Hype – Ein Java-Werkzeug für die schnelle Realisierung von webbasierten Dokumentenmanagement-Anwendungen

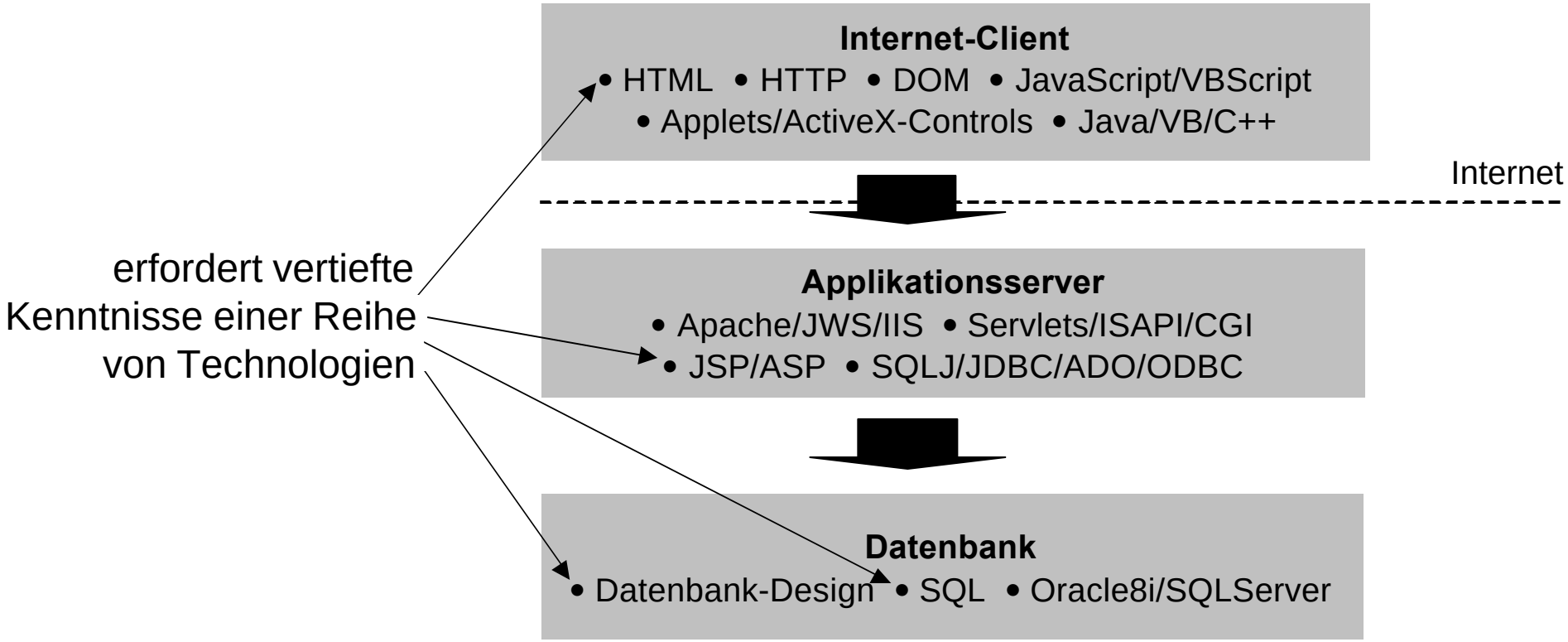
Enno Scholz **Rainer Lischetzki**

DaimlerChrysler AG, Forschung und Technologie 3
Alt-Moabit 96a, 10559 Berlin
{enno.scholz,rainer.lischetzki}@daimlerchrysler.com

