

Jakarta Turbine
Ein Open Source Framework
für Webanwendungen

KNF Kongreß 2001

Henning P. Schmiedehausen

<henning@apache.org>

Turbine - ein Überblick

- Open Source unter Apache License
- 100% pure Java, Java 2 (JDK 1.2+)
- Servlet-basiertes Framework
- Servlet API 2.1 kompatibel
- auf allen gängigen Plattformen
(Unix, Linux, Windows, MacOS) lauffähig
- Entwicklungsumgebung (TDK) verfügbar
- Aktuelle Version Turbine 2.1, 2.2 in Kürze geplant
- Turbine 3 derzeit in Entwicklung (Alpha Release)

Eigenschaften

- Kein „application server“!
- Framework für web-basierten Applikationen
- Servlet-Container notwendig
(z.B. Tomcat, Resin, JBoss, iPlanet Webserver)
- Enge Integration mit anderen Jakarta Projekten
(Velocity, Jetspeed usw.)
- „Baukasten“ mit immer wieder benötigten Werkzeugen
und Modulen
- Umfang ca. 440 Klassen

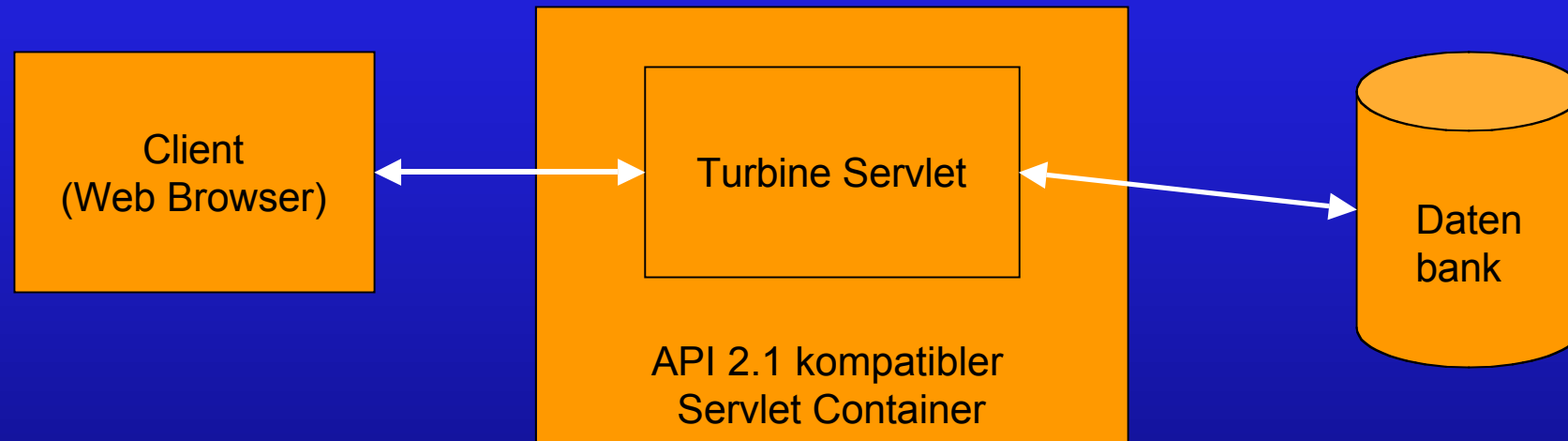
Was ist in Turbine enthalten?

- Verwaltung und Steuerung von User-Sessions
- Parameter Parsing für GET/POST/PATH_INFO
- Automatische, XML basierte Validierung von HTML Formularen (Intake Service)
- Templateverwaltung, Velocity, WebMacro, FreeMarker und JSP (Velocity wird empfohlen)
- Object Mapping zur Datenmodellierung in XML
- Datenbank Routinen und Hilfsklassen (z.B. ID Broker)
- Factory und Caching Services für Java Objekte

Was ist in Turbine enthalten?

- Web-Services (z.B. File-Upload)
- ACL-basiertes Sicherheitssystem mit Benutzern, Gruppen, Rollen und Berechtigungen
- Integration mit JavaMail zum Versenden von E-Mail
- XML-RPC Integration (SOAP in Turbine 3)
- Javabasierter Scheduler (Cron)
- Logging, Lokalisierung, Java Naming Services
- und noch viel mehr...

