



Patent Nr. 1 176 507 beim Europäischen Patentamt

Technical Whitepaper

XANDRA Technology[®] der main {GRUPPE} ist ein Softwareentwicklungssystem für die effiziente Erstellung von Internet und Intranet-Applikationen.

Inhalt

Was ist XANDRA®	2
Leistungsspektrum von XANDRA®	2
Konzept und Architekturmodell	3
Technik am Client	4
Ein Wort zu Beginn	4
Vollkommen neuartige Kommunikationstechnologie zwischen Client und Server	4
The XANDRA®-Way: XML als Transportmedium	6
Vollkommen neue Art des Session Managements	7
Realtime Tracking	8
Neue Möglichkeiten der Erzeugung von HTML-Oberflächen	8
Herausragende Benutzerführung und Usability des XANDRA®-Systems	10
FAQs zur Technik am Client	11
Technik am Server	13
FAQs zur Technik am Server	15
Voraussetzungen für den Einsatz von XANDRA®	16
Clientseitig	16
Serverseitig	16

Fehler! Textmarke nicht definiert.

Was ist XANDRA®

XANDRA® ist ein Softwareentwicklungssystem für die effiziente Erstellung von qualitativ hochwertigen, personalisierten Systemen für das Intra- oder Internet. Dabei wird von XANDRA® die komplette technische Infrastruktur für die Entwicklung und den Betrieb einer Anwendung bereitgestellt.

Leistungsspektrum von XANDRA®

XANDRA® bietet folgendes Leistungsspektrum:

- vollkommen neues, patentiertes Verfahren der Kommunikation zwischen Browser und Web-Server
- J2EE-konform, extrem schnelle Ladezeiten der Web-Seiten
- weltweit das einzige Framework, welches direkt in einem Standard- Browser mit sogenannten **Web-Services** arbeitet
- durch die Verwendung von Web-Services erfolgt eine Reduktion des Entwicklungsaufwandes am Server um **30 – 50 Prozent**
- **Realtime Tracking und Clickstream-Analysen** bis auf die Ebene von Mausbewegungen¹
- einzigartige Kombination von Clickstream-Daten mit Businessdaten; für Marketingabteilungen sind diese Daten extrem wertvoll

¹ laut Computerwoche Ausgabe 39/2001, Seite 25, forscht das MIT gerade an einem ähnlichen Verfahren. Wir sind über den Status der Forschung bereits hinaus und vermarkten es.

Konzept und Architekturmodell

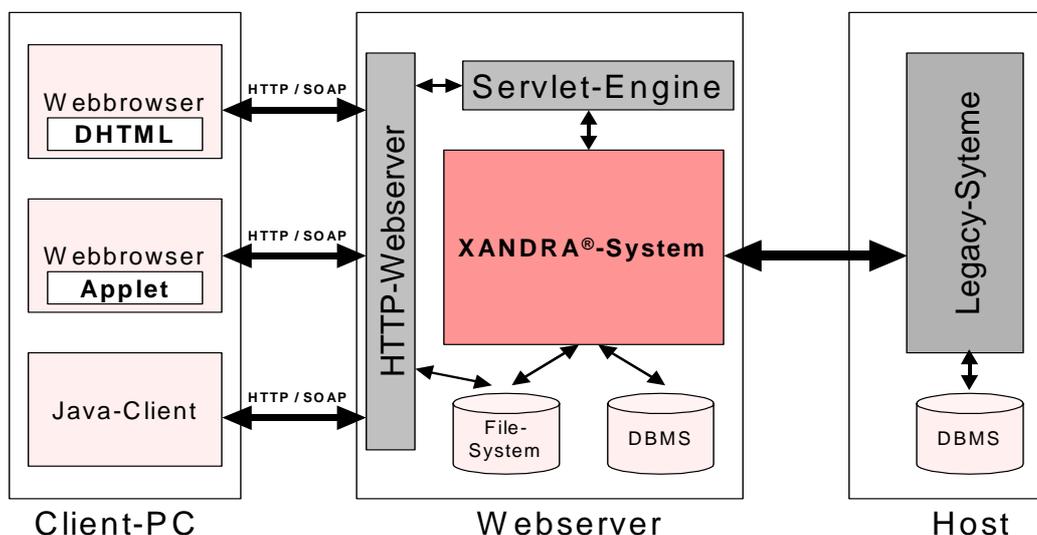
XANDRA® basiert konsequent auf Internet-Technologien. Dabei wird auf eine klare Trennung zwischen Präsentation und Geschäftslogik gesetzt. Zwischen Client und Server werden Web-Services auf der Basis des Standardprotokolls HTTP und SOAP verwendet.

XANDRA® stellt eine Vielzahl vorgefertigter Web-Services zur Verfügung, z.B.:

- Komplette Authentifizierungs-Mechanismen (Login, User-Verwaltung)
- Datenbank-Anbindung über den Database Service (z.B. Oracle, DB2, Access)
- E-mail Dienste
- Kalender-Funktionen mit Terminverwaltung
- SSL-Modul für verschlüsselte Kommunikation
- Host-Anbindung über JMS
- komplettes Realtime Tracking und Clickstream-Modul
- kompletter PIN/TAN- Service für gesicherte Transaktionen

Clientseitig wird die Darstellung der Oberfläche und die gesamte Interaktion mit dem Anwender des Systems über einen Webbrowser durchgeführt. **Es muss keine weitere Software installiert oder vorhanden sein.** Die Darstellung im Webbrowser geschieht über DHTML und Java Script. Für komplexe Oberflächen in einem Intranet kann die Oberfläche auch über ein Java-Applet oder eine Java-Applikation dargestellt werden. In jedem Fall bleibt die Schnittstelle zu den Server-Komponenten gleich: Web-Services auf SOAP- Basis.

