BOXALINO
eBusiness Software

BOXALINO WHITEPAPER

Kunde / Auftraggeber	BOXALINO
Autor(en)	BOXALINO Team
Letzte Änderung	März 2006
Version	6.2
Dokumenten-Nummer	Doku-001-4

BOXALINO AG, Technoparkstrasse 1, 8005 Zürich
Telefon 043 210 33 63, Telefax 043 210 33 64
info@boxalino.ch, www.boxalino.ch

Rechtliche Hinweise

Bei der Zusammenstellung von Texten und Abbildungen wurde mit größter Sorgfalt vorgegangen. Trotzdem kann das vorliegende Handbuch noch Druckfehler oder drucktechnische Mängel aufweisen. Die Angaben in diesem Handbuch werden regelmäßig überprüft, eventuell notwendige Korrekturen sind in den folgenden Auflagen enthalten.

Dieses Produkt wird mit den entsprechenden Software-Anwendungen, die von einer Drittfirma entwickelt wurden, vertrieben. Empfangsbestätigungen (Bestätigungen), Urheberrechte, Verzichtserklärungen und weitere Rechtszugeständnisse für diese Produkte finden Sie in der Readme.txt Datei.

Java™, Java™ Runtime Environment, Java™ Virtual Machine, JSDK 2.0 Servlets, Java™ Script sind Warenzeichen der SUN Microsystems GmbH.

Windows® 2000, Windows® 98, Windows® 95, Windows NT®, Windows XP® Microsoft® XML Parser, Microsoft® SQL Server sind eingetragene Warenzeichen der Microsoft Corp.

Pentium® III ist ein eingetragenes Warenzeichen der Intel Corporation.

Andere hier erwähnte Produkt- und Firmennamen sind möglicherweise geschützte Warenzeichen ihrer jeweiligen Eigentümer.

Alle angemeldeten und eingetragenen Marken Dritter werden lediglich als beschreibende Hinweise im Sinne des § 23 Nr. 3 MarkenG verwendet.

Inhaltsverzeichnis

1 Einleitung

1.1 Was ist BOXALINO, welche Aufgaben werden damit gelöst?

BOXALINO ist eine Software, welche Ihnen verschiedene Funktionen anbietet. So können Sie zum Beispiel:

- Internet-Auftritte mit BOXALINO verwalten – Content Management
- online-Shops betreiben – eCommerce
- Portale für Mitarbeiterinnen und Mitarbeiter betreiben – Intranet
- Portale für einen geschlossenen Benutzerkreis erstellen – Extranet
- Adressen verwalten und Kundenbeziehungen pflegen – Customer Relationship Management
- Dokumente effizient verwalten – Document Management System
- online-Zeitschriften publizieren – Redaktionssystem
- Newsletter verschicken, Umfragen durchführen
- Informationsextrahierung mittels Heuristiken
- und vieles mehr

Zudem haben Partnerfirmen von BOXALINO Funktionen für spezifische Anwendungen/Branchen/Märkte realisiert, wie z.B.

- Crossmedia-Lösungen/Katalogproduktion
- Gemeindelösungen
- Verbandslösungen

Und last but not least stehen Ihnen die Bausteine all dieser Lösungen für eigene Anwendungen zur Verfügung!

Wir nennen das die „BOXALINO Application Toolbox“ - der Baukasten für umfassende und leistungsfähige Software-Lösungen.

Diese Lösungen sind in der Regel sogenannte n-tier Lösungen mit Web-Frontends und/oder Rich-Clients und können natürlich mehrsprachig sein. Dabei unterscheiden wir noch zwischen reinen online-Anwendung und offline-fähigen Anwendungen.

BOXALINO als Application Toolbox

- ist standardbasiert (JAVA, .NET, XML, (X)HTML, WebServices, etc.)
- hilft, die Vorteile einer Standardsoftware mit den Vorteilen der Individualprogrammierung zu verbinden
- hilft, die Anwendungserstellung substantiell zu beschleunigen und die Qualität auf einem hohen Niveau sicherzustellen
- hilft, änderungsfreundliche Anwendungen zu erstellen

1.2 Wie ist BOXALINO entstanden?

Im Jahre 1998 wurde auf Basis von langjähriger Projekterfahrung mit verteilten Applikationen im Banken-/Versicherungsbereich BOXALINO entwickelt.