Definition: Webbasierte Dokumentenmanagement-Anwendung

- Zentraler Server für elektronische Dokumente:
  - in Standardformaten:
    •Emails •Office-Dokumente •Multimedia-Dateien
  - **in anwendungsspezifischen Formaten**:
    •"Konstruktionsgruppe" •"Schadensmeldung" •"Qualitätsbericht"
- Zugriff über das WWW:
  - Standard-Webbrowser genügt (keine Installation von Client-Software)
  - nicht nur Ansehen, sondern auch **Erstellen, Löschen, Bearbeiten und Verbinden von Dokumenten geschieht direkt im Browser**

Realisierung webbasierter Dokumentenmanagement-Anwendungen als 3-Schichten-Architektur



Ideen von Hype

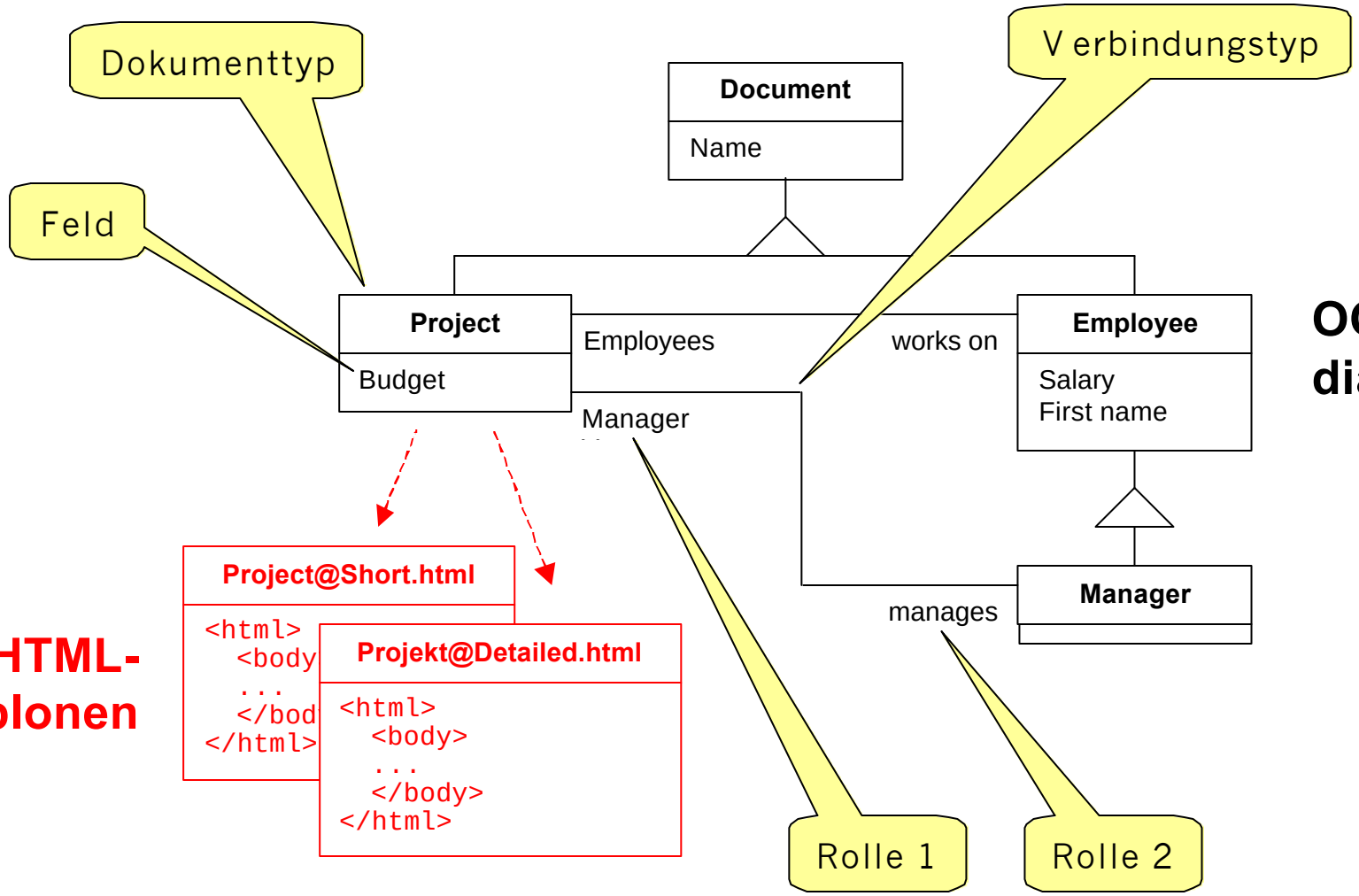
- 90% der Anwendungserstellung durch computererfahrene Nichtprogrammierer (sogenannte **Power-User**)
- 10% der Anwendungserstellung durch Java-Programmierung
- weitgehende **Generierung** einer 3-Schicht-Anwendung aus:
  - **objektorientiertem Klassendiagramm**
  - **klassenspezifischen HTML-Schablonen**
- Minimierung des Programmieranteils durch Verwendung von **Platzhaltern** in HTML-Schablonen
- Platzhalter koexistieren friedlich mit **WYSIWYG**-Editoren für HTML
- maximale Nutzung von **Typisierung** ermöglicht statische Überprüfungen:
  - getypte Platzhalter
  - serverseitige Programmierung
    - in einer getypten Sprache (Java statt z.B. JavaScript) und
    - **mit getypten persistenten Objekten** (statt ungetypt z.B. mit JDBC)

