

# Chancen und Risiken von AJAX in Fachanwendungen

Markus Eisele

markus.eisele@msg-systems.com



# Agenda

- Software im Unternehmenseinsatz
- AJAX im Unternehmen
- Anwendungsdesign und Usability
- Denkbare Einsatzszenarien
- Wo AJAX sich nicht eignet
- Blick in die Glaskugel

# Über mich

- Markus Eisele
- Software Architekt im CoC IT-Architecture im Cross-Bereich Software-Engineering der msg systems ag.
- BEA Certified Server, SAP WebAS Consultant, MCSE
- BEA Technical Director
- Tätigkeitsschwerpunkte JEE, SAP, Webframeworks, Portale, DMS/CMS, UI und Usability
- Autor für verschiedene deutsche IT-Magazine
- Im Web etwas mehr unter <http://www.eisele.net/> und natürlich auch unter <http://www.msg-systems.com/>

# Was ist AJAX?

- AJAX ist ein **Designpattern** zur Kommunikation von von Webfrontends mit den Serverkomponenten
- „**AJAX Engines**“ kapseln über JavaScript-Klassen die XML HTTP Kommunikation zum Server
- In Summe sind aktuell ca **150++** verschiedene **Frameworks** verfügbar, welche das AJAX Pattern für Entwickler einsetzbar machen.
- Frameworks haben verschiedene Ausrichtung (Clientseite, Serverseite, Widget-Ansatz/Generierung)

# Software im Unternehmenseinsatz

- Werkzeug zum Unternehmenserfolg
  - Kein Selbstzweck
- Geeignet für spezielle Infrastruktur
  - Intranet,- und Extranet
- Stabil und Zuverlässig
  - Was nicht funktioniert => kostet Geld
  - Falsche Ergebnisse => kosten Geld
  - Verlorene Ergebnisse => kosten Geld
  - Daten sind vertraulich
- Kosteneffizient
  - Entwicklung, Betrieb, Wartung-/Weiterentw.

# Software im Unternehmenseinsatz

muss „**rock-solid**“ sein und unterliegt darum zumeist Vorgaben und Richtlinien

- Strategie-Roadmaps und Blueprints
- Styleguides und UI-Vorgaben
- Implementierungsvorgaben
- Qualitätsrichtlinien
- Sicherheitsvorgaben
- Betriebsrichtlinien

# Software im Unternehmenseinsatz

- Besonderheiten von **Fach-/Unternehmensanwendungen**
  - Harte Performance Vorgaben
  - Bearbeitung und Anzeige von Massendaten
  - Batchverarbeitung
  - Workflowanteile
  - Synchrone Verarbeitung
  - Komplexe oder auch ungewöhnliche Oberflächen
  - Begrenzter Nutzerkreis

# Software im Unternehmenseinsatz

- Besonderheiten bei der **Technik**
  - Integration in Unternehmensstandards
  - Softwareentwicklung
  - Softwarearchitekturen
  - Softwarequalität
  - Entwickler-, Funktions-, Last- und Stabilitätstests
  - Anwendungssicherheit
  - Systemüberwachung
  - Komplexe Webinfrastrukturen und Lastverhalten

# AJAX im Unternehmen

AJAX im Unternehmenseinsatz bedeutet mehr, als „download and run“

- + Noch recht neue Technologie mit durchaus bekannten Tücken
  - + Besonderheiten von Unternehmensanwendungen
  - + Vorgaben, Richtlinien und Standards
- = Paket von Chancen und Herausforderungen  
..oO(oder auch einfach Problemen ...)

# AJAX im Unternehmen

- Integration in Unternehmensstandards
  - Plattformen, Server und Umgebungen
  - Zentral verwaltete und standardisierte Clients (Hardware, Sicherheitseinstellungen)
  - AJAX und MVC Frameworks
  - Evaluierungen / Freigabeprozesse / Untersuchungen / Prototypen

# AJAX im Unternehmen

- Softwareentwicklung, und -architektur
  - Werkzeugunterstützung für Entwickler
  - Debugging und Fehlerhandling
  - JavaScript als Programmiersprache
  - Schichtentrennung und Komponentenbildung
  - Modellierung von AJAX Funktionen

# AJAX im Unternehmen

- Softwarequalität
  - Codingstandards
  - Software Metriken und deren Messung
  - Nightly Builds bzw. Cruise Control

# AJAX im Unternehmen

- Testen
  - Last- und Stabilitätstests
    - Capture Replay mit JavaScript
  - Entwicklertests
    - Integration in Buildprozesse
  - Funktionstests
    - Verschiedene Endgeräte

# AJAX im Unternehmen

- Anwendungssicherheit
  - Allgemeine Einschränkungen wie bei allen Webanwendungen
  - Verlockung zur clientseitigen Berechtigungsprüfung
  - Internetbrowser und Cache

# AJAX im Unternehmen

- Systemüberwachung / Systemsmanagement
  - Runtime Tracing und Logging
  - Fehlerüberwachung
  - Performance Monitoring

# AJAX im Unternehmen

- Komplexe Webinfrastrukturen
  - Deployment
  - Clustering und Failover
  - Loadbalancing
  - Netzwerklast

# Anwendungsdesign und Usability

- Oberflächenanpassungen und Styleguides
- Aus Anwendungen heraus drucken
- Scrollen und dynamische Seitenlängen
- Innovative Nutzerkonzepte
- Asynchron und im Hintergrund heisst nicht „unsichtbar“

