

Alfresco



NikeWebConsulting



Non solo un documentale

«1» Nuove operazioni

- E' possibile ampliare l'insieme delle "operazioni" eseguibili da Alfresco
- Implementazione di:
 - Nuove azioni
 - Nuovi comportamenti
- Esempio: sistema di rating
 - Incremento contatore visite spazio padre
 - Incremento contatore visite contenuto
 - Contatore visite spazio/contatore visite contenuto

«2» Definizioni dei dati

- I dati memorizzati nel Repository sono descritti dal “Content Model”
- Insieme di file XML che definiscono:
 - Tipi
 - Vincoli
 - Associazioni
 - Aspetti

«3» Estendere il modello

- E' possibile definire nuovi tipi, vincoli, associazioni e aspetti, in base alle necessità
 - A partire da zero
 - Ereditando le proprietà presenti nel modello proposto da Alfresco
- Vantaggio: tutte le funzioni di Alfresco possono essere applicate anche ai nuovi dati

«3»

Estendere il modello: un esempio

- Contenuto di tipo “Tecnico”
 - Piattaforma (proprietà)
 - Linguaggio di programmazione (proprietà vincolata)
 - Redattore (associazione con un utente)
- Contenuto di tipo “Amministrativo”
 - Numero documento (proprietà)
 - Data scadenza (proprietà)

«3» Estendere il modello

- Non necessariamente estensione orientata alla gestione documentale
- Possibilità di definire nuovi modelli per descrivere e rappresentare realtà di interesse
- Utilizzo delle potenzialità di Alfresco su tali modelli
- Un esempio: Trouble-Ticketing

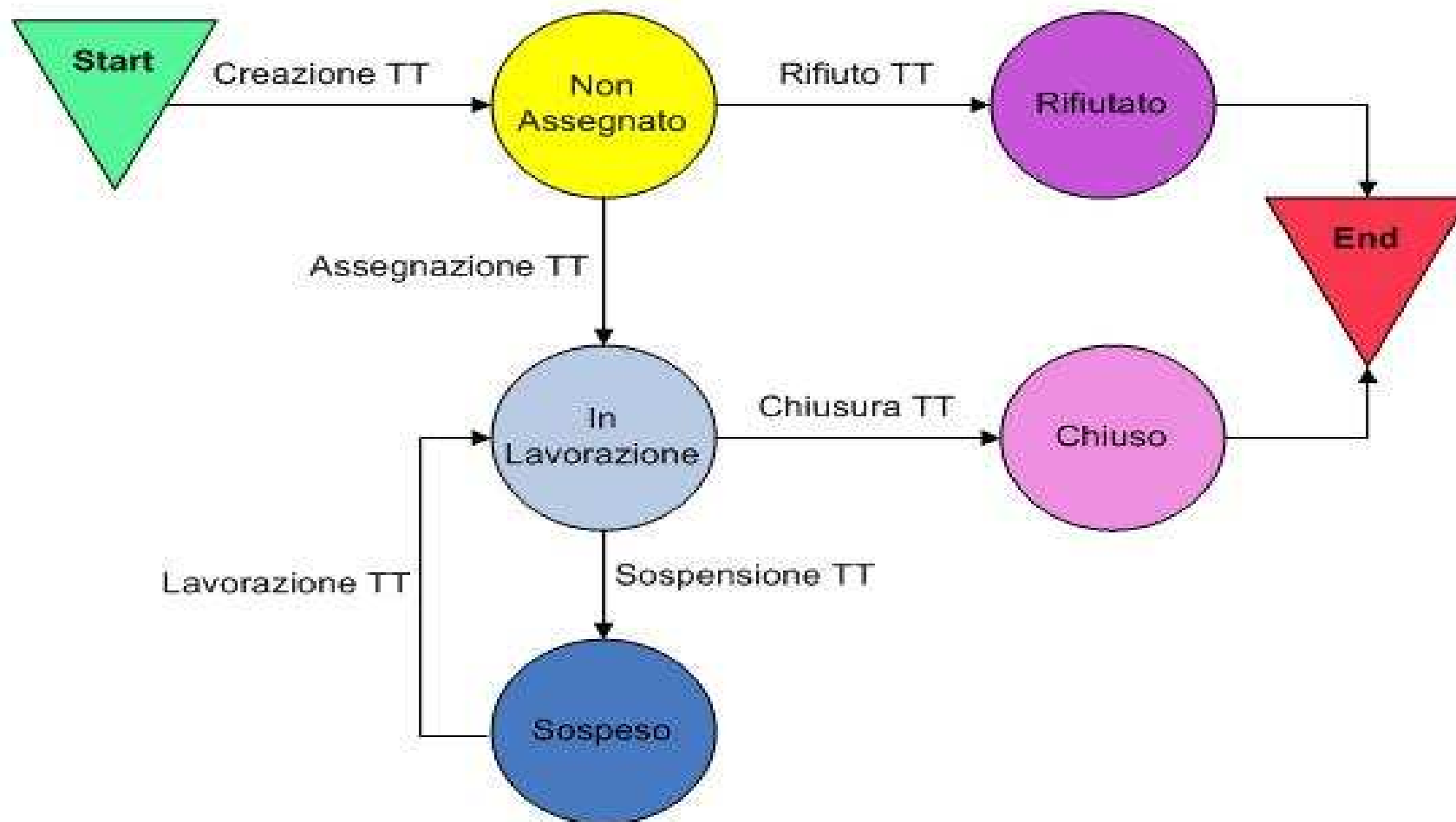
«4» Un esempio avanzato: Trouble-Ticketing

