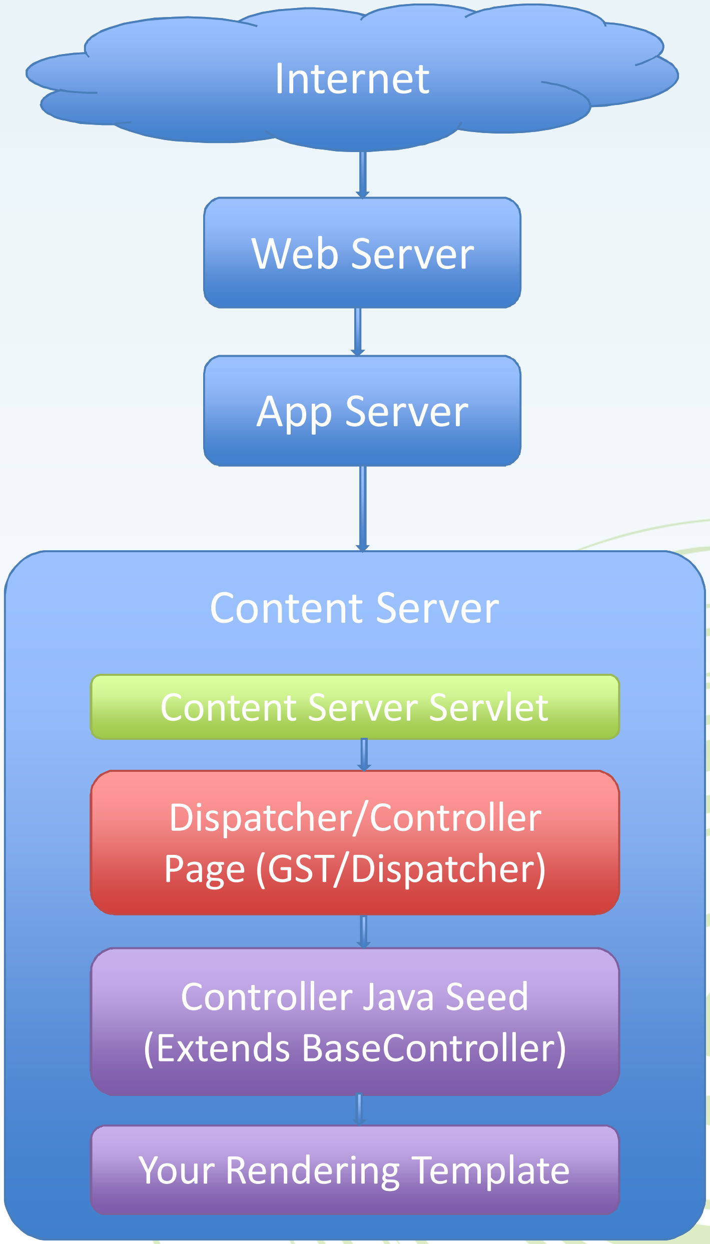
Vanity URLs

- **How it works**
  - FULLY-qualified URL is manually entered into the path attribute of a WRA
  - URL assembler replaces entire URL with the URL specified
  - URLs are registered into a database table for quick lookups
  - URL prefixes are registered in Virtual Webroot assets
  - Virtual Webroot assets are exported to webserver configuration to drive URL rewriting
  - Controller decodes URL & looks up c/cid in URL registry

- **Enhancements**
  - Customizing the path field UI form can be very beneficial
    - Enforce business rules
    - Suggest URLs based on textmining or rules
    - Prevent manual input if you must
  - Now instead of building a custom assembler, you just optimize the UI

Controllers

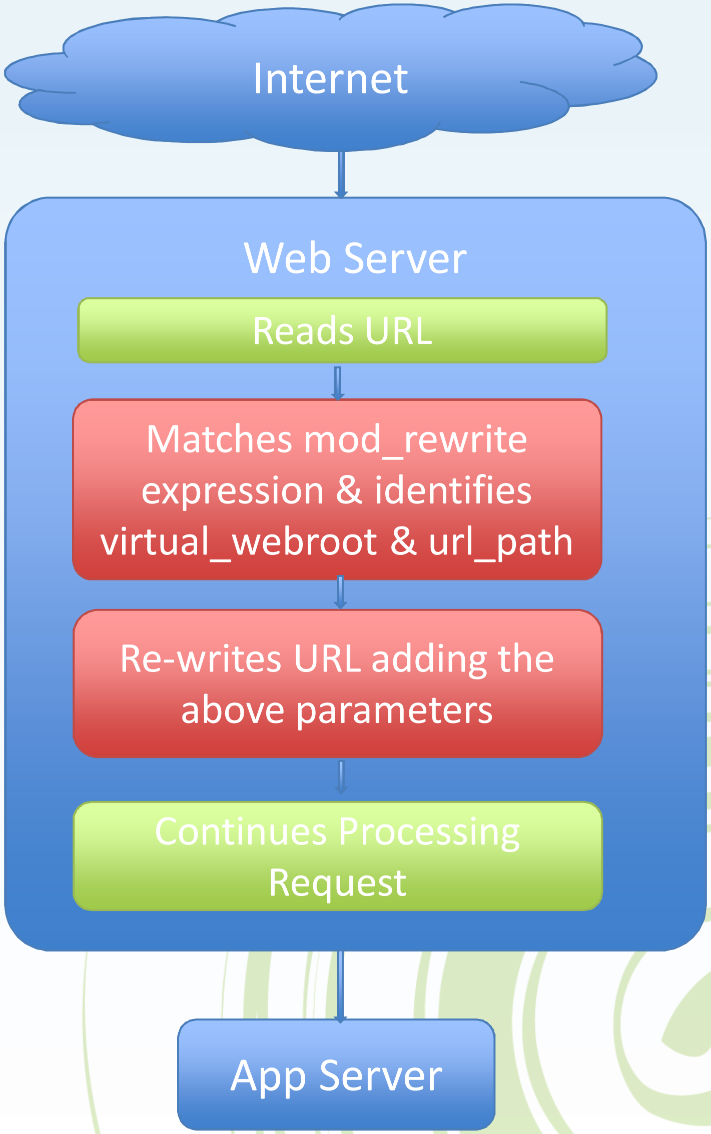
- **Most Content Server sites require a controller**
  - Typically we call this the *Wrapper* or the *Dispatcher*
  - This business logic is often written in JSPs – not a good fit
- **A controller framework is defined and specified plainly**
  - Pure java
  - Can redirect and send errors
  - Easily extensible
  - High-performance


