

Landesintranet Niedersachsen auf Linux/PHP-Basis

7. Juni 2004

Note légale

Dieser Beitrag ist lizenziert unter der UVM Lizenz für die freie Nutzung unveränderter Inhalte.

Zusammenfassung

Es wird von Erfahrungen bei der Einführung und Nutzung zweier Lösungen für das Behördenintranet des Landes Niedersachsen berichtet, die auf Basis von Linux und PHP realisiert wurden: Einerseits ein zentrales Informationsportal für die rund 50.000 Landesbediensteten in 500 Dienststellen. Andererseits eine große Zahl eigenständiger Intranets für einzelne Ministerien und Dienststellen. Grundlage der technischen Realisation sind ein Portalserver und ein mandantenfähiges Enterprise Content Management System - beide in PHP programmiert. Die Mischung aus kommerzieller und nicht-kommerzieller Open-Source Software entspricht dabei der langfristigen IT Strategie des Landes Niedersachsen. Fortgeschrittene Nutzer können in ihrem Browser verschiedene Portlets aus den Bereichen Wissensmanagement, assoziative Suche, Groupware und Redaktionssystem zu persönlichen Desktops zusammenstellen und damit arbeiten.

1 Ausgangslage: Das bisherige Intranet im Land Niedersachsen

Die gewachsene Informationsstruktur des Landes Niedersachsen ist das Produkt der dezentralen Organisation der Landesbehörden. In den vorhandenen Intranets herrscht absolute Autonomie und Heterogenität bezüglich der technologischen und inhaltlichen Infrastruktur. Als Betriebssysteme sind Linux, Microsoft Windows und Solaris im Einsatz, als Datenbanken unter anderem MySQL, Oracle, Informix und MS-SQL. Die Intranets sind mit verschiedenen Programmier- und Skriptsprachen auf verschiedenen Webserverarchitekturen realisiert. Einige Hausintranets wurden mit Hilfe eines Java-Baukastens für dynamische Webs inklusive MySQL Datenbank aufgesetzt. Design und Nutzerführung sind uneinheitlich, die Qualität der Auftritte variiert erheblich hinsichtlich inhaltlicher Aktualität und Güte. Unklar ist, welche Personen Zugriff auf welche Inhalte welcher Dienststellen haben dürfen. Es fehlt also u.a. die Gliederung des Angebotes in einen öffentlichen Bereich, der von allen Ministerien und Dienststellen aus zugreifbar ist und einen internen Bereich, der nur im Hausintranet gelesen werden darf. Die Größenordnung des Projektes liegt zukünftig bei ca. 500 Intranets von Ressorts, Bezirksregierungen, Behörden, Ministerien usw. mit insgesamt rund 50.000 Landesbediensteten. Zu Beginn lagen ca. 80.000 Dokumente nahezu unstrukturiert in 30 verschiedenen Intranets vor. Die bereits vorhandenen zentralen Dienste wie Vorschrifteninformationssystem, E-Mailverbund, Fachdatenbanken sind in die neue Lösung zu integrieren.

2 Anforderungen, Ziele und Schritte auf dem Weg zum zukünftigen Landesintranet

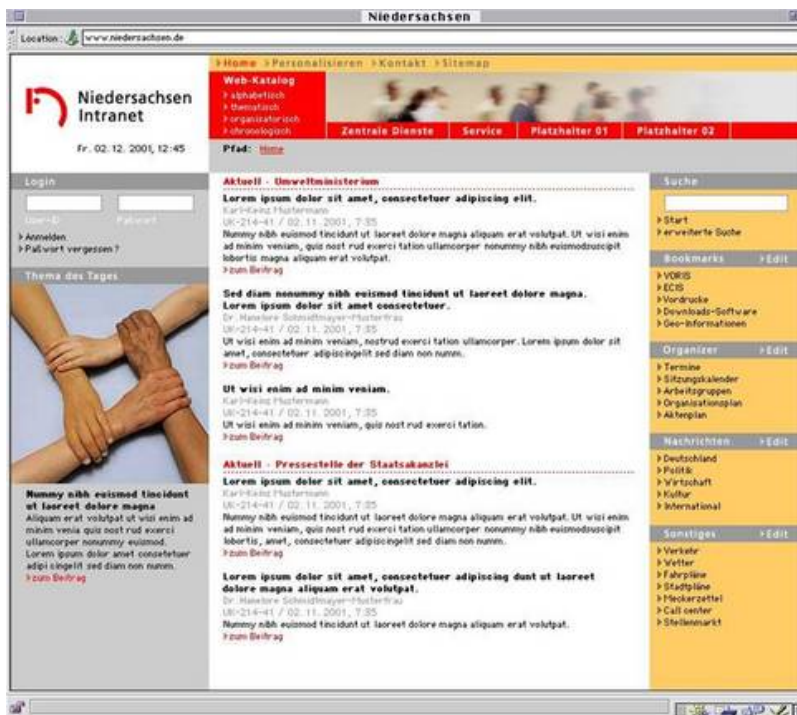
Der im Jahr 2001 begonnene Erneuerungsprozess unter Federführung des zentralen IT-Managements des Landes Niedersachsen umfasst die Analyse des Ist-Zustandes, Anforderungsdefinitionen, Produktauswahl und die Realisierung des zukünftigen Landesintranets. Wichtige Eckpfeiler sind die Einrichtung eines zentralen Dachportals für alle Landesbediensteten und die Standardisierung der 500 Intranets: Beides im Sinne eines umfassenden Kollaborations- und Wissensmanagementansatzes mit der langfristigen Vision eines personalisierten Desktops für jeden Mitarbeiter. Mit Basiswerkzeugen wie Enterprise Content Management System (ECMS), Workflow Engine, Portalserver und assoziativer Suchmaschine werden Dachportal und Intranets

nach Vorgabe eines einheitlichen Styleguides mit möglichst identischer Benutzerführung aufgebaut. Der Einsatz von kommerzieller und nicht-kommerzieller Open-Source Software entspricht dabei der langfristigen IT Strategie des Landes Niedersachsen. Im Fokus dieses Beitrages stehen die vollständig in PHP entwickelten Produkte der Firma flying dog software, die als Basistechnologien im Landesintranet zum Einsatz kommen.