- Il Ticket
 - Applicativo di interesse
 - Utente richiedente
 - Utente assegnatario
 - Utente operatore
 - Data di apertura
 - Data di chiusura o di rifiuto
 - Descrizione
 - Note
 - Priorità

«4» Un esempio avanzato: Trouble-Ticketing

- Definizione di una nuova tipologia di dato, ovvero il Ticket
 - Non è un content “classico”
 - Non è un documento
 - Necessita di proprietà ad-hoc
- Applicazione di un “Workflow Avanzato” al Ticket

«4» Un esempio avanzato: Trouble-Ticketing



«5» Workflow avanzati

- Oltre al “Workflow Semplice”, offerto da Alfresco, possono essere realizzati “Workflow Avanzati” a seconda di ogni esigenza
- Stilizzabili come grafi orientati, dove:
 - I nodi sono gli stati
 - Gli archi sono le transizioni
- Le operazioni da compiere possono essere scaturite dal transito in uno stato o dal passaggio per una transizione

«6» Oltre al WebClient

- Come per tutte le altre funzionalità di Alfresco, anche l'applicazione dei Workflow Avanzati può avvenire attraverso il WebClient (previa opportuna configurazione)
- Tuttavia potrebbe verificarsi l'esigenza di interagire con il Repository, attraverso un "Presentation Layer" avente un proprio stile grafico
 - Estremamente complesso customizzare completamente il WebClient
 - Meccanismi di navigazione indipendenti dal WebClient

«7» Uno scenario: Il Portale dello Studente

- Deploying di Alfresco come Portlet di Liferay Portal Server
- Realizzazione di pagine funzionali e indipendenti dal WebClient, per la visualizzazione dei contenuti, dotate di una propria grafica
- Efficienza:
 - Comunicazione diretta tra il Presentation Layer (gestito da Liferay) e il Repository (gestito da Alfresco)