Internet
Web Server
App Server
Content Server
Content Server Servlet
Dispatcher/Controller Page (GST/Dispatcher)
Controller Java Seed (Extends BaseController)
Your Rendering Template

Web Server Configuration

- **Web servers forward requests to the application server**
  - Which requests?
  - Any rewriting requirements?
  - Any blocking requirements?

- **GSF requires URL-rewriting to enable full vanity URL support**
  - Rules around how URLs are forwarded and rewritten are defined
  - Asset infrastructure is in place to support it
    - *"Virtual Webroot"*


Internet
Web Server
Reads URL
Matches mod_rewrite expression & identifies virtual_webroot & url_path
Re-writes URL adding the above parameters
Continues Processing Request
App Server

Introducing the GST Site Foundation

The Developer's Perspective

FatWire SOFTWARE

The Developer's Perspective

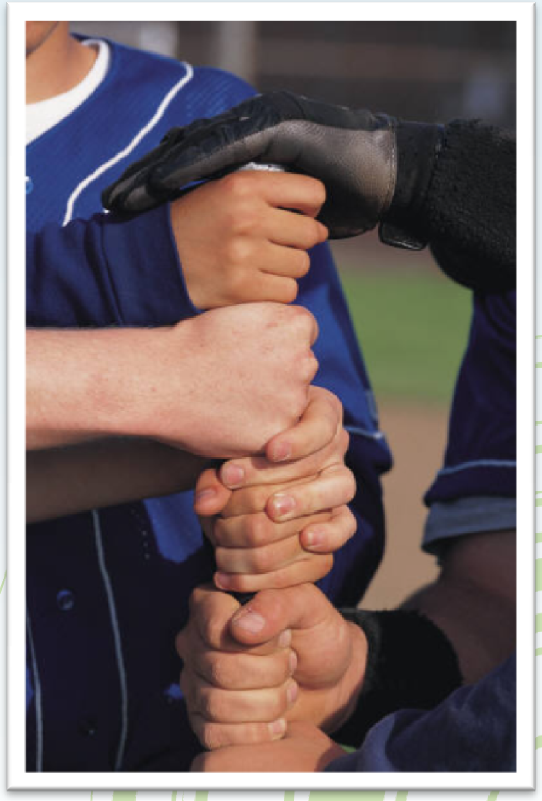
- GSF makes work more fun!
- Hit the ground running! After installation, start with your definition immediately
- No more hours spent on basic site wiring
- Lots of robust libraries handle core functions for you
- Simple, elegant architecture is easier and cheaper to maintain

The Developer's Toolkit

- **The GST Site Foundation contains a host of goodies for developers**
  - Simpler, more elegant designs from the architect
  - Code
  - Tutorials (coming soon)
  - Training (existing classes are updated to discuss the GSF)
  - Peer support
- **As a new standard and best practice, the user base will grow and the toolkits will be enhanced**
- **Code, you say?**

GST Site Foundation Code

- **Over 2.5 MB of code & supporting documentation**
- **Licensed under Apache 2 license**
  - (i.e. business-friendly Open Source)
- **Hosted in a public subversion repository**
- **Version 1.0 of the library was released this morning**



GST Site Foundation Code

- **Core Features**
  - WRA DAO
  - Alias DAO
  - Abstract & base controllers
  - Full vanity URL support
  - Built-in Alias support
  - Full tagging support
  - Integrated cache management

- **Scripting Features**
  - Navigation bar support
  - Link generation support

- **General Features**
  - Full set of convenience tools for working with common CS functions, Asset API, ILists, Exceptions & more

- **Available Today**

GST Site Foundation Website

http://gsf.fatwire.com/

Demo & Code Walk-through

A fully-functioning microsite
Built in one hour

Questions?

FATWIRE INTERACT 2010
GLOBAL USER CONFERENCE

Thank You

Tony Field
Global Solutions Architect
tony.field@fatwire.com