

Plattformunabhängigkeit

So macht's VEDA

Thomas Osterloh

Bereichsleiter
Anwendungsentwicklung

VEDA



10 Jahre AS/point - Schloss Zweibrücken
23.09.2005

Plattformunabhängigkeit So macht's VEDA

- ▶ **Entscheidungsfindung**
- ▶ **Architektur**
- ▶ **Entwicklung**
- ▶ **Migration**
- ▶ **Ausblick**
- ▶ **Zusammenfassung**



Plattformunabhängigkeit So macht's VEDA

- ▶ **Entscheidungsfindung**
- ▶ Architektur
- ▶ Entwicklung
- ▶ Migration
- ▶ Ausblick
- ▶ Zusammenfassung



Entscheidungsfindung

- ▶ **Java oder .NET ?**
- ▶ **VEDA ist IBM-Partner**
- ▶ **IBM setzt auf Java**
- ▶ **SAP setzt auf Java**
- ▶ **Alle setzen auf Java außer Microsoft**
- ▶ **Auf der iSeries läuft Java, nicht .NET**



Entscheidungsfindung

- ▶ **Kaufen oder selbst machen ?**
- ▶ **13 Anbieter analysiert (12 Monate)**
 - ▶ **Keine zufrieden stellende Lösung gefunden**



Gewichtete Entscheidungs-Kriterien

- ▶ **Lizenz-Kosten / Externe Kosten**
- ▶ **Runtime-Kosten / Kosten pro Kunde**
- ▶ **Entwicklungs-Aufwand**
- ▶ **Verfügbarkeits-Zeitpunkt**
- ▶ **Hersteller-Abhängigkeit**
- ▶ **Entwicklungs-Tempo**
- ▶ **KnowHow-Zukauf**
- ▶ **Neu / Migration**
- ▶ **One Source**
- ▶ **User-Interface**
- ▶ **Technischer Ansatz**
- ▶ **Java / Sprache**



Entscheidung

- ▶ **VEDA-eigene Migration**
- ▶ **Vorteile**
 - ▶ **Hersteller-Unabhängigkeit**
 - ▶ **Lizenz-Kosten / Externe Kosten**
 - ▶ **User Interface**
- ▶ **Nachteile**
 - ▶ **Entwicklungs-Aufwand**
 - ▶ **Entwicklungs-Tempo**



Plattformunabhängigkeit So macht's VEDA

- ▶ Entscheidungsfindung
- ▶ **Architektur**
- ▶ Entwicklung
- ▶ Migration
- ▶ Ausblick
- ▶ Zusammenfassung