Serverseitig setzt XANDRA® auf eine mehrschichtige Architektur mit vordefinierten Diensten, den sogenannten Services. Jeder Service hat ein genormtes Interface, so dass beliebige neue Services hinzugefügt, oder bestehende durch neue Implementierungen ersetzt werden können.



Die Abbildung zeigt die technische Infrastruktur und grobe Architektur von XANDRA®.

Innerhalb des XANDRA®-Systems ist die Architektur in verschiedene Ebenen aufgeteilt. Jede dieser Ebenen hat genau definierte Schnittstellen und Funktionalitäten. Dadurch können einzelne Ebenen durch neue Implementierungen ausgetauscht werden, ohne die gesamte Architektur zu beeinträchtigen.

Technik am Client

Ein Wort zu Beginn

Da XANDRA® konsequent auf der Internet-Technologie basiert, können damit alle im Web üblichen Technologien verwendet werden. Bei der Entwicklung der Client-Seite bestand nicht der Anspruch, Bewährtes zu ersetzen, sondern nach Verbesserungspotential gängiger Standard-Methoden zu suchen.

Das Ergebnis ist eine Reihe von *Technical-Highlights*, die es in dieser Form kaum im Web zu finden gibt und die Entwicklern die Arbeit wesentlich erleichtern. Ziel dieses Kapitels ist die Beschreibung dieser Highlights und nicht die Wiederholung gängiger Vorgehensweisen bei der Entwicklung von Web-Applikationen.

Technische Highlights von XANDRA®:

- vollkommen neuartige Kommunikationstechnologie zwischen Client und Server
- vollkommen neue Art des Session Managements
- Realtime Tracking
- neue Möglichkeiten der Erzeugung von HTML-Oberflächen
- herausragende Benutzerführung und Usability des XANDRA®-Systems
- konsequente Umsetzung des Thin-Client Ansatzes

Die Punkte im einzelnen:

Vollkommen neuartige Kommunikationstechnologie zwischen Client und Server

Die Kommunikation zwischen Web-Client und Web-Server hat sich seit den frühen Tagen des Internets nicht geändert:

- man nehme ein HTML-Formular
- schicke es an ein Script (z.B. CGI, ASP, Servlets)
- werte dort die Daten aus und
- sende dem Client eine Ergebnisseite zurück

Die Daten des Formulars werden als Key / Value-Paare an das Script gesendet und dort verarbeitet.

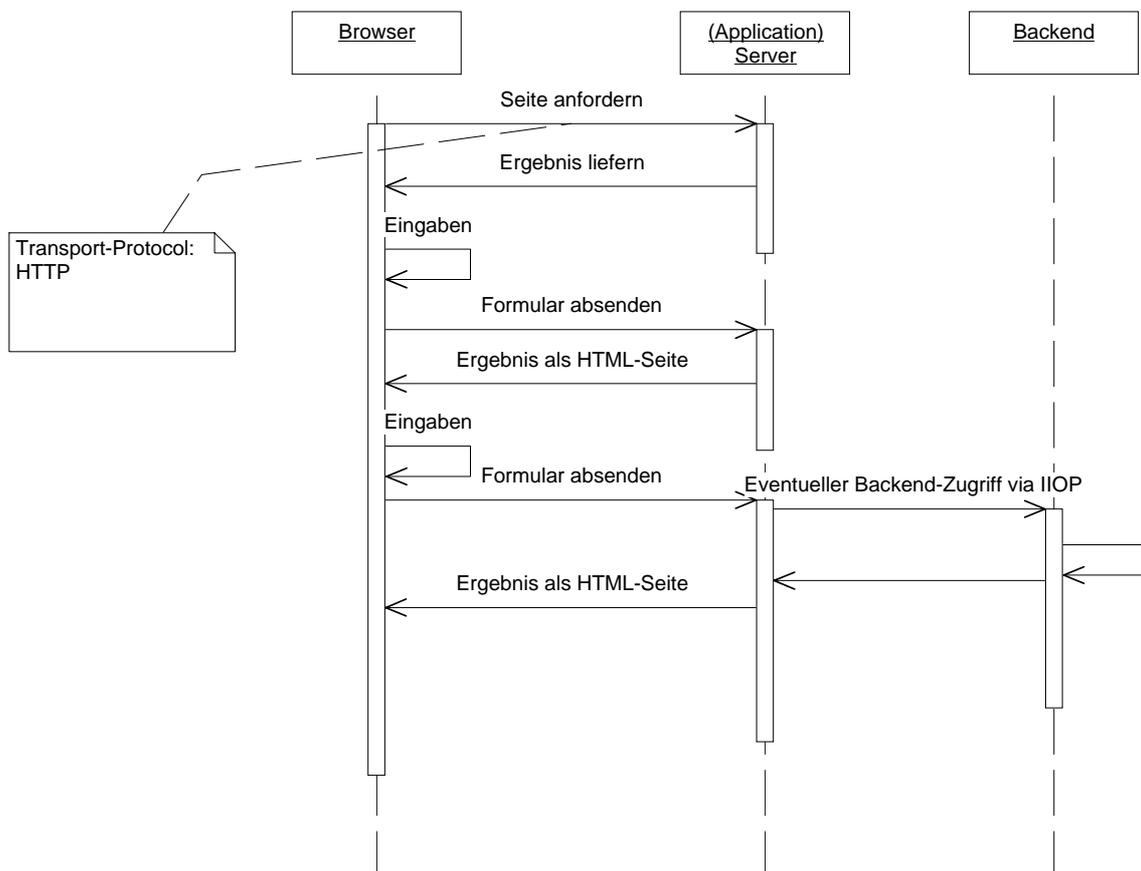
Verfahren bei komplexen Anwendungen:

- Für jede Seite muss ein Script oder „Businessobjekt“ programmiert werden.
- „Zurückblättern“ erfordert, die „alten“ Werte am Server zu rekonstruieren. Hierzu werden die Daten dynamisch in die HTML-Formulare eingetragen. Es findet eine Mischung von Daten und Präsentation der Daten statt.