Überblick über den weiteren Vortrag

- Überblick
- Platzhalter in Hype
- Die letzten 10%: Einbettung von Java-Code
- Status der Implementierung, Vergleich, Zusammenfassung

Überblick

Hype-Anwendung = OO-Klassendiagramm + HTML-Schablonen



OO-Klassen-diagramm

HTML-Schablonen

Dokumenttyp-spezifische Platzhalter

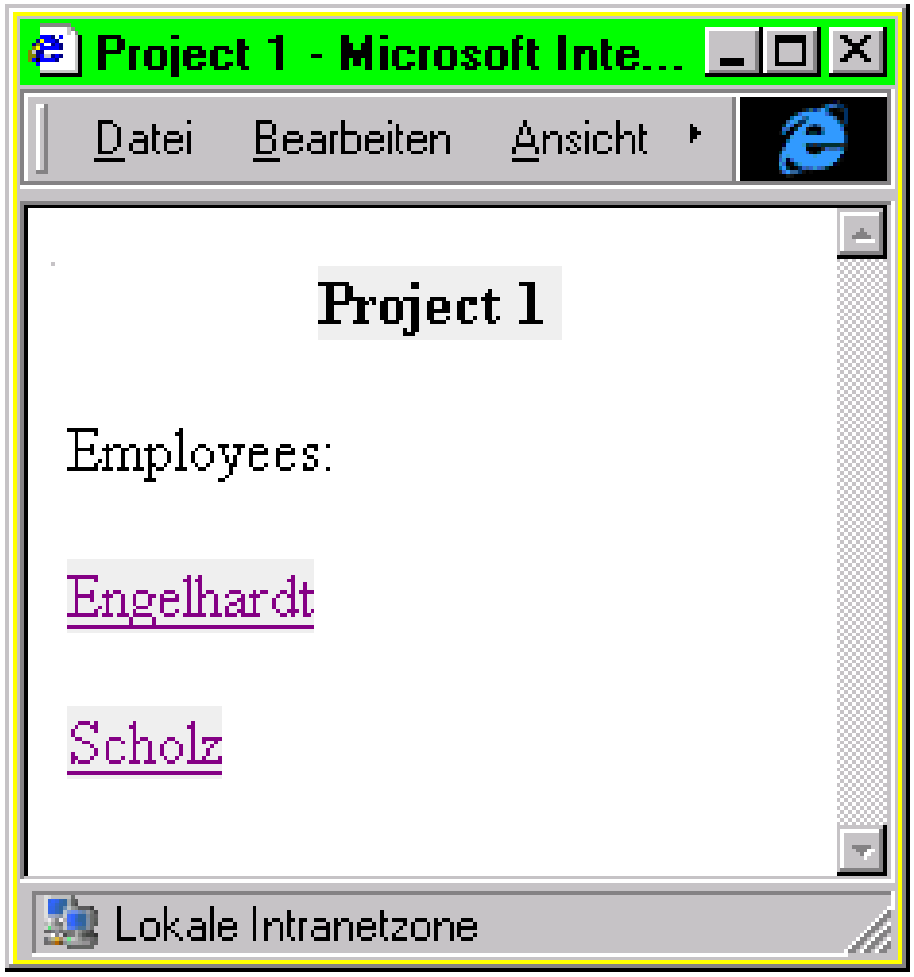
Project

Name: Project 1
Budget: 200 000
Employees: Engelhardt, Scholz
Manager: Mankowsky

```
<html>
  <body>
    <p align="center">