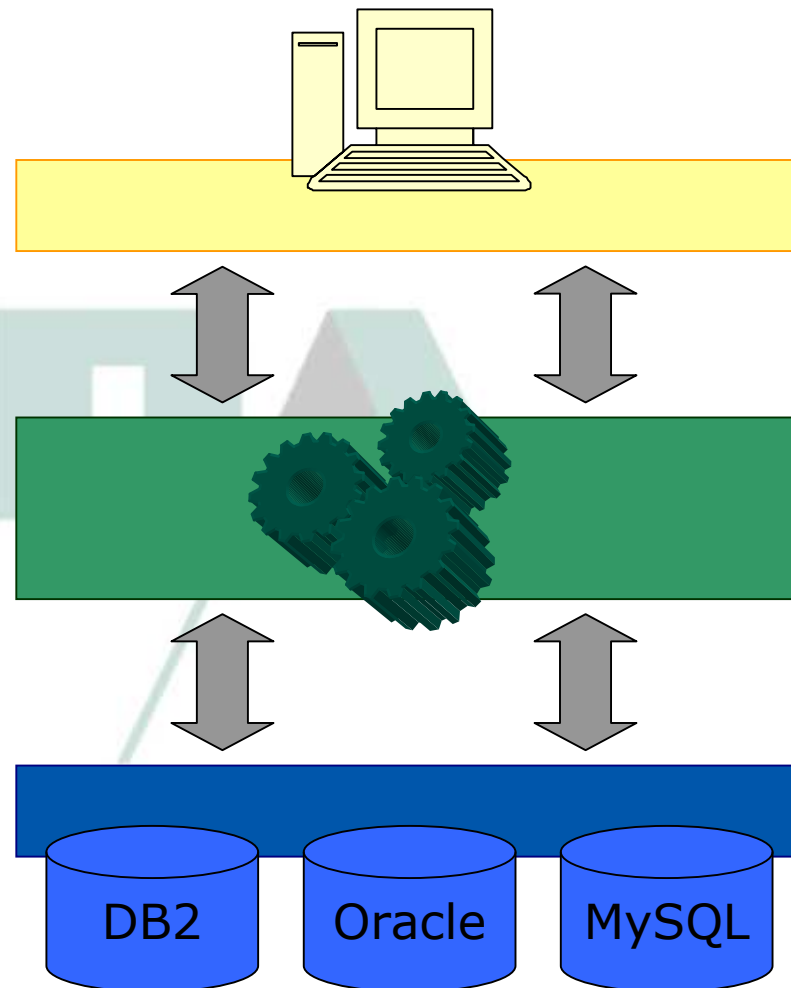
Architektur

3-Schichten-Modell

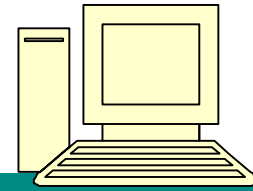
▶ **User-Interface**

▶ **Business-Logik**

▶ **Datenbank**



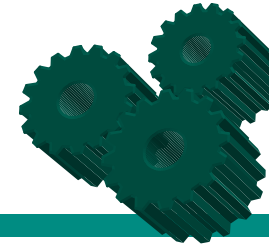
User-Interface



- ▶ **User-Interface ist austauschbar.**
- ▶ **SWING, HTML, 5250**
- ▶ **Programm weiß nicht, wie die Benutzeroberfläche dargestellt wird.**
- ▶ **Grundlage der Kommunikation zwischen Business-Logik und User-Interface und der Darstellung ist DSPF-DDS.**



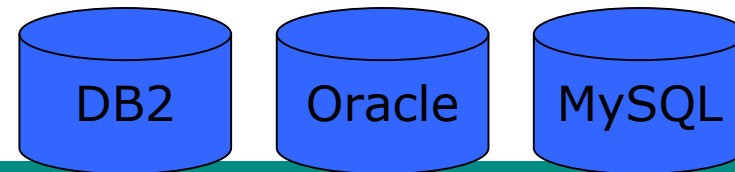
Business-Logik



- ▶ **Alle Anwendungs-Programme sind migriert und haben daher (fast) die identischen Funktionen wie auf der iSeries.**
- ▶ **System-Funktionen sind neu programmiert.**



Datenbank



- ▶ **Datenbank per Konfiguration austauschbar.**
 - ▶ **Zur Zeit MS-SQL, MySQL**
- ▶ **Nur 1 „package“ enthält Datenbank-Funktionen**
- ▶ **Nur einfache Funktionen werden benutzt. (SELECT, UPDATE, INSERT, DELETE)**



Installations-Varianten (1.)

▶ Stand-Alone

- ▶ Alles auf 1 System
- ▶ Kein Application-Server erforderlich
- ▶ Voraussetzungen:

- ▶ Java JRE 1.5
- ▶ Datenbank
- ▶ Prozessor >2 GHz
- ▶ Hauptspeicher >500 MB
- ▶ Plattenplatz

▶ Java	50 MB
▶ VEDA	60 MB
▶ Datenbank	? MB
▶ Daten	? MB



Installations-Varianten (2.)

▶ Client/Server

▶ Client ist über HTTP-Tunneling mit dem Server verbunden.