Seither sind aus Anwendungen in anderen Branchen viele Funktionen, Module – eben unsere Boxen – dazugekommen.

Es wurde speziell darauf Wert gelegt, dass eine klare, verständliche Aufgaben-Trennung und eine klare Trennung von Inhalt, Design und Funktion ein wartbares System ergibt.

BOXALINO nutzt konsequent Standards und ergänzt mit eigenen Konzepten und Klassenbibliotheken, wo dies der Zielsetzung dient.

1.3 Für wen ist BOXALINO geeignet?

Heute nutzen unterschiedliche Kunden unsere Software:

- Web-Agenturen
- Software-Entwickler
- Beratungs-Unternehmen
- Unternehmen mit internen Informatik-Abteilungen
- Bundesämter, öffentliche Verwaltungen, Kirchen
- Parteien und Verbände
- Handels- und Produktionsbetriebe

1.4 Praktische Vorteile beim Einsatz von BOXALINO

- Das Projektrisiko, die Realisierungszeit und die Kosten werden stark reduziert
- Sie profitieren von vielen erprobten Funktionen aus diversen Anwendungsbereichen und können mit gleichem Aufwand auch mehr Funktionalität bereitstellen
- Sie sind in der Lage, schnell auf ändernde/neue Kunden- und Marktbedürfnisse zu reagieren und Ihre Investitionen zu schützen
- Neue Technologien können Sie schnell und effektiv nutzen
- Lösungen werden auf einheitliche Art und Weise realisiert – Sie sind nicht von einzelnen Fachspezialisten abhängig
- Zukunftssichere Technologien schützen Ihre Investitionen
- Sie können bestehende Anwendungen auf neue Technologien migrieren – spezielle „Re-Engineering' Boxen unterstützen Sie dabei

2 Das BOXALINO Konzept

2.1 umfassend – anpassbar – modular

BOXALINO bietet umfassende, praxiserprobte Funktionalitäten, welche auf kundenspezifische Bedürfnisse so angepasst werden können, dass die Lösung immer noch von Aktualisierungen der Basissoftware profitiert – also releasefähig bleibt.

Das zugrunde liegende Konzept ist die Modularisierung, welche über diverse Schnittstellen klar definiert, wie die Aufgaben, Kompetenzen und Verantwortungen der einzelnen System-Teile gegliedert sind.

BOXALINO ist somit auch eine optimale Plattform für die Integration bestehender Systeme/Software. Dank der diverser Schnittstellenmöglichkeiten und vielen vorgelösten Funktionen ist eine Integration von bestehender Software/Infrastruktur gut möglich.

2.2 box-by-box

Unsere Kunden können eine Lösung Schritt für Schritt auf- und ausbauen. Dies erlaubt kontrolliert und kompakt zu starten und mit den Bedürfnissen mitzuwachsen.

2.3 Standardsoftware und Individualprogrammierung

BOXALINO kombiniert die Vorteile von Standardsoftware mit den Vorteilen der Individualprogrammierung.

Durch die Ergänzung von bestehenden – in der Regel konfigurierbaren – Standardboxen und eigenen Boxen, werden die Vorteile von Standardsoftware mit den Vorteilen der Freiheiten der Individualprogrammierung kombiniert und 1 + 1 ergibt 3.

2.4 Customizing – Anpassung an konkrete Kundenbedürfnisse

Dazu braucht es verschiedene Möglichkeiten zum Customizing:

- Einfach zu implementierende Java-Interfaces für häufig verwendete Spezialkonfigurationen, welche die Fähigkeiten der Parametrisierung übersteigen, wie beispielsweise für komplexe Rabatt- und Lieferkostenberechnungen

- Java-Interfaces und diverse Hilfsbibliotheken unterstützen bei der Implementation eigener, anpassbarer und releasefähiger Boxen

- Metamodell-gesteuerte Konfiguration der Datenstrukturen (ohne SQL-Kenntnisse nutzbar) erlauben Anpassungen im Minutenbereich, welche sonst Stunden/Tage brauchen würden