# Anwendungsdesign und Usability

- Browser Navigation
- Bookmarks und Favoriten
- Langsame Browser
- Teilfunktionen und Gesamtkontext

# Denkbare Einsatzszenarien

- Technische Lösungsansätze
  - Standards- und Produktintegration
    - JEE5 / .NET
    - Portale und andere Kaufsoftware
  - Serverseitige Entwicklung und Generierung
  - Pattern und Best Practices
  - Werkzeugunterstützung

# Denkbare Einsatzszenarien

Anbindung von „langsamen Systemen“ /

## Workflow Ansatz

- Klassische Lösung:  
JMS Queue und Arbeitsvorrat, Emailbenachrichtigung
- Ajax Lösung:  
Asynchrones Laden im Hintergrund  
Generieren von Meldungsfenstern / „Live Arbeitsvorrat“

# Denkbare Einsatzszenarien

## Komplexe Formulare

- Klassische Lösung:  
Langsame „Klick-Orgien“ durch die Anwendung
- Ajax Lösung:  
Nur Seitenfragmente werden bei Bedarf nachgeladen  
und aktualisieren den Fensterinhalt bei Bedarf

# Denkbare Einsatzszenarien

## Datenbäume und Topologien

- Klassische Lösung:  
Load on Startup (ggf. aufsplitten auf „Lazy server loading“ bei erneutem Klick auf einen Knoten)
- Ajax Lösung:  
Dynamisches, bedarfsgerechtes und asynchrones Lazy Loading der benötigten Datensätze in den Client.

# Denkbare Einsatzszenarien

## Daten Filtern und Sortieren

- Klassische Lösung:  
Server Round-trip pro Aktion und Datenaufbereitung auf dem Server
- Ajax Lösung:  
Client-Funktionalität zur Sortierung und Filterung der Datensätze.

# Denkbare Einsatzszenarien

## Automatische Vervollständigung und Hinweise

- Klassische Lösung:  
Leere Textfelder mit beschreibendem Label  
Vorausgefüllte Textfelder mit Beispielinhalten
- Ajax Lösung:  
Automatische Vervollständigung bei Werten, die bekannt sind. Ähnlich „Google Suggest“

# Denkbare Einsatzszenarien

## Community Funktionen

- Klassische Lösung:  
Konfigurierter Page-Refresh alle x-Sekunden
- Ajax Lösung:  
intelligentes Einbinden von „pseudodynamischen Bereichen“, „Push“-ähnliches Verhalten für den Anwender transparent simulieren.

# Wo AJAX sich nicht eignet

## Technisch:

- Reine JavaScript Client-Frameworks
  - K.O. bei Entwicklung und Betrieb
- Terminal und Kiosk-Systeme
  - K.O. bei Gesamt-Performance und Testbarkeit
- Embedded Browser
  - K.O. bei Speicher, JS Funktionen und Performance
- UI Widgets und Spielereien
  - K.O. weil verfügbare Elemente den Bedarf von Fachanwendungen nicht ausreichend abdecken.

# Wo AJAX sich nicht eignet

- Inhaltlich:
  - Einfache Formulare
  - Einfache Navigation
  - Komplexe Seitenoperationen
  - AJAX zum Oberflächendesign
  - Unnütze Oberflächenelemente
  - Massendatendarstellung

Name	Benutzername	C...	Speicher...
ieexplore.exe	eiselem	94	192.072 K
java.exe	eiselem	04	65.548 K

# Blick in die Glaskugel

„The bottom line is it [AJAX] can result in an improved user experience, which offers significant business value, but that is not automatic. Many developers will do what they always do—they don't understand user-centric design. We've seen this with Java applets. We've seen this with Flash.”  
(Ray Valdes, Gartner)

# Blick in die Glaskugel

- Wunsch zur verbesserten Nutzerinteraktion im Web bleibt Treiber von Innovationen
- Einsatz von AJAX in Standard-Produkten wird zunehmen (Portale, CRM, ERP, etc.)
- SOA und AJAX

# Blick in die Glaskugel

- „Industrial strength“ AJAX wird **serverseitig generiert** und lehnt sich an moderne **Komponentenmodelle** an (bspw. JSF, Atlas)
- Die **Menge der Frameworks** wird sich **konsolidieren**
- Es wird **stabiles und etabliertes Standard AJAX** im Rahmen von **bekannten Plattformen** geben (JEE/.NET)
- Es werden mehr **BestPractices** und **Vorgaben** entstehen
- Die **Technologiespirale** bei UI Technologien hat das Ende aber auch noch nicht erreicht („XHTML2, XFrames, XForms“ und und und....)

# Blick in die Glaskugel

„Die Kunst des Fortschritts besteht darin, inmitten des Wechsels Ordnung zu wahren und inmitten der Ordnung den Wechsel aufrechtzuerhalten.“

(Alfred Northon Whitehead)

# Links und Informationen

- <http://publikationen.eisele.net/>
- <http://ajaxpatterns.org/Patterns/>
- <https://blueprints.dev.java.net/ajax-faq.html>
- <https://blueprints.dev.java.net/bpcatalog/ee5/ajax/javascript-recommendations.html>
- <http://www.ajaxmatters.com/blog/ajax-tutorials/>
- <http://www.lowing.org/ajax/>
- <http://www.ajaxmatters.com/blog/articles/>
- [http://de.wikipedia.org/wiki/Ajax\\_%28Programmierung%29](http://de.wikipedia.org/wiki/Ajax_%28Programmierung%29)
- [http://labs.google.com/suggest/ ;\)](http://labs.google.com/suggest/)