- ▶ Internet, „Port 80“
- ▶ SSL
- ▶ Komprimierung



Client

- ▶ **Java J2SE 1.5**
- ▶ **Client-Installation per Web-Start**
- ▶ **Nur für User-Interface**



Server

- ▶ **Java J2SE 1.5**
- ▶ **Anwendungs-Logik**
- ▶ **Datenbank**
- ▶ **LDAP (Lightweight Directory Access Protocol) zur Benutzer-Authentifizierung**
- ▶ **HTTP-Server**
 - ▶ z. Zt. Apache
- ▶ **Application-Server erforderlich**
 - ▶ z. Zt. Tomcat



Plattformen

- ▶ **Betriebssysteme**
 - ▶ **Zur Zeit : Windows**
 - ▶ **Später : Linux, Unix, OS/400**
- ▶ **Datenbanken**
 - ▶ **Zur Zeit : MS-SQL, MySQL**
 - ▶ **Später : DB2, Oracle**



Plattformunabhängigkeit So macht's VEDA

- ▶ Entscheidungsfindung
- ▶ Architektur
- ▶ **Entwicklung**
- ▶ Migration
- ▶ Ausblick
- ▶ Zusammenfassung



Entwicklungs-Umgebung

▶ **Eclipse 3.0.1** (www.eclipse.org)



▶ **Java JDK 1.5.0** (java.sun.com)





Package Explorer

- >Migration [vacsvr01.veda.de]
 - >Runtime [vacsvr01.veda.de]
 - >src
 - com.veda.admin
 - com.veda.dient
 - com.veda.dient.test
 - com.veda.dp
 - com.veda.dp.expression
 - com.veda.cmd
 - com.veda.cmd.guide
 - com.veda.cmd.guide.parm
 - com.veda.cmd.value
 - com.veda.data
 - com.veda.db
 - com.veda.db.test
 - com.veda.dds
 - com.veda.html
 - com.veda.java.lang
 - com.veda.java.util.regex
 - com.veda.job
 - Job.java 1.53 (ASCII -k)
 - JobFile.java 1.1 (ASCII
 - LogWindow.java 1.3 (AS
 - Quasidialog 1.4 (ASCII

```

VJPropertiesDialog.java | Job.java x | VJSwingPanel.java | VJCalendarPanel.java
  /**
  * sets the outQueue for this job.
  * @param outQueue - the new outQueue as 20A String (outq + lib).
  */
  public void setOutQueue(String outQueue) {
    this.outQueue = outQueue;
  }

  /**
  * @return the requested program-queue
  */
  public ProgramQueue getProgramQueue(String queue) {
    Program pgm = null;
    for (int i = pgmStack.size()-1; i >= 0; i--){
      pgm = pgmStack.getEntry(i).getProgram();
      if (pgm.getName().equals(queue)){
        return pgm.getProgramQueue();
      }
    }
    return null;
  }

  /**
  * @return the program-queue of the last program in the stack.
  */
  public ProgramQueue getLastProgramQueue() {
    return getProgramQueue(pgmStack.size() - 1); // letztes Element
  }
  /**

```

Outline

- getOverrides(char, String)
- getPgmConstructorStartMillis()
- getPreviousProgram()
- getProgramFromStack(String)
- getProgramInstance(String)
- getProgramQueue(int)
- getProgramQueue(String)
- netProgramStack()

Problems Console Progress Tasks

0 errors, 1 warning, 0 infos (Filter matched 1 of 22 items)

Description	Resource	In Folder	Location
The import com.veda.msg.ProgramQueue is never used	VJSwingPanelTOS...	Runtime/src/com/veda/swing	line 62