3 Architektur des zentralen Dachportals

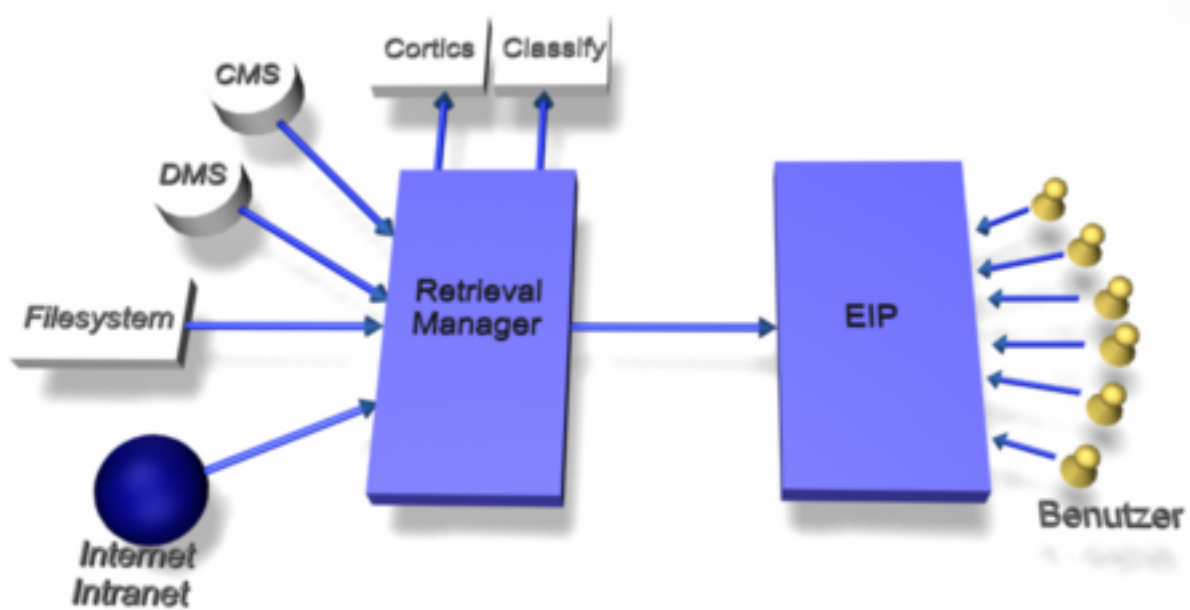
Zielgruppe für das zentrale Portal sind alle ca. 50.000 Landesbediensteten. Ein Web-Redaktionssystem als Teil des ECMS ermöglicht die schnelle Bereitstellung von neuer Information. Ist ein Nutzer am zentralen LDAP-Verbund angemeldet, kann er darüber hinaus ein persönliches Archiv nutzen und aktiv an Projektgruppen und Communities teilnehmen. In der Umsetzung entspricht jede Projektgruppe bzw. Community einem eigenen zentralen Intranetmandanten, wie sie weiter unten ausführlich besprochen werden.

Bild1 - Portal aus Nutzersicht:



Der technologisch interessanteste Teil des Dachportals beschäftigt sich mit dem Zugriff auf vorhandene Information aus den öffentlich zugänglichen Teilen der Hausintranets. Wie soll der Zugriff auf die fast hunderttausend unstrukturiert vorliegenden Dokumente erfolgen, solange nicht jedes Intranet auf dem einheitlichen ECMS basiert? Die Grundidee ist einfach: Die Originaldokumente bleiben dort wo sie sind, werden lediglich von einem Crawler erfasst und in einen zentralen Suchindex aufgenommen; ähnlich wie man das von Internetsuchmaschinen kennt. Die so erfassten Dokumente werden dabei zusätzlich vollautomatisch in den thematischen und organisatorischen Webkatalog des Landesintranets eingeordnet. Die Erfassung der Dokumente erfolgt dabei inkrementell und in regelmäßigen Zeitabständen neu. Die Landesbediensteten greifen anschließend im Portal über eine semantische Volltextsuche oder über die Webkataloge auf alle öffentlich zugänglichen Dokumente der Hausintranets zu. *Bild2* zeigt schematisch die Architektur der Lösung mit der Retrieval-Komponente im Mittelpunkt, die wir im Folgenden noch etwas genauer beleuchten wollen.

Bild1 - Architektur des zentralen Dachportals:



3.1 Der Dokumenten Retrieval Prozess: SOAP und die Idee eines flexiblen Queue-Plugins

Aufgabe des Retrieval Managers ist mithin die Bestandsaufnahme der dezentralen Unternehmensdaten. Welche Datenquellen auf welchen Rechnern zu erfassen sind, wird von den Administratoren im webbasierten Backend des Dachportals konfiguriert. Als flexible Kommunikations- und Steuerungszentrale für den Informations- und Dokumentenfluss zwischen den verschiedenen Softwarebestandteilen, gibt der Retrieval Manager diese Quellenkonfigurationen an die eingesetzten Web- bzw. Filesystem-Crawler sowie andere Konnektoren weiter und startet die Crawlprozesse. Die Dokumente werden in eine Warteschlange (Queue) zurückgeliefert. Ein Queue-Plugin übernimmt die weitere Verarbeitung: Die Dokumente aus der Warteschlange werden mit einem Filterserver zunächst nach HTML gewandelt, dann mittels einer Klassifizierungssoftware den angelegten Rubriken des Webkatalogs zugeordnet und schließlich in einen (assoziativen) Volltextindex inseriert. Alle rechenintensiven Anwendungen werden konsequent über Webservices angebunden. Die Kommunikation ist dabei in der Regel asynchron, d.h. der Retrieval Manager selbst und die einzelnen Webservices besitzen jeweils SOAP Client und Server. So sind je nach Anwendung mindestens 6 SOAP Server aktiv, wobei 90% der Kommunikationslast in der gewählten Standardinstallation (alle Komponenten bis auf das Dachportal EIP selbst sind auf demselben Rechner installiert) auf localhost anfallen, also rein interne Kommunikation ist. Was sind Vorteile dieser aufwändigen und auf den ersten Blick mit viel Kommunikationsoverhead belasteten Architektur? Folgende Gründe waren für die Entwicklung ausschlaggebend:

- Herstellerunabhängigkeit: Die einzelnen Software-Komponenten bzw. Fremdprodukte anderer Firmen (Crawler, Konnektoren, Filter, Suchmaschinen, Klassifizierungsenigines) sind jederzeit austauschbar
- Skalierbarkeit: Einzelne Bestandteile können leicht auf eigene Server ausgelagert werden
- Wartbarkeit: SOAP Schnittstellen sind transparent, leicht verständlich und mit entsprechenden Tools einfach zu testen

Der Hauptgrund war allerdings unser Wunsch, die eigentliche Steuerungszentrale als möglichst flexible PHP-Applikation zu schreiben, um schnell auf die unterschiedlichen Gegebenheiten in verschiedenen Organisationen reagieren zu können. Damit auch sehr große Datenmengen (mehrere hunderttausend Dokumente bzw. mehrere GB an Daten) verarbeitet werden können, ist der Ablauf dieser Verteilungsprozesse - getrennt vom eigentlichen Crawlen - in eine zentrale Queueverwaltung eingebettet. Durch die gewählte Architektur ist diese mitunter komplexe Verteilerlogik jetzt als (PHP-)Queue-Plugin im Retrieval Manager programmiert und kann, da PHP Code, leicht geändert werden. Interessant zu erwähnen ist, dass unsere Performancetests ergeben haben, dass die starke Kommunikation der Webservices und die in PHP implementierten Programmteile (Retrieval Manager mit Queueverwaltung, PHP-Queue-Plugin) keine nennenswerte CPU-Zeit in Anspruch

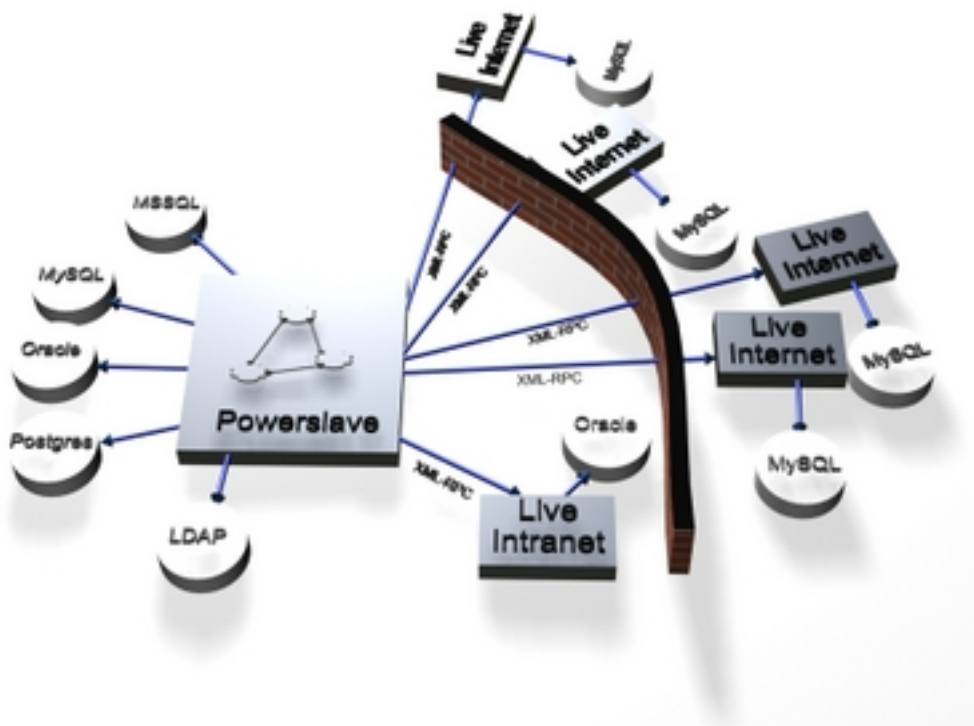
nehmen: 99% der CPU-Zeit wird in den einzelnen Komponenten wie Filtersoftware, Klassifikation oder Crawler verwendet. Auf PHP Seite wurde NuSOAP für die Implementation der Webservices eingesetzt.

4 Architektur der 500 Hausintranets: Wie zentral kann und darf Dezentralität sein?

Die zweite Säule der Neugestaltung des Behördenintranets im Land Niedersachsen besteht in der Standardisierung der ca. 500 Intranets von Ressorts, Bezirksregierungen, Behörden, Ministerien... Dabei besteht jedes Intranet aus mindestens einem Web-Redaktionssystem mit Mediendatenbank und Dokumentenverwaltung sowie einigen Workflow für das Antragswesen. In jedem Intranet gibt es öffentliche und geschützte Bereiche, die nur innerhalb der Behörde sichtbar sind. Alle Intranets haben dasselbe Design gemäß Styleguide und auf der obersten Ebene dieselbe Navigation. Unter dieser Ebene kann alles unterschiedlich sein, wobei manche Dienststellen erheblich mehr Funktionalität innerhalb ihres Intranets anstreben.