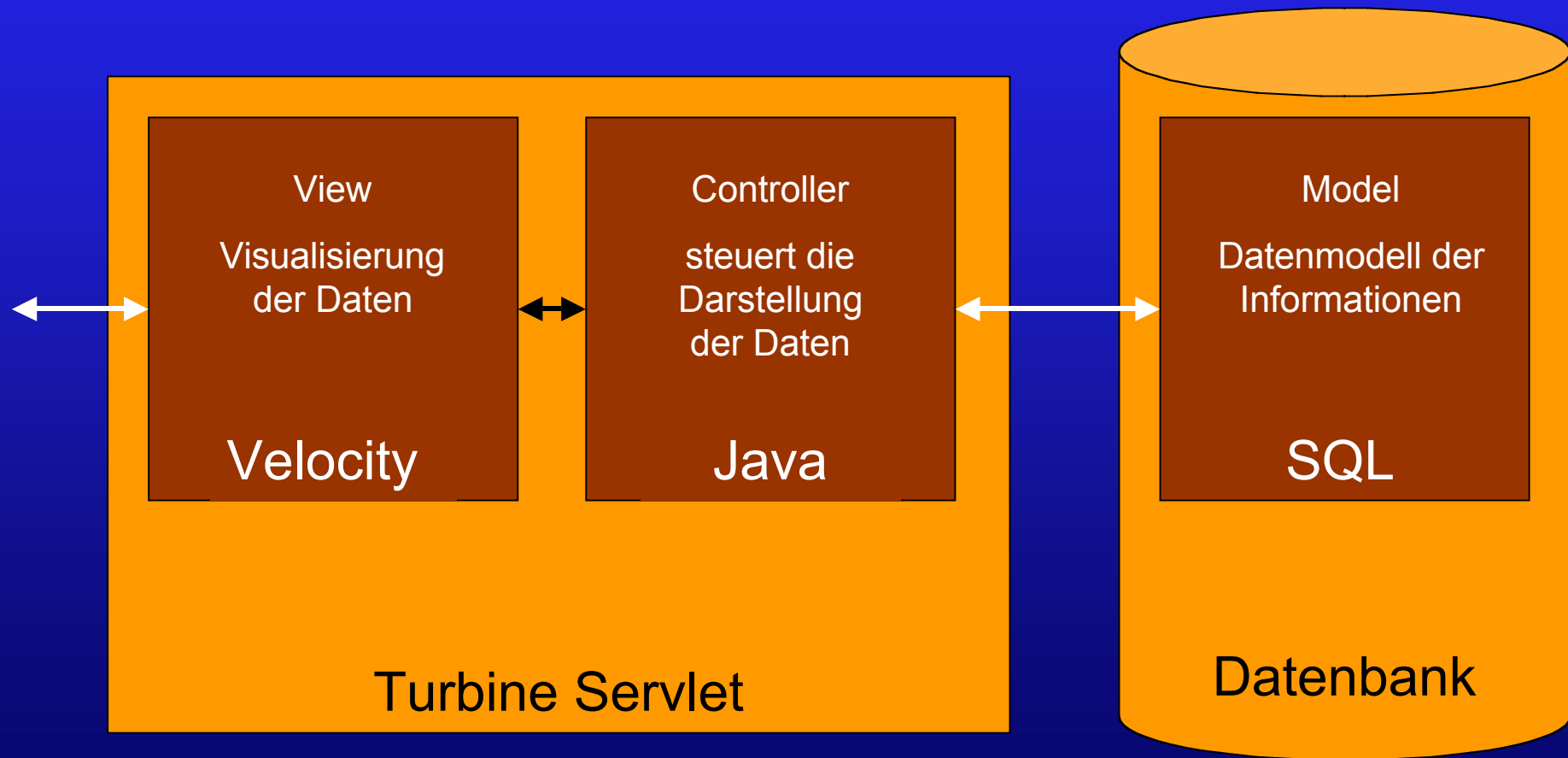
Three Tier Architektur



HTML/XML
über HTTP

JDBC

MVC Architektur



Datenbank Zugriffe

- Torque dient zur automatischen Erzeugung von Klassen für SQL-Tabellen und Datensätze
- Aus einem XML Template werden datenbankunabhängige Java Klassen und spezifischer SQL Code erzeugt
- Pools zur Verwaltung von JDBC Verbindungen ermöglichen effektive Datenbankzugriffe
- Ein ID-Broker unterstützt die ID Verwaltung
- Unterstützt DB/2, Hypersonic, Interbase, MySQL, MS SQL, Oracle, PostgreSQL, SyBase, SAPDB usw.