Projekte in eclipse

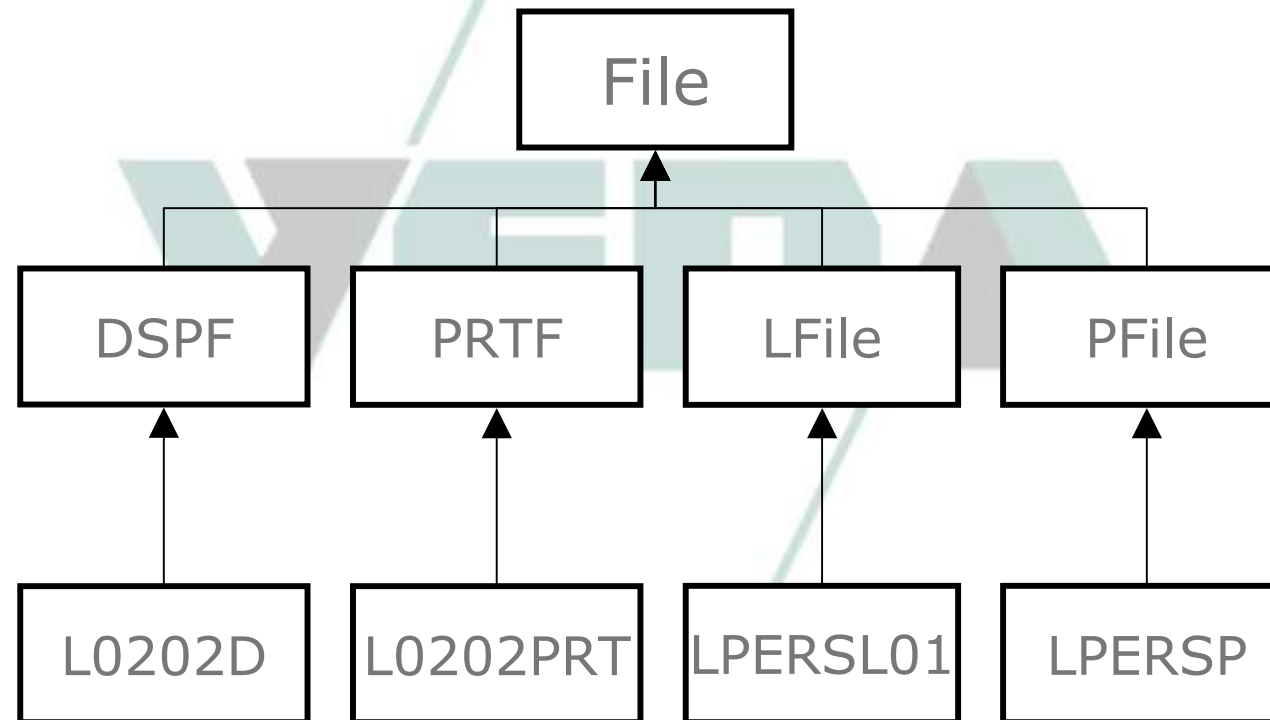
▶ Runtime

- ▶ Enthält alle Funktionen, die zur Laufzeit der Anwendung notwendig sind.
- ▶ Enthält keinen migrierten Code.
- ▶ Pakete: client, clp, cmd, db, dds, job, monitoring, msg, print, properties, rpg, server, swing, system, util