In der technischen Umsetzung wurde auf eine Mischstrategie aus verschiedenen Architekturen gesetzt: Die meisten Intranets werden von der zentralen IT als so genannte "Mandanten" gehostet, deren Nutzung den Dienststellen als Service angeboten wird. Andere Behörden arbeiten auf virtuell- oder echt autonomen Instanzen des ECMS, entweder ebenfalls auf den zentralen Servern oder auf hauseigenen dezentralen Servern. Es besteht auch die Möglichkeit, zunächst als Mandant auf dem zentralen System zu starten und sich später als dezentrales Intranet abzukoppeln. Die Entscheidung für die eine oder andere Variante hängt u.a. von der Vertraulichkeit der zu verwaltenden Inhalte, den Sicherheits- und Datenschutzrichtlinien der Behörde sowie den aufzuwendenden Lizenz- und Betriebskosten ab (für die zentrale Lösung gibt es z.B. bereits eine Backup-Lösung). Durch die flexible Architektur des ECMS (vergl. Bild3), in der Entwicklungserver, Liveserver und Datenbankserver getrennt sein dürfen, sind auch spezielle Architekturen möglich, in denen Ministerien eigene Datenbankserver mit ihren Daten (hinter der Firewall) betreiben, aber trotzdem als Redaktionssystem den zentralen Server nutzen.

Bild 3 - Beispiel einer Serverarchitektur:



4.1 Mandanten: Konzept und Grenzen des Ansatzes

Baut man mit dem eingesetzten ECMS "Powerslave" Webapplikationen wie z.B. ein Intranet mit Redaktionssystem auf, laufen im Hintergrund u.a. folgende Prozesse ab:

- Applikationen bestehen aus Powerslave-Modulen (ein einfaches Intranet z.B. aus Benutzern, Rechten, Navigation und Inhalten)
- Solche Module haben flexible Datenmodelle mit Entsprechungen in Datenbanktabellen
- Module können Teile der Filesysteme auf den Servern benutzen, die ihnen zugeordnet wurden
- Corporate Design bei Ein- und Ausgabe wird über Templates erreicht, die den Modulen zugeordnet sind
- Powerslave generiert auf mehreren Ebenen automatisch Programmcode, z.B. zur Abfrage der Datenbanken
- alle angezeigten Webseiten werden dynamisch zur Laufzeit mittels der Templates und dem automatisch generierten Programmcode aus den in der Datenbank gespeicherten Informationen erzeugt

Die Grundidee bei der Umsetzung der zentralen Mandanten ist, dass alle zentralen Mandanten dieselben Powerslave-Module benutzen, i.e. dieselben Datenbanktabellen, Filesysteme und Templates. Die Trennung der Mandanten erfolgt auf der untersten SQL-Ebene durch Erweiterung der Datenbanktabellen um ein weiteres Attribut "MandantenID" und Modifikation des automatisch erzeugten Programmcodes um WHERE-Bedingungen, die alle Datenbankzugriffe auf den jeweiligen Mandanten einschränken.

Die nächst "dezentraleren" Lösungen bestehen darin, alle Module für jedes Intranet zu duplizieren bzw. neu aufzubauen. Entweder auf demselben Server mit denselben Datenbanken wie die zentralen Mandanten oder auf neuen virtuellen oder echten Servern, ggf. mit weiteren Liveservern und Datenbankservern (vergl. *Bild3*).

Aus den unterschiedlichen Varianten ergeben sich unmittelbar Vorteile und Grenzen der implementierten Lösungen. Als Vorteil der zentralen Variante kann die Tatsache gewertet werden, dass nicht für jeden Mandanten neue Tablespace in der Datenbank oder neue virtuelle Server angelegt werden müssen. Fragen der Skalierbarkeit betreffen dann vorrangig die eingesetzte Datenbank. Mandanten mit Standardvorgaben (z.B. erste Navigationsebene automatisch importiert, Administratoren und Beispielnutzer des Mandanten mit Standardrechten automatisch erzeugt,...) können mit wenigen Mausklicks von den verantwortlichen Nicht-Technikern neu angelegt werden, wie in *Bild4* gezeigt. Diese einfache Bedienbarkeit ist ein Grund, dieselbe Funktionalität auch für das Anlegen der eigenständigen Projektgruppen und Communities im Dachportal (s.o.) zu verwenden. Änderungen am Datenmodell, den Templates oder generelle Softwareupdates werden unmittelbar für alle Mandanten wirksam.

Bild4 - Mandantenverwaltung für zentrale Mandanten:



Jeder dieser Vorteile kann in anderem Kontext natürlich zum Nachteil werden: So möchten manche Dienststellen eigene Templates benutzen, eigene Applikationen entwickeln, die nicht von allen Behörden genutzt werden können/sollen/dürfen, Modifikationen im Datenmodell speziell für ihre Bedürfnisse vornehmen und echt getrennte Datenbanken, Filesysteme und Webserver verwenden. Deswegen ist es im Landesintranet Niedersachsen ebenfalls vorgesehen, zentrale Mandanten zu einem bestimmten Zeitpunkt abzukoppeln. Ab diesem Zeitpunkt müssen allerdings alle Änderungen an den zentralen Mandanten, die der dezentrale Mandant mitmachen möchte, manuell nachgezogen werden (Redundanz in den Templates, etc.). Wir erwarten derzeit, dass ca. 95% aller Mandanten im Behördennetz auf die zentrale Lösung setzen werden.

5 Ausblick: Ein PHP basiertes Entwicklungs-Framework für personalisierte Desktops

Wie ist es möglich, mit PHP basierten Lösungen, Anschluss an visuelle Entwicklungsumgebungen zu halten, wie sie z.B. derzeit mit Microsoft Visual Studio .NET in aller Munde sind. Unsere Antwort lautet: Durch die Schaffung ähnlicher visueller Umgebungen und mächtigen Frameworks auch für PHP. Die nächste Ausbaustufe im zentralen Dachportal des Landesintranets Niedersachsen ist die Bereitstellung persönlicher Desktops für fortgeschrittene Nutzer: Diese User können in ihrem Browser verschiedene Portlets aus den Bereichen Wissensmanagement, assoziative Suche, Groupware und Redaktionssystem zu persönlichen Desktops zusammenstellen und damit weiterarbeiten. Wir verwenden den Begriff "Portlet" hier nicht im strengen Sinne der Java Portlet Spezifikation (JSR 168), wohl aber in weitestgehender Analogie, was Funktionalität und Architektur betrifft.