- Jeder Seitenwechsel bedeutet einen Server-Kontakt, um die nächste Seite als Ergebnis der vorherigen Eingabe dynamisch zu ermitteln. Die Ermittlung der Daten bedeutet bei vielen gleichzeitigen Benutzern eine erhebliche Server-Belastung.
- Ist die Ergebnisseite stark von den vorherigen Eingaben abhängig, erhöht sich die Komplexität zusätzlich, da jede Seite nicht eine, sondern viele Ergebnis-seiten nach sich ziehen kann.

Auf solche Seiten hat die Standardtechnik eine große Menge feingranularer und recht unübersichtlicher „Server-Objekte“ zur Folge, selbst wenn diese - gemäß einem Model-View-Controller (MVC) Muster - zerlegt werden. Diese Objekte müssen innerhalb einer Kontrollinstanz zu einer Transaktion gebündelt werden, um komplexe Abläufe überhaupt modellieren zu können.

Der „Stand der Technik“ in Form eines UML-Diagramms:



Die Technik der HTML-Formulare wurde nicht dafür entworfen, eine komplexe Abfolge von voneinander abhängigen Dialogen miteinander in Beziehung zu setzen. Dies ist aber eine Grundvoraussetzung einer nicht-trivialen Internet-Applikation ...

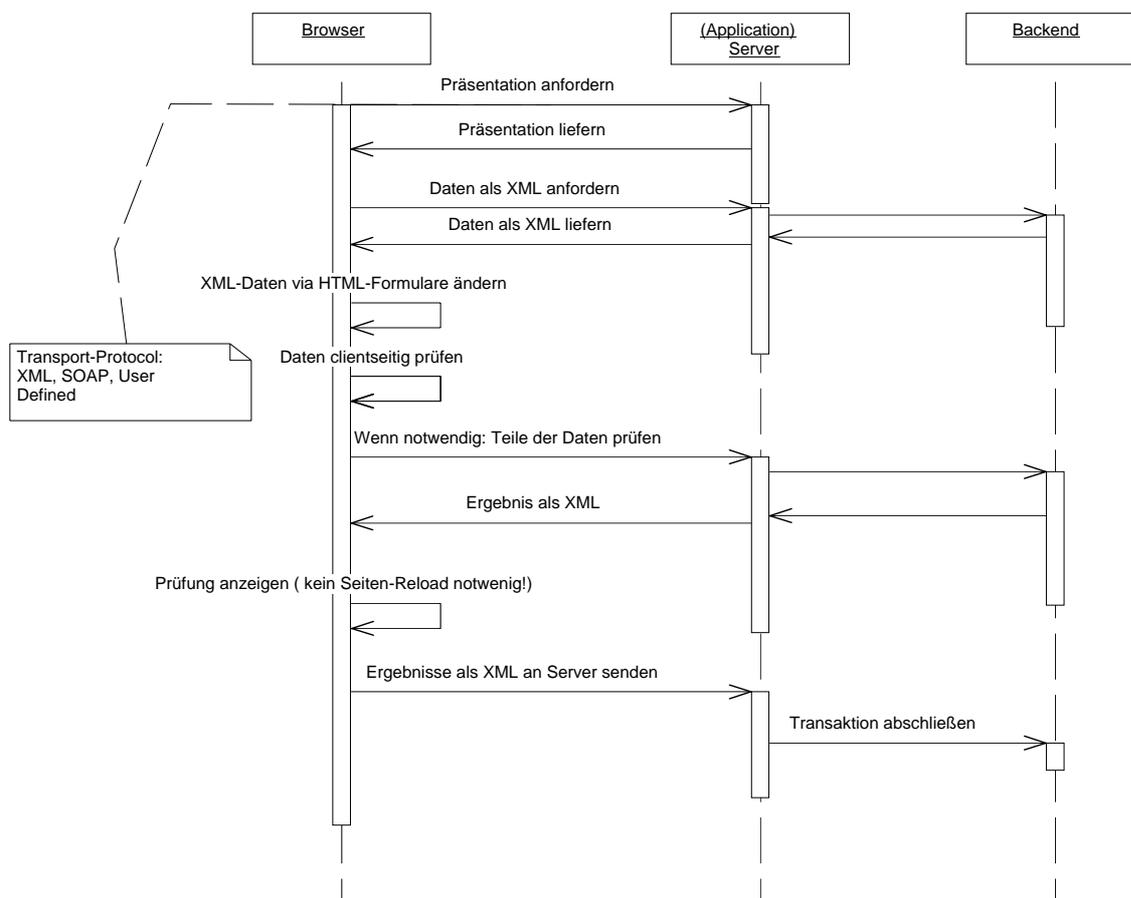
The XANDRA®-Way: XML als Transportmedium

Vorweg: Mit XANDRA® kann auf die altbewährte Art und Weise weitergearbeitet werden. Wer aber den neuen Weg kennen gelernt hat, wird nicht mehr so weiterarbeiten wollen: Die Kommunikation zwischen Web-Client und Web-Server erfolgt in XANDRA® auf folgende Art und Weise:

- Man definiert ein XML-Schema, welches die Daten Ihrer Dialog-Abfolge festlegt.
- Man entwickelt ein serverseitigen Web-Service, welcher die gesamten Daten Ihres Geschäftsprozesses als XML-Strom verarbeitet. Hierbei wird in der Regel **lediglich die „Built-In“- Services von XANDRA® kombiniert**. Wichtig ist: Die serverseitigen Kommandos haben *keinen* Bezug zu den HTML-Formularen. Die Daten müssen sich in keiner Weise an den Formularen orientieren.
- Man entwickelt die grafische Gestaltung der Dialoge für die Daten in Form von Java Script.
- Die Daten können nun in jedem Browser erfasst werden. Man muss sich an dieser Stelle nicht um das Zuordnen der Daten zu Formularen, Senden der Daten oder Session Management kümmern. Dies wird vom XANDRA®-Laufzeitsystem transparent übernommen.
- Bei Bedarf werden *alle* Daten des Geschäftsvorfalles in Form von XML an den Server gesandt, der diese verarbeitet.

Die Eleganz dieses Ansatzes liegt in der vollständigen *Trennung von Daten und Präsentation*. Auch am Client sind Daten streng von der Präsentation getrennt. Der Datenaustausch erfolgt nicht seitenorientiert, sondern *transaktionsorientiert*. Die Komplexität am Server sinkt dramatisch.

Folgende Abbildung zeigt den Ansatz:



Zu beachten ist, dass:

- zwischen Client und Server XML (SOAP) und keine „serialisierten Formulare“ übertragen werden. Die Umwandlung von Formulardaten in „Businessdaten“ entfällt komplett.
- Ergebnisse „en bloc“ an den Server gesendet und dort verarbeitet werden
- die Zahl der Server-Zugriffe bei komplexen Dialogabfolgen nicht höher wird
- Server-Zugriffe *ohne reload der HTML-Seite* durchgeführt werden können. Damit sind z.B. komplexe Prüfungen innerhalb eines Formulars möglich. Z.B.: Überprüfung der Postleitzahl innerhalb eines Formulars über einen Web-Service.
- der Server über andere SOAP-Architekturen ansprechbar ist und z.B. als Web-Service für .NET oder SUN's ONE verwendet werden kann.

Fazit: Mit XANDRA® müssen Applikationen endlich nicht mehr wie „aufgebohrte“ Terminal-Applikationen entwickelt werden.

Vollkommen neue Art des Session Managements

Da es sich bei http um ein zustandsloses Protokoll handelt, muss die bei klassischen Client / Server Architekturen vorhandene Session künstlich nachgebildet werden. Hierzu sind eine Reihe von Techniken bekannt z.B.:

- Cookies
- URL-Rewriting
- Hidden-Fields in den HTML-Seiten

Das Session Management XANDRA® basiert auf Cookies oder alternativ auf URL-Rewriting. Diese Verfahren werden jedoch erheblich vereinfacht. Daten einer Session werden so lange wie möglich am Client gehalten, wodurch

- der Server stark entlastet
- die Komplexität des Session Managements am Server reduziert

wird. Das Speichern der Session-Daten am Client erfolgt für den Entwickler vollkommen transparent. Er muss sich schlichtweg nicht darum kümmern.

Bei Bedarf werden die gesammelten Daten als SOAP-Kommando an den Server gesendet, der die Daten verarbeitet. Fehlermeldungen können über SOAP an den Client zurückgegeben werden, ohne dass eine Daten-Reload notwendig wäre.

Fazit: XANDRA® verhält sich bezüglich des Session Managements wie eine klassische Client / Server Architektur. Das Session Management erfolgt transparent durch das Laufzeitsystem.

Realtime Tracking

XANDRA[®] erfasst das Surf-Verhalten des Users ab dem *ersten Kontakt in Echtzeit*. Erfasst werden in der Standard-Variante:

- besuchte Links
- Verweildauer auf jeder Seite
- alle bisher auf der Seite gemachten Eingaben

Optional können

- Anzahl von Mausbewegungen pro Zeiteinheit und Seite
- Anzahl der Tastatur-Anschläge pro Zeiteinheit und Seite

erfasst werden. Diese Daten können dazu verwendet werden, die Akzeptanz der Seite zu erhöhen. Realtime Tracking ist besonders effizient, wenn es darum geht, individuell auf die Eingaben des Benutzers zu reagieren. So kann z.B. aus der Information, dass der Benutzer

- 35 Jahre alt ist,
- sich sehr schnell und sicher durch die Webseite bewegt

in Verbindung mit demoskopischen Daten mit einer gewissen Wahrscheinlichkeit auf ein über dem Durchschnitt liegendes Einkommen schließen. Mit XANDRA[®] ist es möglich, den Benutzer *beim ersten Kontakt* zielgruppengerecht aufbereitete Informationen anzubieten oder die Vermutung über das Einkommen durch gezielte, weiter Fragen zu eruieren.

Ist der Benutzer über User-ID und Passwort identifizierbar, können die Daten unter Berücksichtigung der Datenschutzbestimmungen am Server abgespeichert und beim erneuten Kontakt sofort verwertet werden. XANDRA[®] wird dann zu einem *lernenden System*, welches sich mit jedem Kontakt in einem höheren Maße an den Kunden anpasst.