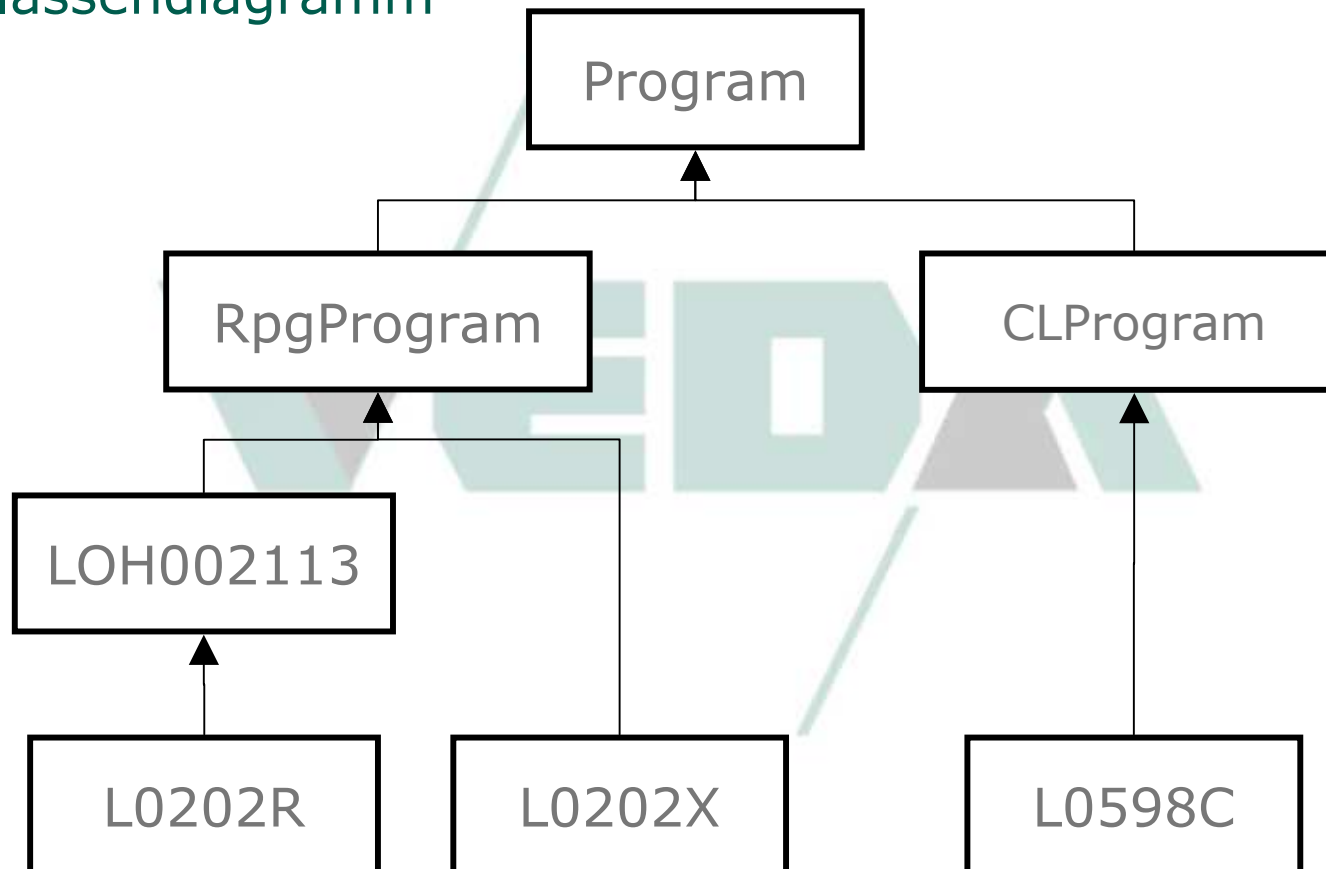
OO-Programmierung (Beispiel 1)

Klassendiagramm



OO-Programmierung (Beispiel 2)

Klassendiagramm



Projekte in eclipse

▶ System

- ▶ Enthält alle Funktionen, die OS/400 Commands betreffen.
- ▶ Pakete: clp, cmd, cpp, dds, rpg



Projekte in eclipse

▶ VEDA HR Entgelt

- ▶ LOH3020db (Datenbank)
- ▶ LOH3020ds (Datenstrukturen)
- ▶ LOH3020ui (User-Interface)
- ▶ LOH3020cmd (Commands)
- ▶ LOH3020clp (CL-Programme)
- ▶ LOH3020gen (Sets)
- ▶ LOH3020rpg (RPG-Programme)



Plattformunabhängigkeit So macht's VEDA

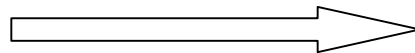
- ▶ Entscheidungsfindung
- ▶ Architektur
- ▶ Entwicklung
- ▶ **Migration**
- ▶ Ausblick
- ▶ Zusammenfassung