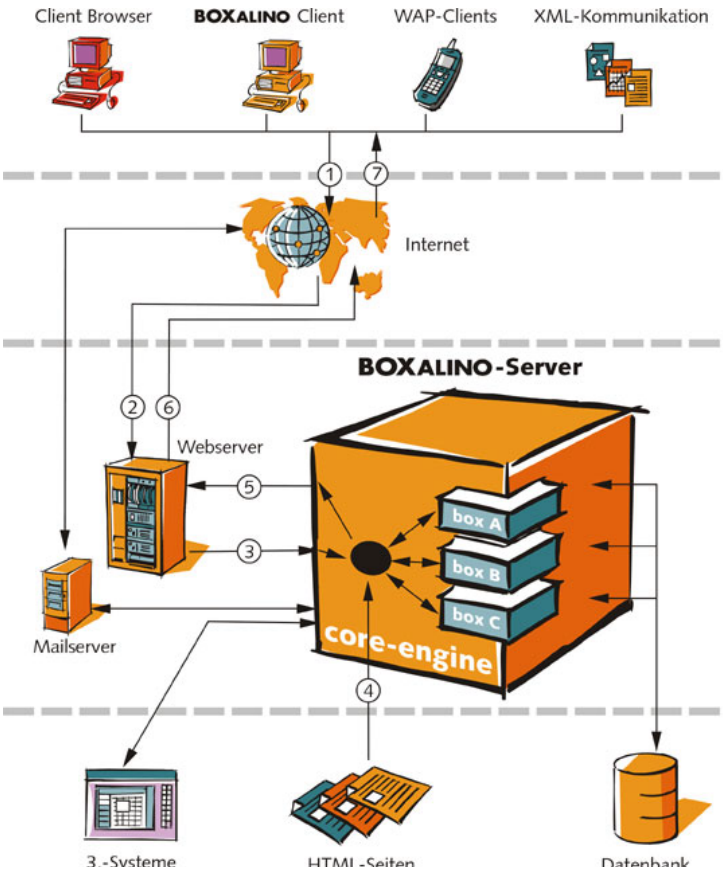
- Server-seitiges JavaScript ermöglicht beliebige Freiheiten beim Customizing, ohne dabei die Komplexität unnötig zu erhöhen

- Diverse Boxen, welche das Customizing vereinfachen stehen Ihnen zur Verfügung – z.B. auch Boxen für einfach anpassbare Workflows

- völlig frei bestimmbare Nutzungsoberflächen (HTML, XHTML, Rich-Clients mit .NET und Java)

3 Die Funktionsweise

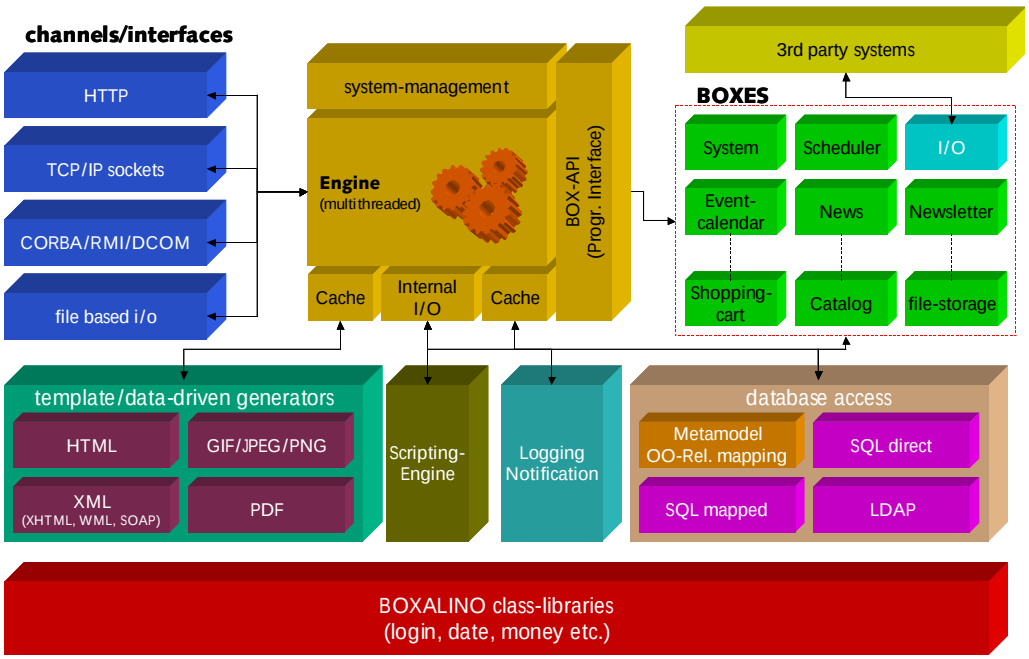
3.1 Wie funktioniert BOXALINO?