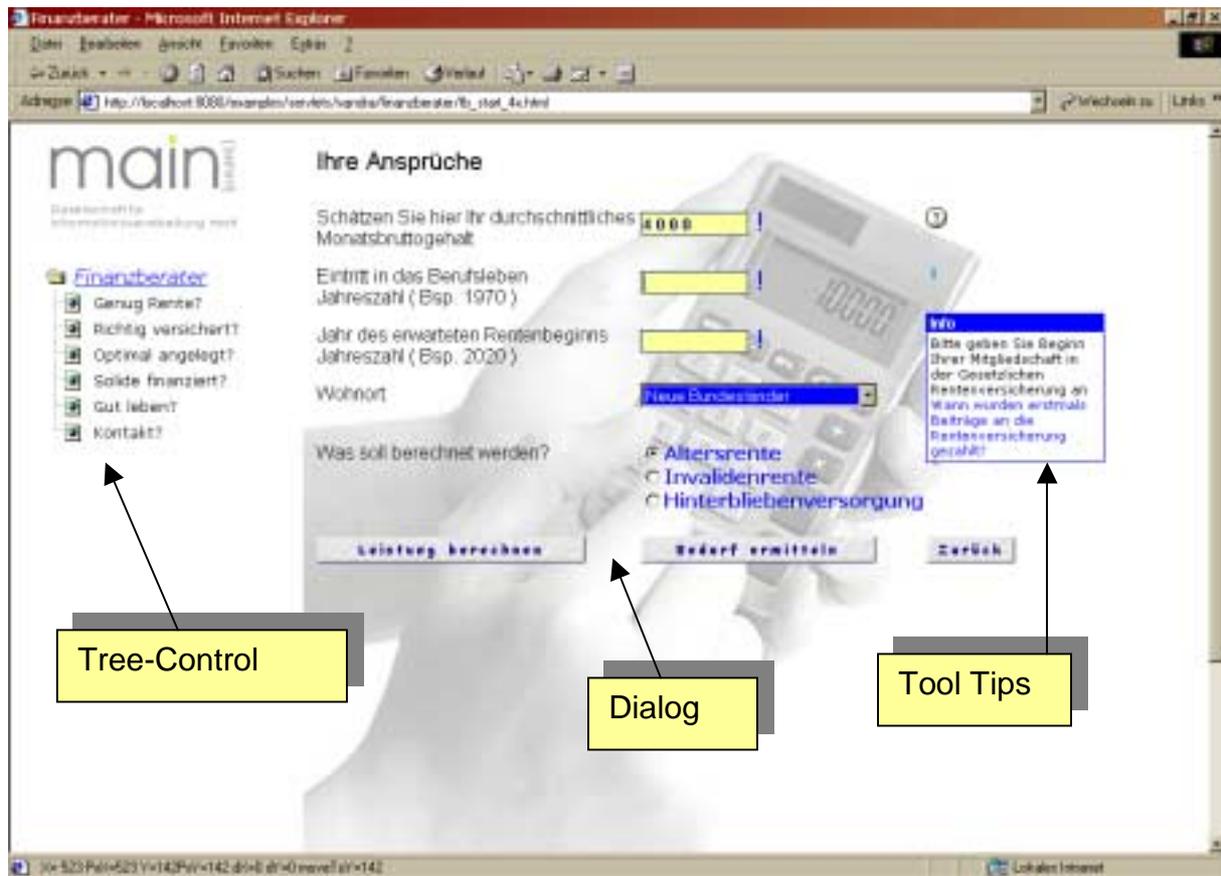
Fazit: Mit XANDRA[®] ist es möglich, die Kunden ab dem ersten Kontakt in einer bisher nicht gekannten Qualität kennen zu lernen.

Neue Möglichkeiten der Erzeugung von HTML-Oberflächen

Mit XANDRA[®] ergeben sich vollkommen neue Gestaltungsmöglichkeiten für Webseiten. Die Oberflächen werden nicht mehr in HTML, sondern über ein Java Script-Objektmodell erzeugt. XANDRA[®] besitzt die umfangreichste, kommerziell verfügbare Bibliothek zur Erzeugung von grafischen Oberflächen. Neben Standard-Controls gehört zu XANDRA[®]:

- eine Reihe von neuen Controls, die vollkommen neue Nutzer-Interaktion ermöglichen
- automatische Validierung von Eingabefeldern am Client
- automatische Erzeugung eines „stateful environments“ am Client

Die folgende Abbildung zeigt einige Beispiele von Controls:



Das XANDRA[®] Objektmodell ist für Java-Programmierer sehr leicht zu erlernen. Nach zwei Tagen Schulung können bereits Oberflächen mit XANDRA[®] entwickelt werden. Der große Vorteil dieser Art der GUI-Entwicklung liegt jedoch in der möglichen, einfachen Modularisierung des Codes.

Mit XANDRA[®] können wiederverwendbare Komponenten entwickelt werden, die

- sich in jede XANDRA[®]-Oberfläche einbinden lassen
- voll mit dem Backend integriert sind

Eine Reihe vorgefertigter Komponenten ist zu XANDRA[®] erhältlich, z.B. eine komplette Login-Steuerung, die mit zwei Zeilen(!) Code einbindbar ist:

```
g_login_dlg=SECURITY_LOGIN.createLoginDlg("LOGIN",onSendDataDone,onNewUser,
10,10);
g_login_dlg.invalidate();
```

Das erzeugt folgenden Output:



Die Login-Komponente ist sofort an das Backend-System angebunden und kann auf eigene Bedürfnisse angepasst werden.