Migration

- ▶ **Alle** auf der iSeries vorhandenen Quellen werden migriert.
- ▶ **1 komplette Anwendung mit 1 Aufruf**
- ▶ **Beispiel VEDA HR Entgelt**
 - ▶ **> 1.000.000 RPG-Zeilen**
 - ▶ **ca. 8 Stunden Migrationszeit**

RPG
CL
DDS



Migration

- 
- ▶ **CVTPUMSRC (PUMA-Quellen umsetzen)**
 - ▶ **Feldreferenzdatei**
 - ▶ **Physische Dateien**
 - ▶ **Logische Dateien (Join-Dateien)**
 - ▶ **Bildschirmdateien**
 - ▶ **Druckerdateien**
 - ▶ **CL-Programme**
 - ▶ **RPG-Programme (RPG III)**
 - ▶ **Befehle**

Migration

- ▶ **CVTPUMOBJ (PUMA-Objekte umsetzen)**
 - ▶ ***DTAARA Datenbereiche**
 - ▶ ***USRPRF Benutzerprofile**
 - ▶ ***JOBBD Job-Beschreibungen**



Projekte in eclipse

▶ Migration

- ▶ Enthält alle Migrations-Funktionen.

- ▶ Paket: migration

- ▶ Klassen:

- ▶ ARATransformer (DTAARA)

- ▶ CLPTransformer (CL-PGM)

- ▶ CMDTransformer (CMD)

- ▶ CPPTransformer (Cmd-Processing-Pgm)

- ▶ DDSTransformer (DDS)

- ▶ PRFTransformer (USRPRF)

- ▶ RPGTransformer (RPG-PGM)




Laufzeit-Umgebung

▶ Objekt-Typen

- ▶ *LIB Bibliotheken
- ▶ *JOBQ Job-Beschreibungen
- ▶ *JOBQ Job-Warteschlangen
- ▶ *MSGQ Nachrichten-Warteschlangen
- ▶ *OUTQ Ausgabe-Warteschlangen
- ▶ *USRPRF Benutzerprofile
- ▶ *DTAARA Datenbereiche
- ▶ *USRSPC User-Space



Laufzeit-Umgebung

- 
- ▶ **Realisierte Befehle**
 - ▶ **CALL, SBMJOB, ENDJOB, RCLRSC, SIGNOFF**
 - ▶ **DBG STR-, END-**
 - ▶ **DTAARA CRT-, CHG-, RTV-, DLT-**
 - ▶ **DSPJOBLOG, WRKACTJOB, WRKSBMJOB, WRKUSRJOB**
 - ▶ **JOB DSP-, CHG-, HLD-, RLS-, RTV-A**
 - ▶ **JOB D CRT-, DSP-, CHG-, DLT-, WRK-**
 - ▶ **JOBQ CRT-, DLT-, HLD-, RLS-, WRK-, CLR-**
 - ▶ **LIB CRT-, DSP-, DLT-**
 - ▶ **LIBL DSP-, EDT-, CHG-**
 - ▶ **LIBLE ADD-, RMV-**
 - ▶ **MSGQ CRT-, CHG-, DLT-**
 - ▶ **MSG RCV-, SNDPGM-, RMV-**
 - ▶ **OBJ CHK-, CRTDUP-, RNM-**
 - ▶ **OUTQ CRT-, CHG-, DLT-, HLD-, RLS-, WRK-, CLR-**
 - ▶ **OVRDBF, DLTOVR, OVRPRTF, OPNQRYF, CLOF**
 - ▶ **SPLF WRK-, DLT-, DSP-, HLD-, RLS-**
 - ▶ **USRPRF CRT-, DSP-, CHG-, DLT-, RTV-**
 - ▶ **USRSPC CRT-, DLT-, CHG-**

Plattformunabhängigkeit So macht's VEDA

- ▶ Entscheidungsfindung
- ▶ Architektur
- ▶ Entwicklung
- ▶ Migration
- ▶ **Ausblick**
- ▶ Zusammenfassung



Ausblick

- ▶ **Fertigstellung aller VEDA-Anwendungen**
- ▶ **Weitere Betriebssysteme**
- ▶ **Weitere Datenbanken**