Folgende Schritte zeigen die Funktionsweise:

1. Der Client schickt via Browser eine Anfrage ins Internet
2. Das Internet leitet diese zum Web-Server weiter
3. Der Web-Server leitet die Anfrage wiederum an den BOXALINO Server weiter
4. Die core-engine verteilt Eingabe-Werte (Formularwerte, Parameter des Hyperlinks) an die Boxen und liest dann die HTML-Seite, welche zurückgegeben werden soll
5. Die core-engine kombiniert mit Hilfe der Boxen die Vorlage mit Datenbank-Werten und gibt die Resultatseite an den Web-Server zurück
6. Der Webserver gibt die Seite via Internet weiter
7. Der Client erhält die neue Seite

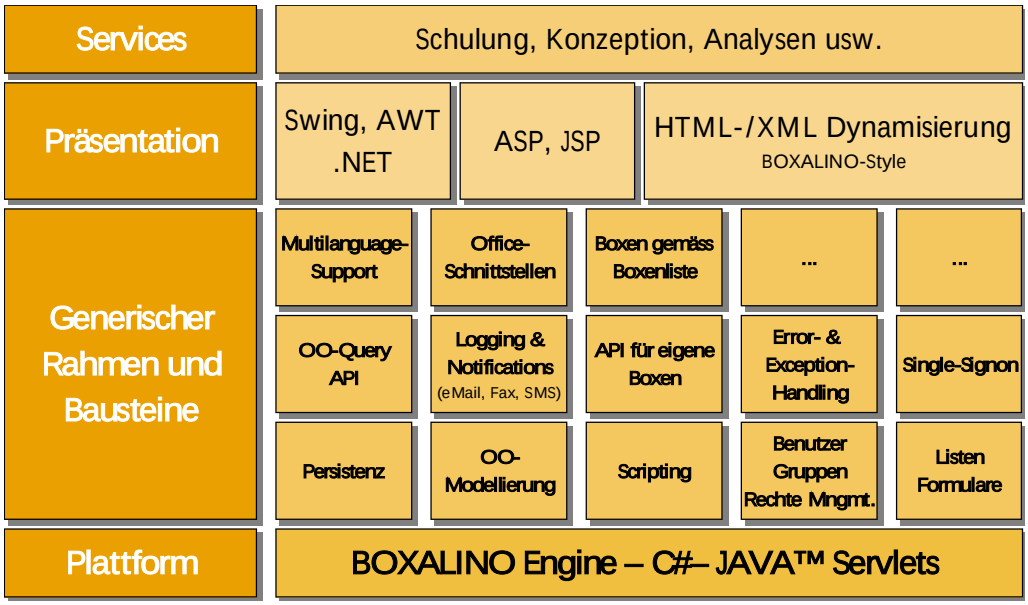
4 Die innere Struktur – Eigenschaften des Systems



4.1 Kurzbeschreibung

- BOXALINO kann über verschiedene Kanäle (blau) angesprochen werden
- Die Engine beinhaltet die komplexeren Systemteile, wie z.B. Caches, Systemmanagement-Aufgaben, Boxen-API, Multithread-Support usw.
- Die Boxen (im Bild grün) sind die eigentlichen Funktionsbausteine
- Diverse Daten-/Template-gesteuerte Generatoren erzeugen den Output
- Eine Server-seitige Scripting-Engine erlaubt leistungsfähiges Scripting in JavaScript
- Für Aufzeichnungen (Logging) und Benachrichtigungen steht eine eigene Komponente zur Verfügung
- Für den Datenzugriff steht eine leistungsfähige, Metamodel-gesteuerte OO-Relation-Bridge zur Verfügung, Boxen können auch direkt via SQL und anderen Datenzugriffstechniken (XML) arbeiten
- Funktionen zur Historisierung, zur fachlichen Benachrichtigung ermöglichen einfache Lösungen bei hohen Anforderungen an die Nachvollziehbarkeit
- Diverse Funktionen stehen als Basis-Bibliothek zur Verfügung

5 Die Application Toolbox



Die Application-Toolbox beinhaltet alle Bestandteile, um leistungsfähige Lösungen zu realisieren, welche auf mobilen Endgeräten bis zu ge-clusterten High-End Systemen betrieben werden können.