Fazit: *Kein kommerziell verfügbares System kann diese Features bieten.*

Herausragende Benutzerführung und Usability des XANDRA®-Systems

Grafischen Oberflächen, die mit XANDRA® erzeugt werden, sind sehr benutzerfreundlich gestaltbar. Die Akzeptanz wird durch

- die konsequente Verwendung von Tool Tips
- kontext-sensitive Online-Hilfen
- Realtime Tracking und „lernende Anpassung an den Benutzer“²

erreicht. Mit konventioneller Technik sind solche Anwendungen nur mit immensem Aufwand möglich.

Fazit: *Derartig benutzerfreundlich ist das Web nur sehr selten.*

² Mit XANDRA® kann die ganze Seite „on the fly“ geändert werden, wenn ein bestimmtes Benutzerprofil erkannt wurde.

Konsequente Umsetzung des Thin-Client-Ansatzes

XANDRA[®] verwendet weder Plug Ins, noch Java. Es benötigt lediglich einen Standard-Browser ab der Version 4.x mit Java Script. Die Datenübermittlung erfolgt mit Hilfe der normalen HTML-Standards.

Fazit: XANDRA[®] realisiert einen echten „Thin Client“ mit „Fat Functionality“.

FAQs zur Technik am Client

Benötigt man für XANDRA[®] Java Script am Client?

Ja, XANDRA[®] arbeitet nur, wenn Java Script eingeschaltet ist.

Ist die Programmierung der Oberflächen nicht sehr aufwändig im Vergleich zu „normalem“ HTML?

Mit „normalem“ HTML ist es praktisch unmöglich, die anspruchsvollen Oberflächen von XANDRA[®] zu erstellen. Der Grund: Der HTML-Code müsste mit sehr viel browserabhängigem Code versehen werden, um z.B. Tool Tips zu erzeugen. Deshalb wird der HTML-Code über ein Objektmodell gekapselt und je nach verwendetem Browser „on the fly“ am Client generiert.

Die XANDRA[®]-Klassen wurden mit dem Ziel entworfen, anspruchsvolle grafische Schnittstellen mit einem einfachen Objektmodell zu erzeugen. Ein Entwickler, der Grundkenntnisse in einer objektorientierten Sprache wie z.B. Java besitzt, kann nach zwei Tagen Schulung bereits produktiv XANDRA[®]-Oberflächen entwickeln.

Kann XANDRA[®] auch mit Standard-HTML arbeiten?

Ja, die Vorteile von XANDRA[®] kommen jedoch nicht voll zur Geltung.

Welche Vorteile hat das XANDRA[®]-Vorgehen sonst noch?

- XANDRA[®]-Seiten werden extrem schnell geladen, da nur noch Java Script, jedoch kein (aufgeblähter) HTML-Code mehr übertragen wird.
- XANDRA[®]-Code lässt sich modularisieren. Es ist üblich, immer wiederkehrende grafische Details als Unterprogramm zu implementieren. Beispiel ist die Navigations-Leiste. Dieser Code wird nur einmal übertragen und danach aus dem Cache des Browsers verwendet.
- Mit XANDRA[®] werden eine Reihe *vorgefertigter Komponenten* mitgeliefert, die sofort mit Anbindung an das Backendsystem verwendet werden können. So ist eine komplette Benutzerberechtigung mit zwei Zeilen(!) Code in die Seite einbindbar.

Wie groß ist XANDRA[®]?

Das Basissystem umfasst ca. 80 KB Java Script-Code. Zusätzliche Module können über eine Loader nachgeladen werden. Um die Ladezeiten dieser 80 KB zu vermindern, werden die Scripts i.R. via Pre Fetching bei der Startseite im Hintergrund geladen.

Was ist Pre Fetching von HTML-Seiten?

Unter Pre Fetching versteht man das Laden von HTML-Seiten in den Browser-Cache bevor diese Seiten tatsächlich angezeigt werden. Pre Fetching erhöht die Performance am Client erheblich und wird von XANDRA[®] unterstützt.

Benötigt XANDRA[®] Applets?

Nein. Oberflächen können aber bei Bedarf mit Applets entwickelt werden.

Ändert sich bei der Verwendung von Applets der Aufbau der Applikation?

Nein. Das Kommunikationsverfahren zwischen Client und Server ist in beiden Fällen gleich. Lediglich die Oberfläche wird mit anderen Mitteln entwickelt.

Hat XANDRA[®] Probleme mit Firewalls?

Nein, alle Protokolle setzen auf http und Standard-Ports auf.

Ist XANDRA[®] für das Intranet geeignet?

Ja.

Benötigt XANDRA[®] Plug Ins?

Nein.

Kann XANDRA[®] um neue GUI-Elemente erweitert werden?

Ja, das API zur Erstellung von Controls ist offen und dokumentiert.

Benötigt man für XANDRA[®] Cookies am Client?

Nein, XANDRA[®] kann auch ohne Cookies arbeiten.

Welche Browser werden von XANDRA[®] unterstützt?

XANDRA[®] unterstützt Netscape ab Version 7.0, Internet-Explorer ab Version 5.x und Mozilla ab Version 1.4.

Damit deckt XANDRA[®] mehr als 95% der im Internet verwendeten Browser ab.

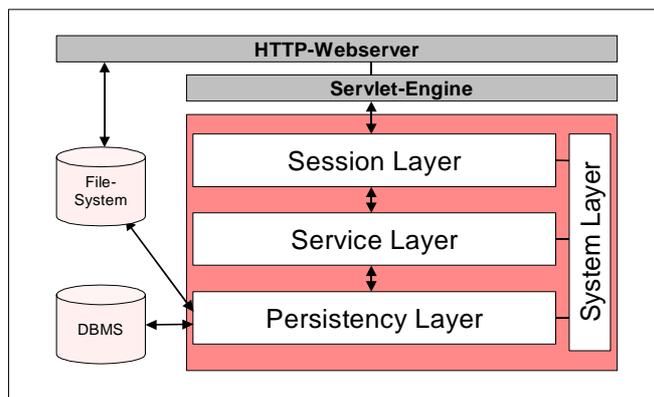
Unterstützt XANDRA[®] SSL?

