

SpinTag Komponenten

SpinTag Technologies entwickelt Frontend Produkte für komplexe, objektorientierte Backendsysteme. Alle SpinTag Komponenten folgen dem Ansatz des „Thin Clients“, d.h. sie benutzen ausschließlich Funktionalitäten eines Standard Webbrowsers (z.B. MSIE 5.01). Durch die Beschränkung auf HTML 4, DOM Level 1 und JavaScript (auch DHTML genannt) benötigen sie keinerlei zusätzliche Module wie Flash, Plug-Ins, JVM, oder ähnliches.

Da typischerweise serverseitig andere Implementationsstrategien und Sprachen als ECMA Script zum Einsatz kommen, liegt technisch die Fokussierung nahe. SpinTag ist Experte im Frontend Bereich und bietet hier interaktive Produkte, die sowohl preislich als auch vom Life-Cycle her gesehen eine attraktive Alternative zu Eigenentwicklung oder reiner Projektentwicklung ist.

Einige Editor Komponenten aus dem SpinTag Portfolio:

SpinTag WebWord WYSIWYG Editor für einzelne Objekte mit größeren Textblöcken, ggf. auch mit einfachen HTML Formatierungen.

Einsatzgebiete Redaktionelle Artikelerstellung, Dokumentenmanagement, u.v.a.

Demo <http://webservice.spintag.de>

Whitepaper <http://www.spintag.de/sthome/docs/wpWebWord.pdf>

SpinTag Table Editor Tabellenorientierter Editor für eine Anzahl von Objekten mit gleicher Struktur und mittlerer Verschachtelungstiefe.

Einsatzgebiete Redaktionelle Tabellenerstellung, Konfigurationsmanagement, Listenverwaltung, u.v.a.

Demo <http://www.spintag.de/sthome/tableEditor/>

Whitepaper <http://www.spintag.de/sthome/docs/wpTableEditor.pdf>

SpinTag Value Editor Editor für einzelne Objekte mit beliebig komplexer Struktur.

Einsatzgebiete Editoren für Objekteigenschaften, Konfigurationsmanagement, u.v.a.

Demo <http://www.spintag.de/sthome/valueEditor/>

Komponenten Integration

Um SpinTag Komponenten in Ihrem System nutzen zu können, müssen die Produkte integriert werden. Kernaufgabe der Integration ist es, eine gemeinsame Sprache zwischen Front- und Backend (GUI und Applikation) zu finden; eine Syntax, welche die Kommunikation über die zu verwaltenden Objekte gewährleistet.

Hierbei handelt es sich in der Regel um

- eine Aufrufsyntax von HTTP Services wie JSP, PHP, ASP, Perl
- oder eine Sprachsyntax wie XML mit Schemas, geliefert über ein beliebiges Protokoll
- oder eine Client API Syntax wie JAVA, C++.

Von der Wahl der Anbindung unberührt bleiben jedoch die Konzepte der Integration. Um unnötige Synchronisationsprobleme zu vermeiden, unterliegen die Sicherheitsüberprüfungen, die semantische Gültigkeit der Werte und ihr spezifischer Workflow nach wie vor alleine der Applikation. SpinTag Komponenten fokussieren sich rein auf die interaktiven Aspekte des Managements von Applikationsobjekten. Objekttypen werden vom System erfragt, um neue Objekte typischer zu erzeugen; existierende Objekte werden geladen und nach ihrer Modifikation zurück in den Workflow gegeben.

Die klare Festlegung dieser Serviceschnittstellen sind die Basis der Integration und werden als solches am Anfang des Integrationsprojektes von SpinTag mit Ihrer Mithilfe erarbeitet, falls sie nicht bereits dokumentiert sind. Im wesentlichen können vier Gruppen von Services identifiziert werden, wovon drei für die Integration elementar sind:

Metadata Services

Metadaten beinhalten Informationen zu unterstützten Datentypen des Backendsystems. Diese werden bei der Erzeugung von neuen Objekten benutzt und bestimmen die Regeln zur Modifikation. Normalerweise enthalten sie die Datenstruktur eines Objekttyps und seine Gültigkeitsbedingungen.