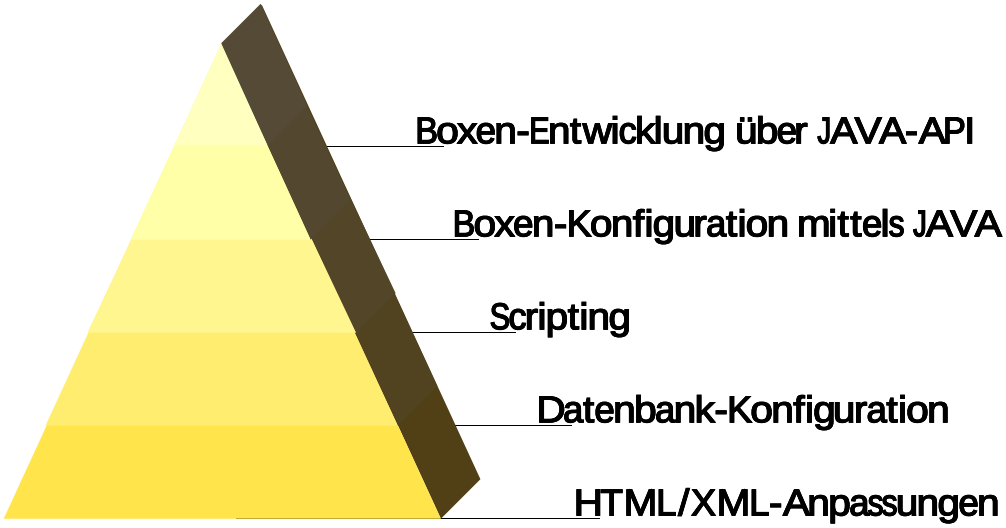
Dabei werden Multichannel- und Crossmedia-fähige Lösungen ebenso ermöglicht, wie gemischte Lösungs-Formen mit Web-Frontend und Rich-Clients.

Stets das Optimum zur Optimierung Ihrer Abläufe, Ihrer Kommunikation.

Für die Ersteller von Boxen stehen eine Reihe von Java-Klassenbibliotheken für diverse, immer wiederkehrende Aufgaben zur Verfügung:

- Logging, Fehlerbehandlung
- Umgang mit Währungen, Mehrsprachigkeit
- Objekt-Relational Mapping
- Security
- Kryptographie
- Messaging, Mail
- XML-Processing
- etc.

6 Know-How-Bedarf



In 80% der Projekte werden HTML-/XML-Templates angepasst und in geringem Umfang Datenbank-Anpassungen via BOXALINO-Frontend (ohne SQL-Kenntnisse) vorgenommen.

In komplexeren Aufgabenstellungen kommt Server-seitiges Scripting dazu, um komplexere Berechnungen, Abläufe etc. ohne JAVA-Programmierung zu realisieren.