Bild5 - Persönlicher Desktop mit Drag und Drop Funktionen:

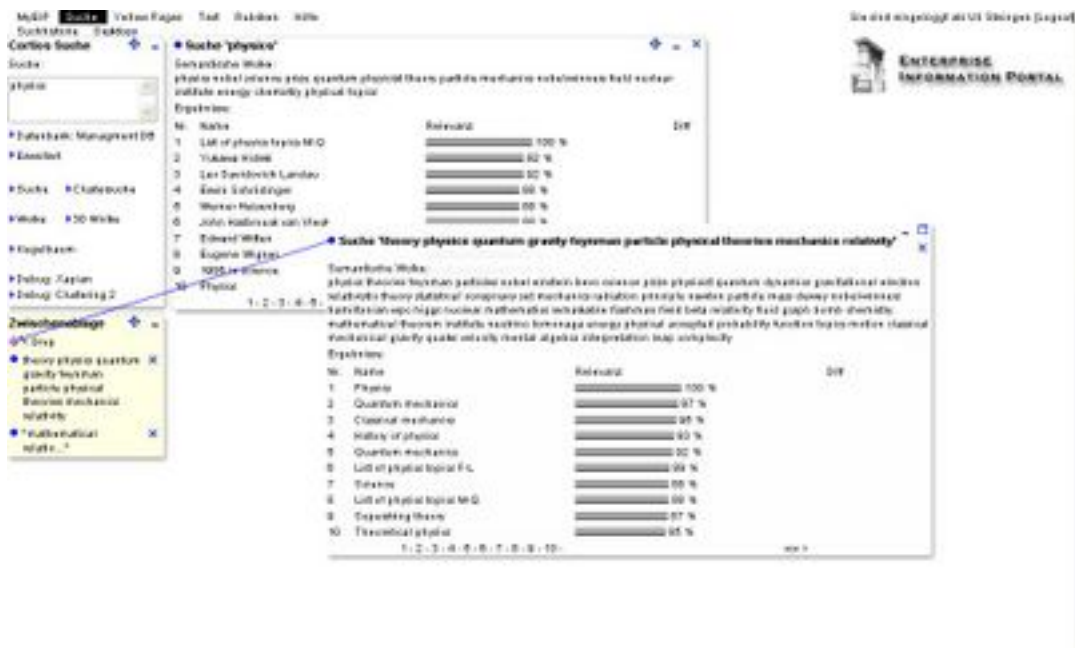


Bild6 - Recherchedesktop:



Bild5 und Bild6 zeigen Beispiele für anspruchsvolle Recherchedesktops. In Bild5 werden Suchergebnisse mit Drag und Drop kommentiert, in die Zwischenablagen geschoben und anderen Personen oder Gruppen verfügbar gemacht. In Bild6 unterstützen eine Macromedia Flash/Actionscript und eine VRML Applikation die Recherche durch die Visualisierung inhaltlicher Zusammenhänge zwischen Texten und Begriffen. Alle Portlet-Fenster sind ein- und ausklappbar und auf dem Desktop frei positionierbar. Die Technologie erzeugt am Ende komplette HTML-Seiten aus einem Guss". Referenzen auf Desktops können problemlos als Bookmarks gespeichert oder als URLs per E-Mail versendet werden.

5.1 Powerslave Enterprise Workflow Engine als grafische Entwicklungsumgebung für Applikationen

Neue Applikationen können vom Entwickler sehr schnell mit der grafischen Entwicklungsumgebung - der Powerslave Enterprise Workflow Engine - erstellt werden. Solche Applikationen können dann entweder im "normalen Intranet erscheinen oder als Portlets auf dem persönlichen Desktop laufen. Im Landesintranet Nieder-

sachsen sind u.a. folgende Portlets vorgesehen: Suche, Ergebnisdarstellung, Projekträume mit Dokumenten-Collections, 2D und 3D Visualisierungen des semantischen Kontextes von Suchanfragen, Kalenderblatt, ECMS inklusive WYSIWYG Editor.

Bild7 - Suchportlet in grafischer Entwicklungsumgebung:

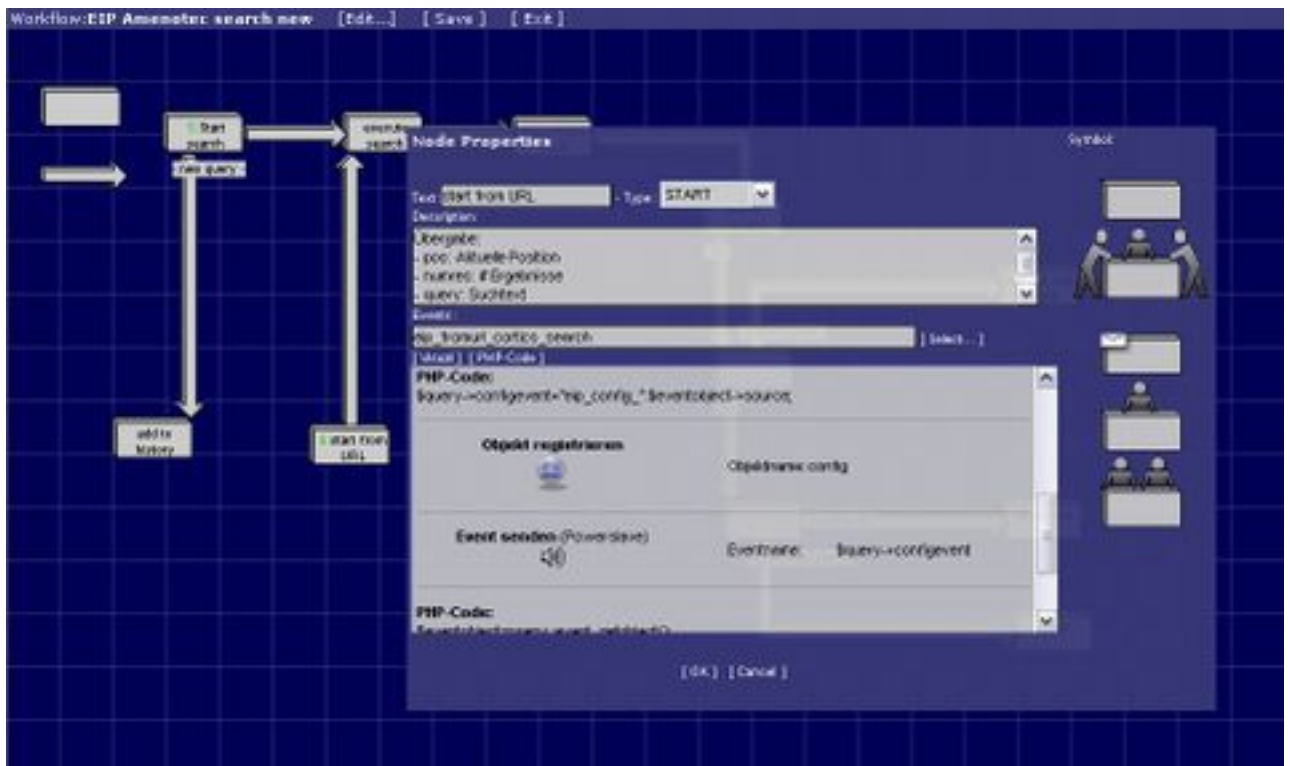
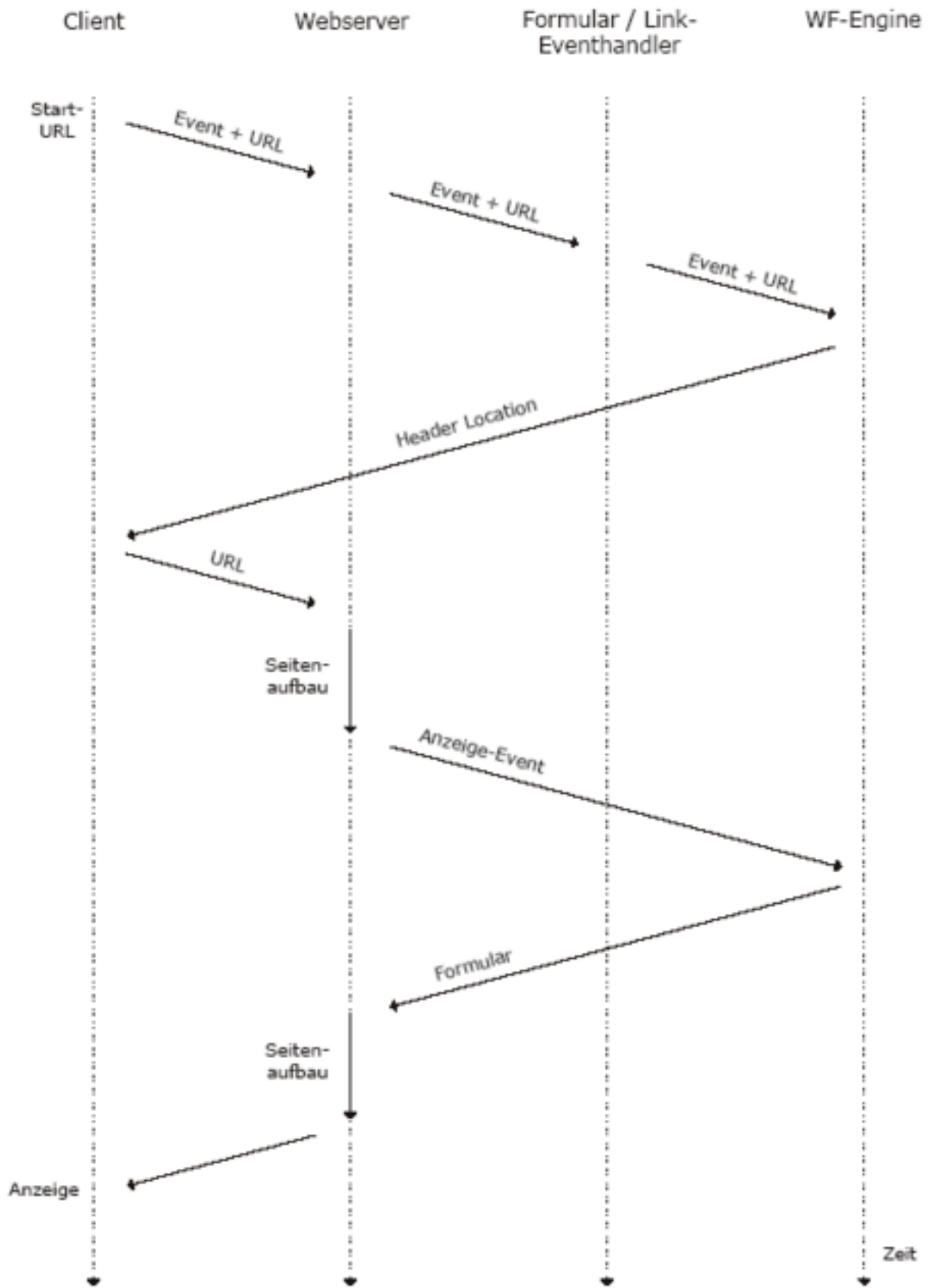