Datenbank Zugriffe

- Tabellen werden durch Peer Objekte repräsentiert
- Einzelne Zeilen werden durch Datenobjekte dargestellt
- Zugriff auf Datenfelder erfolgt mit Bean-Methoden (getXXX/setXXX)
- SQL SELECT Statements werden auf Kriterienobjekte abgebildet
- Mapping von logischen auf tatsächliche Datenbestände erfolgt mit Hilfe einer Konfigurationsdatei

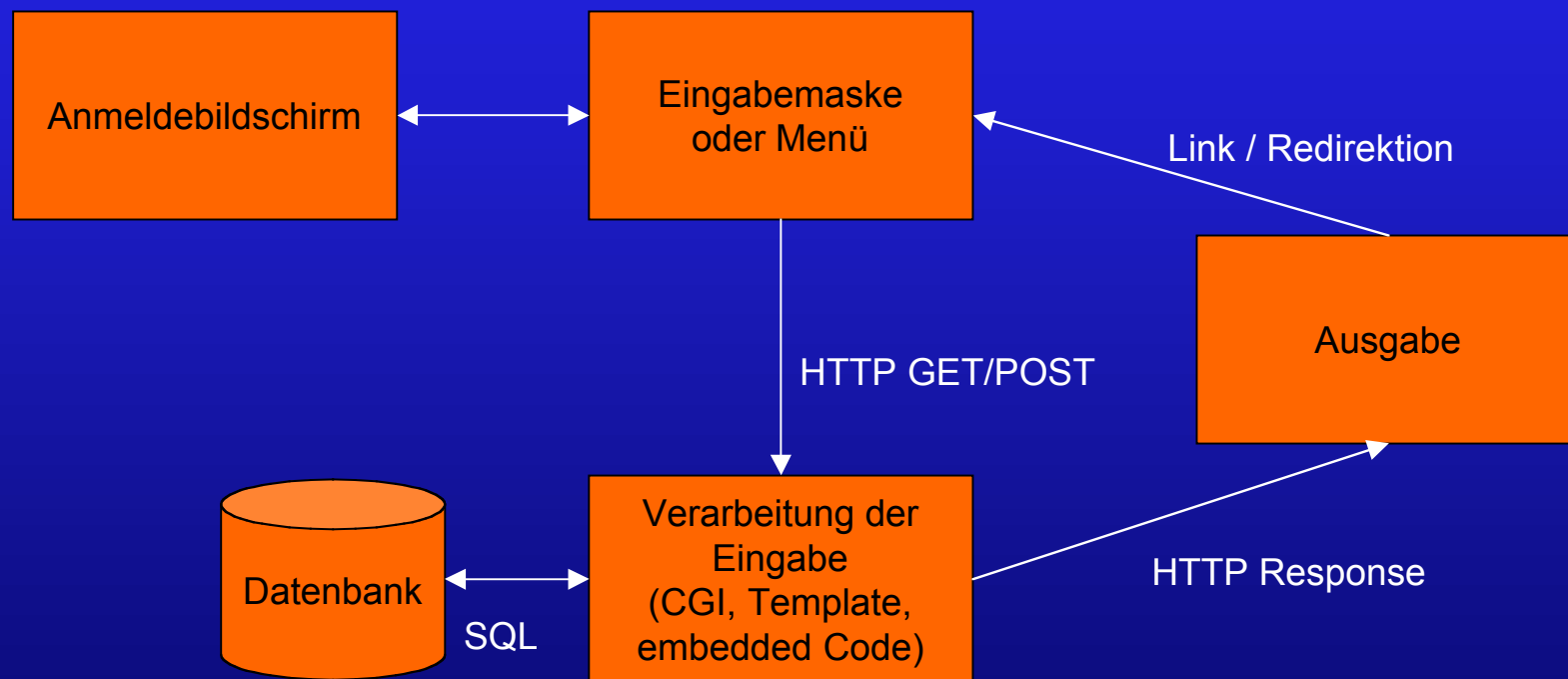
Velocity

- Open Source unter Apache License
- 100% pure Java
- Velocity erzeugt Dokumente aus Templates
- enthält einfache Kontrollstrukturen
- Zugriffsmöglichkeiten auf Java Objekte
- Oft benötigte Programmteile können in Macros und Macrobibliotheken ausgelagert werden
- für jede Art von Text verwendbar, nicht nur HTML / XML