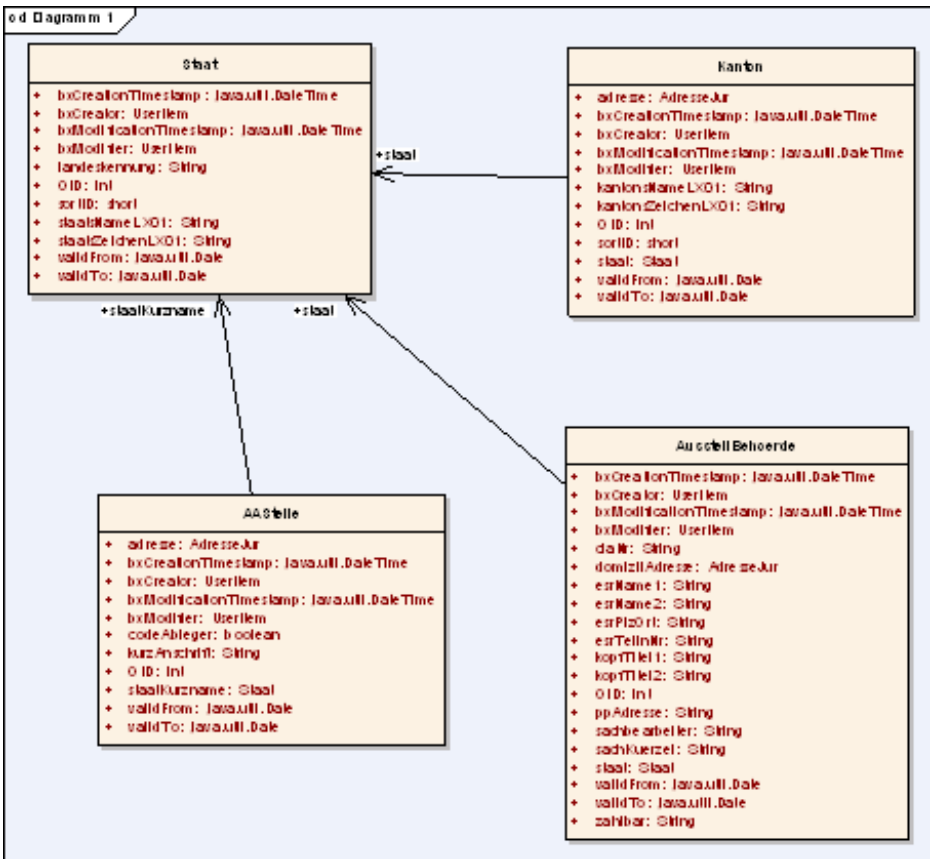
Mittels JAVA können diverse Boxen noch individueller konfiguriert werden und last but not least kann das System um beliebige eigene Boxen ergänzt werden.

7 Lösungsaufbau

Lösungen werden häufig im 3-tier Modell aufgebaut:

- Präsentation
- Fachliche Logik
- Datenschicht (Persistenz)

Die fachliche Logik wird in der Regel objekt-orientiert mit dem sogenannten BOM – Business Object Model – beschrieben.



Dann profitiert man davon, dass BOXALINO automatisch für die Persistenz sorgt.

Die Präsentation wird häufig mittels (X)HTML bereitgestellt – moderne Anwendungen bieten sowohl Rich-Clients, wie Browser-basierte Nutzungsoberflächen. Für beides bietet BOXALINO entsprechenden Support an.

BOXALINO bietet dazu einen WYSIWYG-Editor, um Masken schnell und sicher realisieren zu können.

8 Wichtige Begriffe im BOXALINO Umfeld

8.1 3-tier/n-tier

BOXALINO kann in verschiedenen Konfigurationen betrieben werden. Üblich ist eine Konfiguration mit einem Datenbankserver, einem davon getrennten BOXALINO-Server und einer ebenfalls davon getrennten Präsentation.

Dies betrifft die Prozess-Architektur. Alle Prozesse können auf einem Rechner laufen oder auch auf verteilten.

Somit lässt sich BOXALINO auf einem Notebook installieren, wie auch auf einem Cluster.

8.2 ASP – ActiveServerPages

Mittels Schnittstellen (ActiveX/XML) können BOXALINO Funktionen in ASP-Seiten verwendet werden. Voraussetzung dafür ist, dass eine Java-Laufzeitumgebung auf dem Server vorhanden ist, respektive installiert werden kann, auf dem die BOXALINO-Komponenten ablaufen. Dies kann, muss aber nicht der gleiche Server sein, auf dem auch ASP läuft.

8.3 Authentisierung

Siehe Security

8.4 Autorisierung

Die Autorisierung in BOXALINO reicht von einfachen Berechtigungen, bis hin zu projekt-/aufgabenspezifischen Rollen-basierten Systemen.

8.5 Betriebssicherheit

Der BOXALINO-Server verfügt über verschiedene Überwachungsmöglickeiten. Zum Beispiel kontrollieren Hintergrund-Threads die Verbindung zur Datenbank, via Log-Files werden Ereignisse verschiedener Wichtigkeit aufgezeichnet etc. Bei entsprechend wichtigen Ereignissen erfolgt zusätzlich eine Benachrichtigung via ausgewählten Kommunikationsmittel (eMail, SMS, SNMP ...).

BOXALINO beinhaltet einen einfachen Schutz vor Manipulationen, welcher bei Bedarf ausgedehnt werden kann und Manipulationen von innen und/oder aussen zu erkennen hilft.

Fail-Over Betrieb ist in entsprechenden Konfigurationen mit BOXALINO möglich.