Bild7 zeigt das Portlet für die Suche im grafischen Editor. Der Programmfluss der Applikation wird dabei wie es Workflows bekannt ist als Graph mit Knoten und Kanten modelliert, wobei der eigentliche (PHP-)Programmcode in diesen Elementen verborgen ist. Der Ablauf des Programms ist vollständig ereignisgesteuert. Ereignisse können z.B. im Frontend durch Klicken auf Links oder Formularaktionen ausgelöst werden. Jedem Ereignis kann ein beliebig komplexes (PHP-)Datenobjekt mitgegeben werden. Im graphischen Editor der Workflowengine werden solche und andere Objekte anspruchsvoll visualisiert und dem Administrator in einem Objektbrowser zur Entwicklung angeboten. Im Falle unseres Suchportlets enthält das Ereignisobjekt beispielsweise den Text der Suchanfrage und den gewünschten Zeitraum, auf den die Treffer eingeschränkt werden sollen. Das Frontend zu einem Portlet wird mit Hilfe von Templates realisiert, so dass die eigentliche Programmfunktionalität sauber vom grafischen Ein- und Ausgabefrontend getrennt ist. Das Wechselspiel von Events und templategesteuerter Ausgabe ist in Bild8 als Zeit-Sequenz-Diagramm verdeutlicht.

Bild8 - Zeit-Sequenz-Diagramm beim ereignisgesteuerten Aufbau einer Seite:

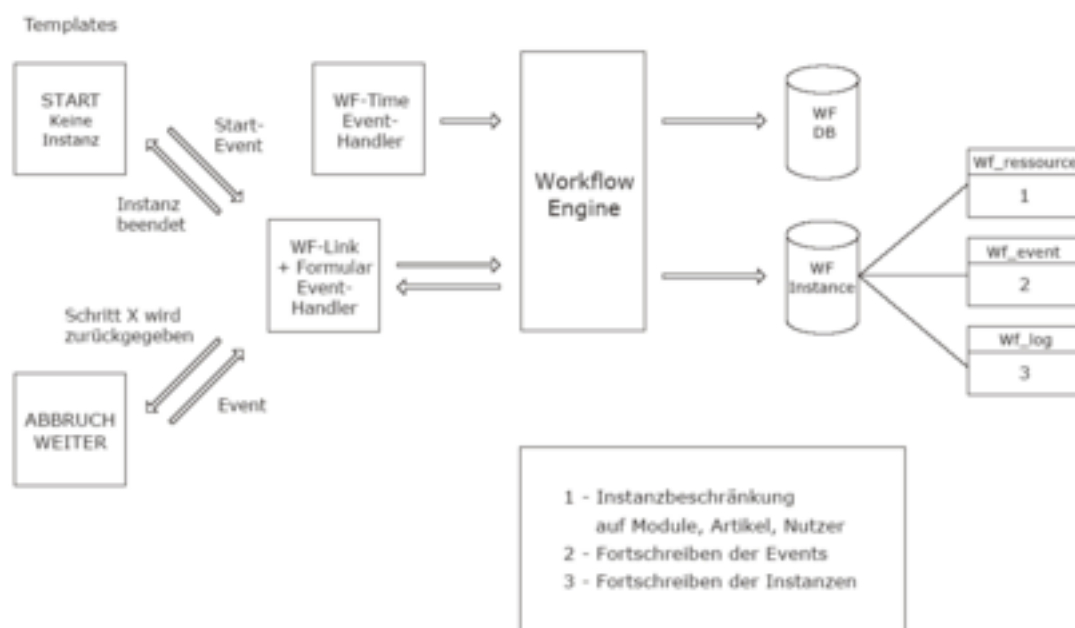


Festzuhalten bleibt, dass die Entwicklungsgeschwindigkeit durch die Einführung einer ereignisbasierten Vorgehensweise (statt mit URLs oder herkömmlichen Formularaktionen zu arbeiten) wesentlich beschleunigt wird. Die Workflowengine stellt dabei mehrere Client-Server-Eventhandler zur Verfügung, u.a. für HTML-Links und -Formulare, Javascript (z.B. für die Drag und Drop Funktionen), Actionscript/Flash und VMRL. Des Weiteren existieren bereits vordefinierte Portlets, die man wieder verwenden kann: Eine besondere Rolle spielt hier das so genannte "Browser-Portlet" für Einbindung externer Webapplikationen (beispielsweise Lotus Notes) inklusive Single-Sign-On auf dem persönlichen Desktop.

5.2 Komponenten des persönlichen Desktops

Sollen viele solcher Portlets oder Applikationen auf einem personalisierten Desktop erscheinen, sind weitere Softwarekomponenten nötig: Die Desktop-Logik selbst ist dabei ebenfalls als Workflow umgesetzt. Dieser Workflow (genauer: eine Instanz dieses Workflows) wird dem eingeloggtten Benutzer zugeordnet und verwaltet alle seine Portlets (in diesem Sinne ebenfalls Workflowinstanzen), die auf seinem persönlichen Desktop ausgeführt werden. Von dieser Desktop-Logik werden auch alle neue Portlets gestartet und Fensteroperationen zum Minimieren, Maximieren und Schließen von Applikationen ausgeführt. Vom Nutzer im Frontend ausgelöste Ereignisse werden auf das zugehörige Portlet auf seinem persönlichen Desktop eingeschränkt, was die Performance im Vergleich zu globalen Ereignissen, die an alle laufenden Portlets aller Nutzer gesendet werden, natürlich erheblich steigert. *Bild9* zeigt die Möglichkeit, Ereignisse auf bestimmte Ressourcen (Personen, Datensätze, Portlets, Desktops, ...) einzuschränken. Der Desktop entspricht somit ungefähr den Portletcontainern in Java-Umgebungen. Da die Fenster mit Hilfe von Templates (mit Titelleiste, den Fensteroperationen, Drag und Drop Punkten...) ausgegeben werden, ist es ein Leichtes, das gesamte Corporate Design der Organisation anzupassen.