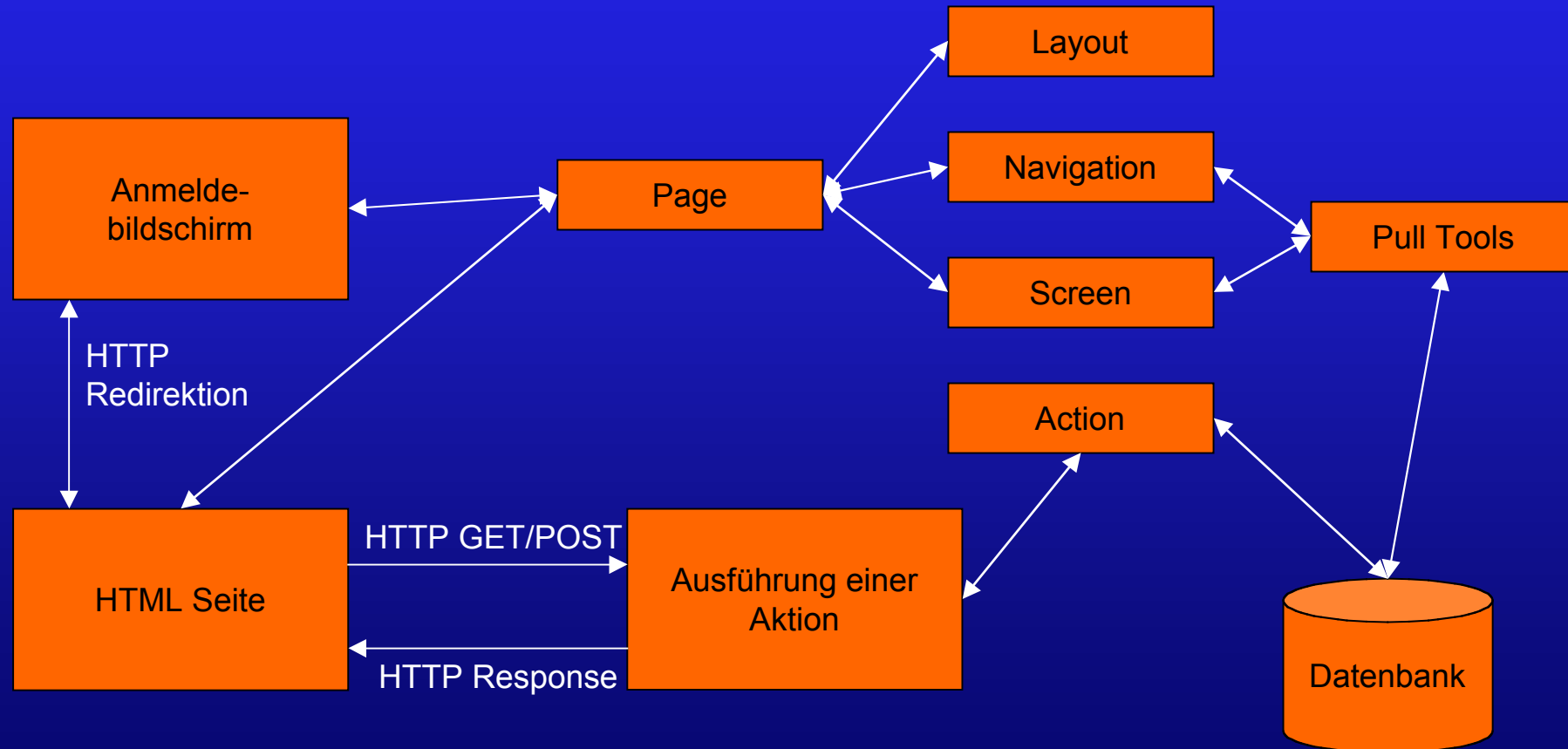
Velocity

- Kommunikation zwischen Java Code und Template erfolgt durch ein Kontext Objekt
 - Java: Hashtabelle mit Objektreferenzen
 - Template: Referenzen über \$variable Schreibweise erreichbar
- Direkter Support für Beanmethoden (getXXX/isXXX)
- unterstützt Iterator() und Array Objekte für Schleifen
- Durch Pull-Tools einfach erweiterbar
- Pull-Tools sind normale Kontext-Objekte
- Turbine unterstützt Pull-Tools in verschiedenen Lebenszeiten (Global, Session, Request, User)