8.6 BOM – Business Object Model

Eine leistungsfähige Möglichkeit das Business Object Model zu beschreiben und mittels Java-Objekten mit Funktionen zu implementieren, bildet die Basis für viele Anwendungen mit BOXALINO.

8.7 Caches

Siehe Performance.

8.8 Client-Server

Siehe 3-tier/n-tier.

8.9 Datenbanken

Es kann via BOXALINO gleichzeitig auf mehrere Datenbanken zugegriffen werden. Hauptsächlich kommt dabei die Standard Java Datenbank Schnittstelle JDBC zum Einsatz.

8.10 Deployment

Für das Deployment von Boxen stehen spezielle Mechaniken zur Verfügung. BOXALINO unterstützt war und ear-Files.

8.11 Dialogabläufe

Vorgänge, welche zur Erfüllung eines Arbeitsschrittes durch eine Person an einem Arbeitsplatz ohne Unterbruch ablaufen, werden durch Dialogabläufe und/oder einfache Dialoge unterstützt.

Siehe auch Workflow.

8.12 EJB

Der BOXALINO-Server kann in EJB-Umgebungen eingesetzt werden. Er kann EJBs nutzen und/oder Dienste zur Verfügung stellen.

8.13 Fail-Over

Siehe Betriebsicherheit.

8.14 Fehlerbehandlung

BOXALINO bietet komponenten-bezogenes Exception-Handling und spart auch bei der Entwicklung individueller Erweiterungen dank einem durchdachten Framework diesbezüglich viel Arbeit.

8.15 Firewall

BOXALINO kommuniziert via HTTP(S) und kann so in den meisten Firewall-Konfigurationen eingesetzt werden. In speziellen Fällen kann BOXALINO mit individuellen Port-Konfigurationen der System-Architektur angepasst betrieben werden.

8.16 Frames (HTML-Frames)

(X)HTML-Frames und iFrames können beliebig eingesetzt werden.

8.17 GUI-Builder

BOXALINO Design-Time Tool zur WYSIWYG-Erstellung von Masken.

8.18 HTTP

Das Hypertext Transfer Protocol ist die Basis für die Kommunikation im Internet.

8.19 HTTP-Tunneling

Unter HTTP-Tunneling versteht man das Verpacken von Kommunikations-requests zwischen Client und Server in HTTP-Requests, so dass die Requests wie normale HTTP-Requests aussehen.

Die Kommunikation des BOXALINO-Servers mit dem BOXALINO-Client und eventuellen anderen Applikationen erfolgt üblicherweise via HTTP(S).

Es können auch Appplets sein, welche in eine HTML-Seite eingebettet sind und mit dem BOXALINO-Server kommunizieren.

Via HTTP können auch eigene Protokolle 'getunneled' werden. Dazu wird im BOXALINO-Server ein eigenes Servlet bereitgestellt, um diesen Kanal zu bedienen.

8.20 Integration

Mittels eigener Boxen, welche als Schnittstellen zu bestehenden Systemen dienen, kann eine homogene Integration erreicht werden.

Offene Schnittstellen – synchron, wie asynchron – helfen, diverse Integrations-Varianten effizient nutzen zu können.

8.21 Java

Der BOXALINO-Server ist vollständig in Java realisiert. Auf der Client-Seite ist Java nicht nötig, kann aber eingesetzt werden.

8.22 JSP – JavaServerPages

In einer JSP-Umgebung können BOXALINO Klassen/Objekte eingesetzt werden und somit die Funktionen von BOXALINO genutzt werden.

BOXALINO braucht selbst keine JSP, sondern verwendet zur HTML-Generierung eine eigene Technik, welche den Vorteil hat, dass im HTLM keine neuen Tags eingeführt werden und im HTML nicht programmiert wird. In bestehenden HTML-Tags werden neue Attribute eingesetzt und diese steuern die Darstellung (Wiederholungen, bedingte Darstellung, Zugriff auf Werte, Auslösen von Kommandi etc.).

In diesem Sinne ist die HTML-Erweiterung von BOXALINO eher als Erweiterung der Markup-Language zu verstehen, als eine Programmiersprache.

8.23 Kanäle

BOXALINO kann mehrere Kanäle bedienen und die meisten Boxen trennen den Inhalt strikt vom Kanal.

Zugriff ist via HTML, WML, XML möglich, ebenso applikatorisch z.B. via CORBA, RMI, DCOM usw.