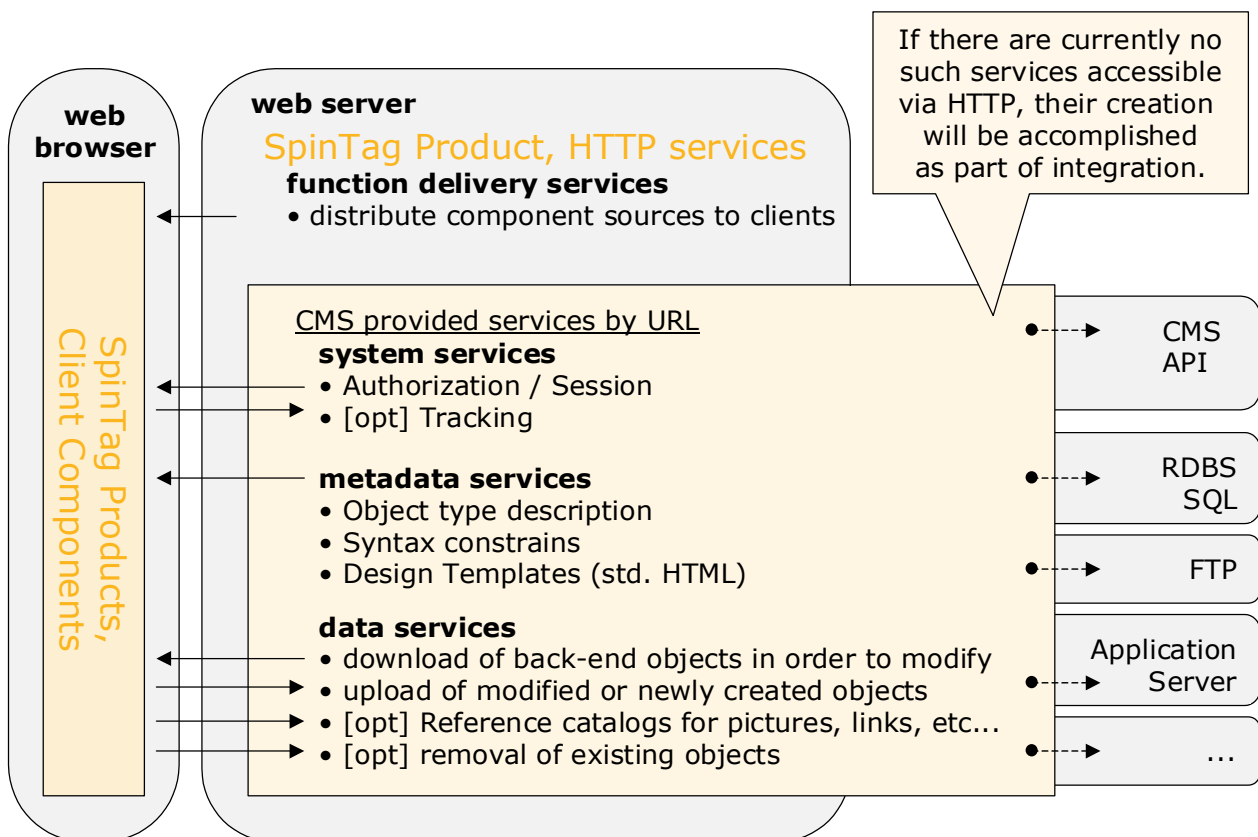
Data Services

Nutzdaten sind die eigentlichen Werte des Backendsystems. Auch sie werden über HTTP übertragen. Neben dem „download“ von CMS Objekten und ihrer Repräsentation ist beim „upload“ insbesondere die Einbettung in den spezifischen Workflow dieser Objekte wichtig. Des weiteren können Daten auch Objektkataloge des Backendsystems umfassen (z.B von Bildern, Sound, Objekten, etc.), die dem Benutzer zum Referenzieren angeboten werden. Je nach Umfang können Kataloge initial geladen werden oder „on demand“ nach Bedarf nachgeladen werden.

System Services

Da die SpinTag Frontend-Komponenten keine eigene Businesslogik dem Backendsystem überstülpen wollen, werden Systemdienste zumindest zur Authentifizierung des Benutzers benutzt.

Das folgende Bild fasst im wesentlichen die Integrationsaufgabe technisch zusammen.



Die vierte Gruppe von Services bedarf technisch keiner weiteren Zusammenarbeit:

Function distribution Services

Diese Standarddienste sind im Produktumfang enthalten und müssen nicht angepasst werden. Sie dienen im wesentlichen der Belieferung des Browsers mit Funktionalität. Zur Zeit nutzen wir PHP Skripte um die JavaScript Komponenten zu distributieren. Sollte jedoch die Homogenisierung der Sourcen gewünscht sein, so können die Komponenten kurzfristig auch über alle anderen webbasierten Services geliefert werden.