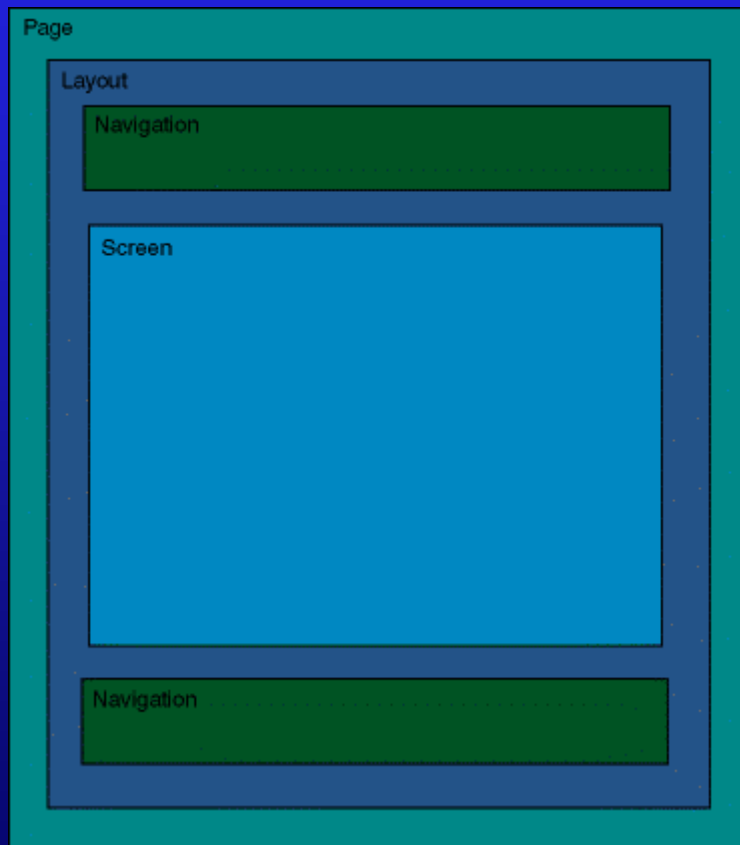
„klassische“ Webapplikation



Turbine Applikation



Aufbau einer Webseite



- Page wird durch eine (mitgelieferte) Java Klasse erzeugt.
- Layout, Navigation und Screen sind Templates
- Layout, Navigation und Screen können durch Klassen ergänzt werden
- Turbine ordnet Templates und Klassen automatisch zu

Demo „Suche Längen-/Breitengrad“

- Einfache, webbasierte Anwendung
- Eingabe und Auflisten von Daten
- Suche nach Daten (Ortsname)
- Validierung aller Eingaben mit Fehlerausgabe
- Speicherung der Daten in SQL Datenbank

Programmumfang

- Im TDK entwickelt, ca. 2h Arbeit
- zwei action-Klassen
- eine screen-Klasse
- drei Pull Tools
- fünf weitere, automatisch erzeugte Klassen
- Vier Screen-Templates
- Vier Templates für Navigation und Layout
- 150 Zeilen Java Code, 100 Zeilen Template Code

Beispiel „Daten auflisten“

```
/*
 * Screen Klasse zum Erzeugen der Koordinaten-Liste
 */

package de.intermeta.coord.modules.screens;

import org.apache.velocity.context.Context;
import org.apache.turbine.util.RunData;

import org.apache.turbine.modules.screens.VelocityScreen;
import org.apache.turbine.util.db.Criteria;

import de.intermeta.coord.om.CoordDataPeer;

public class List extends VelocityScreen
{
    public void doBuildTemplate(RunData data, Context ctx) throws Exception
    {
        Criteria c = new Criteria();

        ctx.put("list", CoordDataPeer.doSelect(c));
    }
}
```

```
$page.setTitle("Koordinaten auflisten")

<H1>Koordinaten auflisten</H1>

<TABLE BORDER="1">
<TR>
  <TH WIDTH="50%">Ort</TH>
  <TH WIDTH="25%">Breitengrad</TH>
  <TH WIDTH="25%">L&auml;ngengrad</TH>
</TR>
#foreach ( $row in $list )
<TR>
  <TD>$row.Ort</TD>
  <TD>$lat.display($row.Lat)</TD>
  <TD>$lng.display($row.Lng)</TD>
</TR>
#end
</TABLE>
```

List.vm (Screen Template)

List.java (zugehörige Java Klasse)

Fazit

- Turbine und Velocity dienen zur schnellen Entwicklung von Webentwicklungen
- Durch Templates und Tools können Weboberflächen einfach gestaltet werden
- Einheitlicher Code kann für verschiedene Datenbanksysteme verwendet werden
- Der Baukasten ermöglicht es, wiederkehrenden Code nicht ständig neu zu erfinden

Links

Turbine: <http://jakarta.apache.org/turbine/>

Velocity: <http://jakarta.apache.org/velocity/>

Demo-Applikation:

<ftp://ftp.hometree.net/pub/kongress2001/turbine/coord.tar.gz>

Fragen?

Vielen Dank für Ihre
Aufmerksamkeit