XANDRA[®] unterstützt SSL.

Technik am Server

Serverseitig baut XANDRA[®] auf eine mehrschichtige Architektur. Vom Client wird über einen HTTP-Request eine Anfrage zum Webserver geschickt. Dieser leitet die entsprechende Anfrage über eine Servlet an das XANDRA[®]-System weiter, welche dann einen Geschäftsprozess in einer Transaktion abarbeitet.

Ein derartiger Geschäftsprozess kann sowohl eine einfache Datenbanktransaktion (z.B. „Suche einen Kunden“) als auch ein komplexer Vorgang - wie die Aufbereitung eines Vertragswerkes mit Ausdruckmöglichkeiten - sein. Einstiegspunkt für das XANDRA[®]-Kernsystem ist immer ein „Worker“, welcher über ein generisches Kommando-Objekt einen oder mehrere Services aufruft. Dieses Kommando-Objekt beschreibt den gewünschten Geschäftsprozess und ist für die Ausführung des gewünschten Service' zuständig.



Die Abbildung zeigt die grobe Aufteilung von XANDRA[®] in die verschiedenen Ebenen bzw. Layer.

Der **Session Layer** übernimmt die Verwaltung und Steuerung der einzelnen Sessions in Zusammenarbeit mit dem Application Server. Alle Aufgaben innerhalb einer Session werden dabei von einem Servlet als zentralen Einstiegspunkt entgegengenommen und an einem Worker delegiert.

Der Worker kann je nach eingegangenen Request die verschiedenen Services des **Service Layers** aufrufen und die geforderten Aufgaben durchführen lassen. Dazu erzeugt der Worker ein entsprechendes Kommando-Objekt und schickt dieses an den zuständigen Service im **Service Layer**.

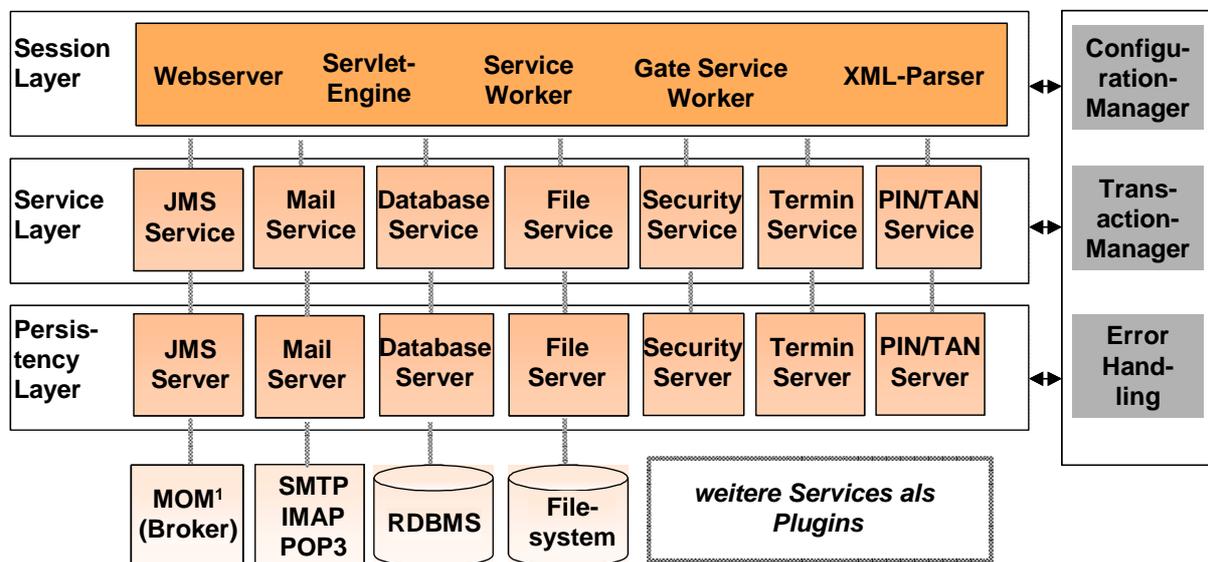
Der **Service Layer** nutzt für die Durchführung von Aufgaben den **Persistency Layer** und Dienste des **System Layer**. Der **Persistency Layer** enthält die Zugriffsschicht für Datenbanksysteme und beliebige andere Datenquellen.

Der System Layer bietet neben den allgemeinen Systemdiensten für das XANDRA[®]-System auch Dienste für Logging & Tracing über alle anderen Schichten an. Über Einstellungen in einer Konfigurationsdatei können die gesamten Ausgabeströme auf den Bildschirm oder in eine Datei gelenkt werden.

Das XANDRA[®]-System stellt seine Dienste in Form von parametrisierbaren Services zur Verfügung. Über Konfigurationsdateien im XML-Format kann die Laufzeitumgebung eines Services gesteuert werden. Jeder Service lädt über einen Configuration-Manager bei seiner Initialisierung die entsprechende Konfigurationsdatei.