Plattformunabhängigkeit So macht's VEDA

- ▶ Entscheidungsfindung
- ▶ Architektur
- ▶ Entwicklung
- ▶ Migration
- ▶ Ausblick
- ▶ **Zusammenfassung**

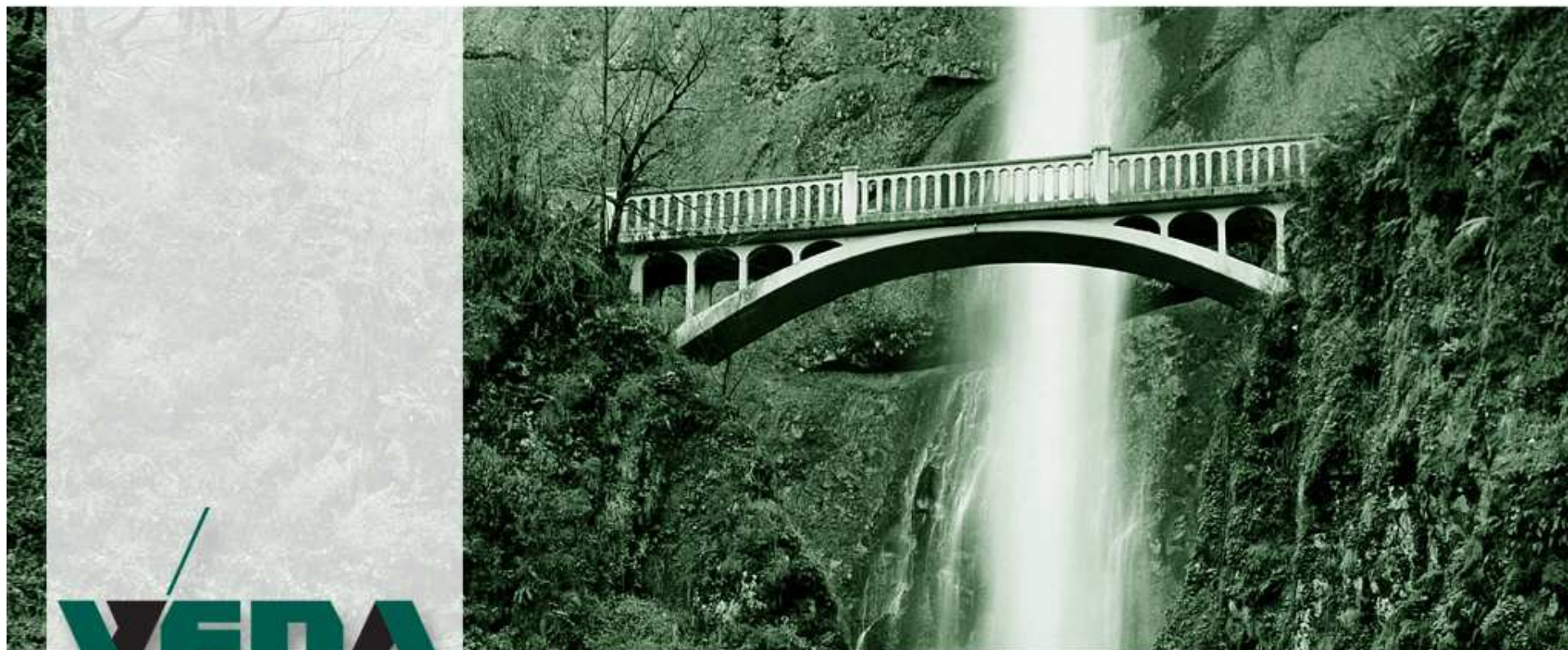


Zusammenfassung

- ▶ **VEDA-Software „j-ware“ ist**
 - ▶ **Plattformunabhängig**
 - ▶ **100 % pure Java**
 - ▶ **3-Schichten-Modell**
 - ▶ **Anwendungslogik komplett migriert**



Danke!



VEDA