Komponenten Benutzung

Komponenten Framework

Die Einbindung der einzelnen Komponenten in eine Webumgebung findet über das Framework statt. Die Framework-Komponente bietet folgende infrastrukturellen Dienste für alle anderen Komponenten:

- Nutzer-Identifizierung
- Komponenten-Management wie das Laden von Komponenten-Code, Aktivierung / Deaktivierung von spezifischen Instanzen oder Manipulierung der Komponenten-Eigenschaften
- Komponenten-Überwachung: Statistik, Monitoring, Eskalation
- Kommunikation zwischen den Komponententeilen
- Datenkomprimierung / Dekomprimierung

Synopsis:

```
<head>  
    <script language="JavaScript" src="framework.js"></script>  
</head>
```

Dieser Link im <HEAD>...</HEAD> Bereich einer Webseite ist ausreichend, um SpinTag Webkomponenten einzubinden.

Komponenten Instanzierung

SpinTag Webkomponenten werden nach folgendem Schema instanziiert:

a) Statisch (in einem HTML-Dokument)

In einem HTML-Container Element (z.B. SPAN, DIV, TD) wird der Typ der Komponente, der Serverkontext und eine Parameterliste eingetragen. Das HTML-Element muss die Klasse SpinTag haben.

Synopsis:

```
<body>
  <div class="SpinTag" [id=clientContext]
    componentType [serverContext] [property=value[property=value]]>
  </div>
</body>
```

Der Standardparameter clientContext wird in der Regel autogeneriert. Soll die Komponente jedoch adressierbar bleiben, kann die ID explizit vergeben werden. Die Bedeutung von serverContext ist komponentenspezifisch. Zumeist werden serverseitige Ressourcen spezifiziert.

b) Dynamisch (aus JavaScript heraus)

Innerhalb von JavaScript können Komponenten auch programmatisch mit der Framework Funktion create() instanziiert werden.

Synopsis:

```
<head>
<script language="JavaScript">
  STFramework.create(componentType,clientContext,serverContext,
  "[property:value[;property:value]]")
</script>
```

```
</head>  
<body>  
    <div id="clientContext"></div>  
</body>
```

Die Position der Instanz im Dokument wird hierbei durch die als clientContext spezifizierte Element ID bestimmt. Diese muss eine gültige, existierende HTML Container ID innerhalb des Dokuments sein.

Kundenspezifische Anpassung

Komponenten können durch Eigenschaftslisten gesteuert werden. Der Name einer Eigenschaft und deren Wertzuweisung wird durch ein Gleichheitszeichen getrennt. Es können auch mehrere Eigenschaften innerhalb einer Komponente spezifiziert werden (abhängig von der Komponente).

Projektschema

Evaluation

Möchten Ihre Kunden den Schulungsbedarf senken?

Möchten Ihre Kunden ggf. ein "customer selfcare model" einführen?

Wünschen Sie weitere "unique selling points" in Ihrem Produkt?

web-basierend WYSIWYG editieren; bald ein notwendiges "check-off item" für Ihr Produkt?

Passt ein (alternativer) "Thin Client" in Ihre "product road map"?

...

Beauftragung

Wir freuen uns auf die Zusammenarbeit mit Ihnen!

Workshop

Eine erfolgreiche Integration bezieht immer alle involvierten Parteien mit ein. Wie in jedem soliden Projekt ist ein gutes Teamwork (auch über Firmengrenzen hinweg) elementar.

Hier auszugsweise eine Liste der Fragen, die der Klärung bedürfen:

1. Integration Level
 - Service, Sprache, API
2. Security Policy
 - z.B. session data mit access key
 - security controlled entities (Values, Kontext, Features/Funktionen,...)
3. Welche Metadaten werden benötigt und wie liegen sie vor ?
 - design (z.B. Entity Relationship Diagramme), Klassendefinitionen, SQL Schemas, XML Schemas oder DTD, ...
4. Wie werden Metadaten zur Laufzeit benutzt ?
 - URL, DB, SOAP/XML-RPC, ...
5. Wie wird der Kontext eines Objekts spezifiziert ?
 - Design Templates, Metadaten (structure, constrains), Values
6. Wie wird auf die Objekte selbst zugegriffen ?
 - create, get, set, delete
7. Embedded Objects
 - Anzahl, Art, Referenzierungsstrategie
8. Festlegung der Test Cases

Spezifikation

Da unserer Erfahrung nach Unklarheiten oder Fehler in der Spezifikation die gravierendsten sind, legt SpinTag hier besonderen Wert auf die Abstimmung mit Ihnen als Auftraggeber. Mit der Abnahme der Spezifikation ist der erste Meilenstein erreicht.

Implementation

Dauer und Umfang der Implementation hängt im wesentlichen von den im Workshop erarbeiteten und in der Spezifikation festgehaltenen Aspekte der Integration ab. Mit Abschluss der SpinTag internen Testszenarios ist der zweite Meilenstein erreicht.

Test und Abschluss

Mit der Abnahme des Projekts gemäss den in den Spezifikationen festgelegten Punkten ist der dritte und letzte Meilenstein erreicht.