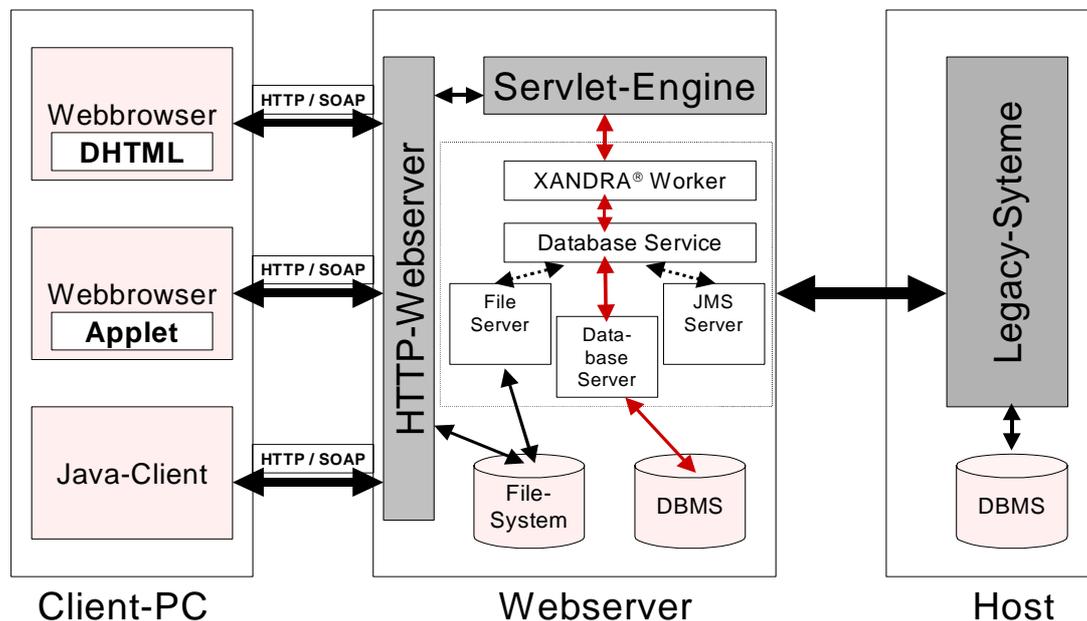
Alle Services bieten ein standardisiertes Interface, sodass ein durchgängiges und einheitliches Programmiermodell für die Nutzung des XANDRA[®]-Systems gegeben ist.

Jeder Service nutzt für seine Aufgaben einen oder mehrere Server. So wird z.B. der Database Service einen Database Server für den Zugriff auf ein DBMS nutzen. Die Services können ihre Aufgaben auch auf andere Server delegieren.



Durch die strikte Aufteilung zwischen Service und Servern kann ein Service verschiedene Ausprägungen eines Servers nutzen. Der Database Service nutzt im Standardfall den Database Server für die Ausführung von Datenbankoperationen. Der Database Server mit einer Ausprägung für RDBMS kann gegen einen Server mit einer Ausprägung für ein ODBMS ausgetauscht werden. Über die Einstellungen in der Konfigurationsdatei zum Database Service kann dann wahlweise zwischen den beiden Servern umgeschaltet werden, ohne in die Implementierung einzugreifen oder den Sourcecode neu zu übersetzen. Anfragen können von den Database Service bei Bedarf auch über den JMS-Server an ein anderes System weitergeleitet werden oder unter Nutzung des File Servers abgearbeitet werden.

Der Zugriff auf ein Datenbanksystem mit dem XANDRA[®]-System geschieht wie folgt:



Die roten durchgezogenen Pfeile zeigen den Standardweg für den Datenfluss über den Database Service. Alternativ kann per Konfiguration der Datenfluss auch über den File Server in das File System oder über den JMS-Server zu einem beliebigen anderen Datenhaltungssystem gelenkt werden.

FAQs zur Technik am Server

Können neue Services hinzugefügt werden?

Durch die offene Architektur ist es problemlos möglich, neue Services zu implementieren und einzubinden. Das bedarf lediglich bestimmter Schnittstellen.

Lässt sich XANDRA[®] in eine EJB-Architektur einbinden?

Grundsätzlich ja. Je nach gewünschten Persistenzmechanismus (CMP oder EMP) und den zur Verfügung stehenden Container muss der Database Service eventuell angepasst werden. Eine Anpassung der gesamten Architektur an die neue EJB Spezifikation und die von der Firma SUN vorgeschlagene „Connectoren“ ist für Mitte 2002 vorgesehen.

Können die Services auch in einem eigenen Systemen angesprochen werden?

Ja. Durch die generische Schnittstelle eines Services kann dieser auch direkt von anderen Systemen angesprochen werden.

Können Datenbanken parallel angesprochen werden?

Ja, es können über den Database Service mehrere Datenbanken parallel betrieben werden. Sie müssen dazu für jede Datenbank eine eigene Instanz des Database Service mit einer zugehörigen Konfigurationsdatei verwenden. Jeder Database Service wird dazu bei seinem ersten Aufruf einen Connection Pool mit den Angaben aus der Konfigurationsdatei initialisieren.

Voraussetzungen für den Einsatz von XANDRA®

Die notwendigen Voraussetzungen für die Installation und den Betrieb eines Systems auf der Basis von XANDRA Technology® sind gering. Sie umfassen:

Clientseitig

- Vorhandensein eines Webbrowsers wie dem Internet-Explorer ab 4.x oder dem Netscape ab 4.x und höher. Netscape 6.1 ab Anfang 2002.
- eine funktionsfähige Interent- oder Intranetverbindung.

Serverseitig

- einen installierten Webserver oder einen Application Server für Java-Umgebungen
- Java-Runtime-Umgebung ab JDK 1.3
- Java-Laufzeitumgebung mit einer Servlet-Engine ab JDK 2.1
- ein installiertes und konfiguriertes XANDRA®-System.