Bild9 - Eventhandler in der Workflowengine:



Wie findet nun der Informationsaustausch zwischen den Fenstern/ Portlets/ Applikationen/ Workflowinstanzen statt? Grundsätzlich kann jedes Portlet (bzw. jede Workflowinstanz) jedem anderen Portlet Nachrichten (d.h. Ereignisse mit zugehörigen Informationen im Ereignisobjekt, s.o.) schicken, wenn es von dessen Existenz (insbesondere die UniquePortletID) weiß. Besonders wichtig aber ist der Informationsaustausch zwischen den Portlets auf demselben Desktop. Zu diesen Zweck wurde ein umfangreiches Javascript Drag und Drop Framework entwickelt. So können beispielsweise Suchtreffer aus den Ergebnisportlets in Projekträume der Collectionportlets gezogen werden. Dokumente, Rubriken oder Suchanfragen können in die Portlets für die persönliche Zwischenablage oder für eigene Abonnements gezogen werden (vergl. *Bild5*). Nutzern aus dem LDAP-Portlet können per Drag und Drop Rechte aus dem Rechteportlet zugeordnet werden. Daten aus dem

Kalenderblatt können in Suchformulare gedroppt werden u.v.m. Solche Drag und Drop Aktionen lösen entsprechende Ereignisse aus, auf die betroffene Portlets dann reagieren können, indem sie z.B. zunächst prüfen, ob die Ressource, die gedroppt werden soll, überhaupt kompatibel zu ihnen ist. So macht es keinen Sinn, Benutzer in Datumsfelder oder umgekehrt, Daten in Emailfelder zu ziehen. Anschließend arbeitet das Portlet mit der Information weiter, zeigt sie an (z.B. Einschränkung des Suchzeitraums) oder legt sie im Backend in einer Datenbank ab (z.B. Zuordnung von Rechten). Für jede Ressource (z.B. Dokument, Benutzer, Suchtreffer, Datum, Rubrik, Projektraum...) können zusätzlich umfangreiche Rechte vergeben werden. Dies geschieht im Portalframework durch einen so genannten "Ressourcenpersonalisierer".

5.3 Im Vergleich: Betriebssystem vs. Portalserver mit persönlichem Desktop

In *Tabelle1* werden abschließend nochmals die Funktionen des Desktops mit den Features eines Betriebssystems wie z.B. Windows verglichen.

6 Schlussbetrachtung: Enterprise Portalserver auf der Basis von PHP

Im Einsatz einer Variante unseres Enterprise Information Portals (EIP) beim Land Niedersachsen sehen wir den Anfang einer Entwicklung, in der Knowledge management und Enterprise Content management in den Intranets größerer Organisationen immer näher zusammenrücken. Strategisch bedeutet das für PHP als Entwicklungsplattform: Der gute Ruf, den sich PHP im Bereich ECM/WCM erarbeitet hat, kann Türen auch für anspruchsvollere Applikationen im Bereich Workflow management und Knowledge management öffnen, in denen traditionell andere Plattformen den Markt besetzten. Mit Hilfe ausgefeilter Frameworks wie der hier vorgestellten Workflow engine, dem ECMS und dem Portalserver können in Zukunft schnell neue anspruchsvolle PHP-Applikationen aufgebaut werden.

Tabelle 1: Vergleich: Betriebssystem vs. Portalserver mit persönlichem Desktop

<i>Betriebssystem</i>	<i>EIP</i>
<i>Core:</i>	
Prozesse	Workflow-Instanzen
Speicherverwaltung	Betriebssystem, Webserver
Semaphoren, Monitor	derzeit nicht vorhanden, wird erweitert werden
Dateisystem	Datenbank, Betriebssystem, Dateisystem
Treiber	nicht vorhanden, kein direkter Hardwarezugriff möglich
Eventhandler, Events (Hardware, User Interface etc.)	Workflow Eventhandler (Client, Server), Events
Dateirechte	Ressourcen Personalisierer
<i>User Interface/Desktop:</i>	
Windows API, .NET Framework, etc.	Powerslave/EIP/Workflow API, PHP API, etc.
Windows GUI, HTML	DHTML, HTML, Flash, VMRL, Powerslave Template Engine
Anwendungen	Workflow-Portlets
Window Manager	Desktop Instance, Desktop Templates
Drag und; Drop	Drag und Drop Punkte
Start Menü	Start Menü
Task Manager	Task Ansicht im Startmenü derzeit noch nicht vorhanden
Task Leiste	eingeklappt Ansicht zur Verkleinerung der Portlets, Desktop Zoom
Reset, runterfahren	"Start Menü" - "Alle Portlets beenden"
Benutzer wechseln	Logout/Login
Datei-Explorer	Rubriken (Collections)
Suche	Suche
Kontextmenü	meist nicht nötig, wg. Drag und Drop
URL, Verknüpfung	URL
Zwischenablage	Zwischenablage + Zwischenablage Portlet
Console	nicht vorhanden, Programm-Start durch Start-URL möglich
SSpeichern unter..."/"Öffnen...in Anwendungen	Drag und Drop Punkte
nicht vorhanden	Portlets/Instanzen einfrieren+abspeichern
nicht vorhanden	Desktop Kompletzoom, geplant
nicht vorhanden	Fenster drehen, geplant
offline Betrieb	noch nicht möglich, geplant
HTML-Browser	geplant, aber nicht so wichtig da Web Technologie
externe Geräte/Hardware verwalten, installieren etc.	kein direkter Hardwarezugriff möglich
nicht vorhanden	n:m Ordner - Dateien
nicht vorhanden, nur Sortieren nach Typ	Inhalte nach Typ gruppieren
<i>Integration Windows Desktop<->EIP:</i>	
Windows Explorer	WebDAV
Drag und Drop	Unterstützung durch IE + Zwischenablage Portlet
Windows Anwendungs-Fenster	Portlet in eigenen Fenster Öffnen, in Arbeit
URL (z.B. in Emails, Bookmarks etc.)	Portlets definieren Start-URLs
Sonstiges	Windows Client in System Tray (noch nicht vorhanden)