8.24 Logging und Benachrichtigungen/Notifikationen

Der BOXALINO-Server schreibt vier Log-Dateien:

- eine für den Betreiber des Systems (hoster)
- eine für den Web-Designer (designer)
- eine für den Besitzer des Systems (owner)
- eine für Techniker (system)

Dabei wird zwischen verschiedenen Severities unterschieden, wobei bei der Stufe Alarm Benachrichtigungen via eMail eingesetzt werden.

Bei sensitiven Installationen können Alarms auch via anderen Kanälen abgesetzt werden (SMS, serielle/parallele Schnittstellen etc.).

8.25 Mandantenfähigkeit

BOXALINO-Systeme können mandantenfähig eingesetzt werden.

8.26 Metamodell

Das BOXALINO-System verfügt über ein leistungsfähiges Metamodell. Es kann zur Realisierung eigener Boxen genutzt werden.

8.27 Middleware

BOXALINO kann als eine Art Middleware eingesetzt werden und via Schnittstellen-Boxen auf andere Middleware zugreifen.

8.28 Multithreading

Der BOXALINO-Server ist multithreaded aufgebaut. Zur Unterstützung stehen Testclients zur Verfügung, welche viele, aktive Anwender simulieren.

Damit die schwierige und fehleranfällige Aufgabe der Realisierung von multithreaded Server-Funktionen etwas vereinfacht und standardisiert wird, stehen von BOXALINO Vorgaben zur Verfügung.

8.29 MVC – Model-View-Controller Muster

BOXALINO ist nach einem angepassten MVC-Muster aufgebaut. Der Model-Teil ist stark vom VC-Teil getrennt. Einige Optimierungen zugunsten der Performance wurden im VC-Teil vorgenommen.

Bei der Realisierung eigener Boxen kann ein vollständig eigener Model-Teil verwendet werden oder aber einer, welcher auf BOXALINO Java Klassenbibliotheken aufbaut.

8.30 Performance

Der BOXALINO-Server verfügt über diverse Cache-Mechanismen zur Steigerung der Performance. Die wichtigsten sind Caches im Bereich der Datenbank (Objekte, Abfrageresultate), der dynamischen HTML-Seiten (eine Art Vorcompilierung) und Objektpools.

Der BOXALINO-Server kann zudem parallel auf mehreren Prozessoren betrieben werden. Es ist ein Load-Balancing möglich.

Für Portale stehen zusätzlich Cache-Möglichkeiten zur Verfügung – bis hin zu einem verteilbaren, adaptiven Cache für extrem hohe Zugriffszahlen.

8.31 Schutz vor Manipulation

Im Gegensatz zu Applikationen unterliegen Web-Applikationen umfangreichen Manipulations-Möglichkeiten. Sicherheitsmechanismen und Design-Vorgaben helfen, solche Manipulationen a) zu erkennen und b) zurückzuverfolgen.

Basismechanismen sind in BOXALINO integriert. Weitergehende Möglichkeiten stehen projekt-/aufgabenbezogen zur Verfügung.

8.32 Security

Authentisierung, Autorisierung, Verschlüsselung von Transport und Speicherung – gezielte Rechtevergabe – nur einige Stichworte zu den verfügbaren Möglichkeiten beim Einsatz von BOXALINO.

8.33 Session-Management

BOXALINO verfügt über eine eigene Abstraktion der Session, damit nicht Servlet-spezifische und andere Details einfliessen. Dies erlaubt die Bedienung von Sessions via verschiedenen Kanälen.

Ebenso können online-Kanalwechsel und persistente Sessions unterstützt werden.

8.34 Skalierbarkeit

BOXALINO kann auf verschiedene Arten getuned werden, so dass auch ausserordentlich hohe Belastungen und Spitzen abgefangen werden können.

Dazu werden zusätzliche Caches, Parallelisierungen und nötigenfalls speziell optimierte Programmwege eingesetzt.

8.35 Workflow

Vorgänge, welche von einer/mehreren Person(en) verteilt über mehrere Zeiteinheiten und/oder verteilt über Arbeitsplätze gehen und in entsprechende Arbeitsschritte unterteilt sind (diese können mit oder ohne EDV-Unterstützung gestaltet sein).

BOXALINO stellt dazu eine Workflow-Box zur Verfügung.

Siehe auch Dialogabläufe.