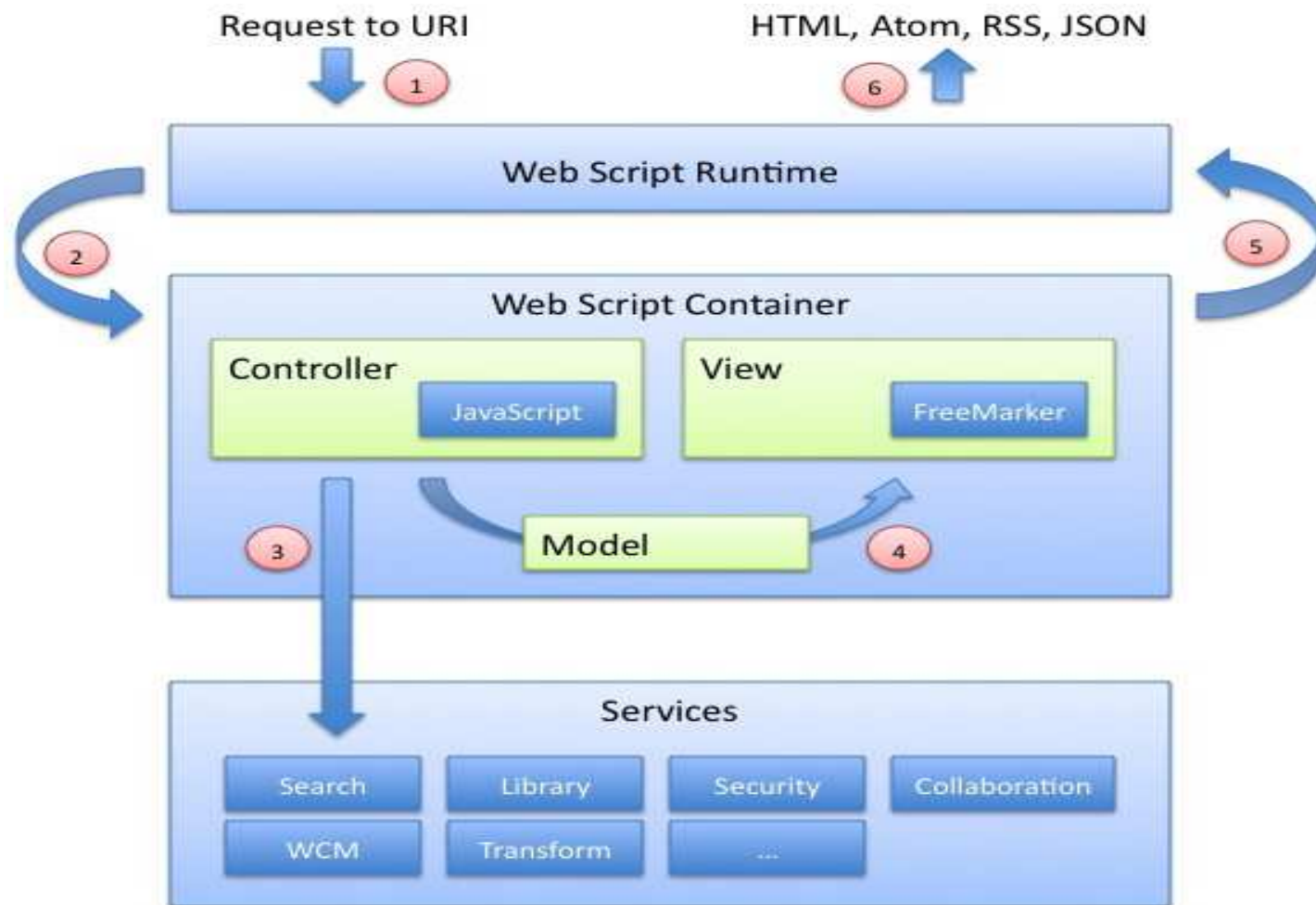
«8» Un altro scenario

- Portal Server e Alfresco su piattaforme diverse
 - Impossibile il deploying di Alfresco come Portlet
- Chi offre il “Presentation Layer” deve comunicare in maniera remota con il Repository
- Soluzione: WebScript

«9» WebScript

- Servizi indipendenti dal WebClient
- Vivono nel WebScript-Container
 - SURF
 - Repository navigabile attraverso chiamate remote
 - Alfresco Webscript-Container
 - Repository navigabile attraverso le Foundation API
- Realizzati spesso attraverso FreeMarker
 - Linguaggio dedicato all'implementazione del "presentation layer"

«9» WebScript



«9» WebScript

- Realizzazione API REST
- Tipicamente
 - Chiamata HTTP al “Presentation” (da browser)
 - Il “Presentation” chiama a sua volta il WebScript d’interesse, sempre via HTTP
 - Il WebScript esegue il suo codice e risponde
 - Risposta semplice (codici HTTP)
 - Interpretazione demandata al “Presentation”
 - Risposta complessa (HTML, XML, ecc...)

«9» WebScript

- Ogni WebScript è costituito da 3 file:
 - Configurazione
 - Ad esempio:
 - Livello di permessi per eseguirlo
 - Path cui deve rispondere
 - Funzione
 - Il codice (Java o Javascript Server-Side), ovvero la logica di business associata
 - Presentazione
 - L'eventuale HTML prodotto
- Il nome del WebScript determina il metodo HTTP cui deve essere associato
 - Ad esempio: miowebscript.get.XXX, miowebscript.post.XXX, ecc...

«9» WebScript

- Aderenza al Design-Pattern MVC (Model-View-Controller). Quindi:
 - Stile grafico completamente personalizzabile
 - Possibilità di adattamento ai principali standard di codice e di accessibilità