      <b>[Name]</b>
    </p>
    <p>Employees:</p>
    <p>
      <a href="Employee@Short.html">
        [Employees]
      </a>
    </p>
  </body>
</html>
```

Platzhalter für Feld

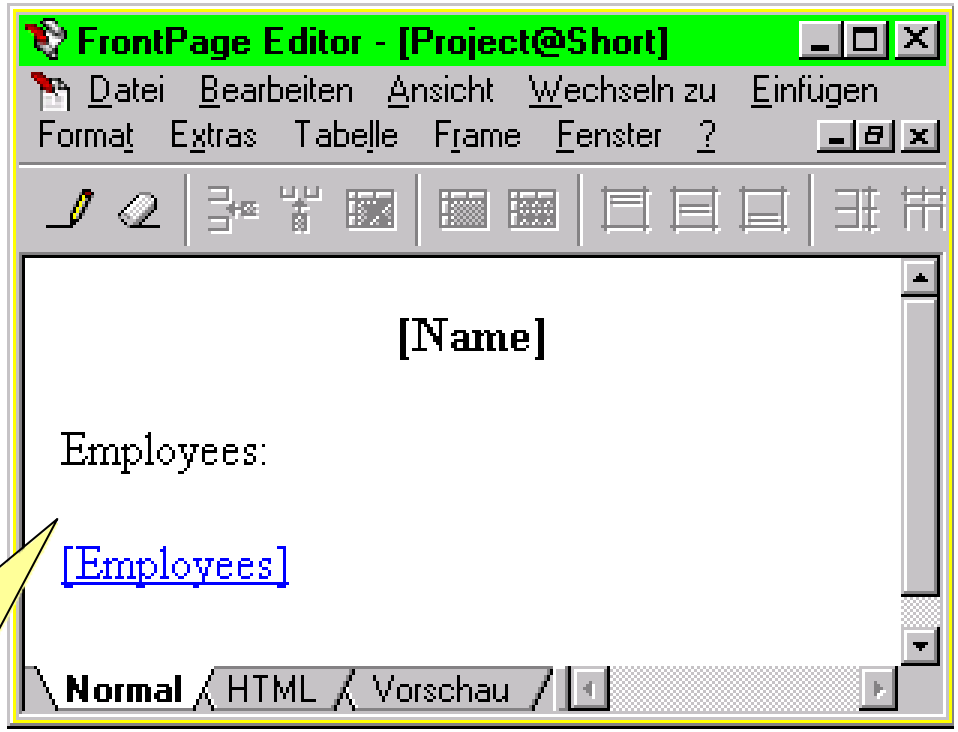
Platzhalter für Rolle



Platzhalter sind "WYSIWYG-tauglich"

"Rohes" HTML

```
<html>
 <body>
  <p align="center">
   <b>[Name]</b>
  </p>
  <p>Employees:</p>
  <p>
    <a href="Employee@Short.html">
    [Employees]
    </a>
  </p>
 </body>
</html>
```



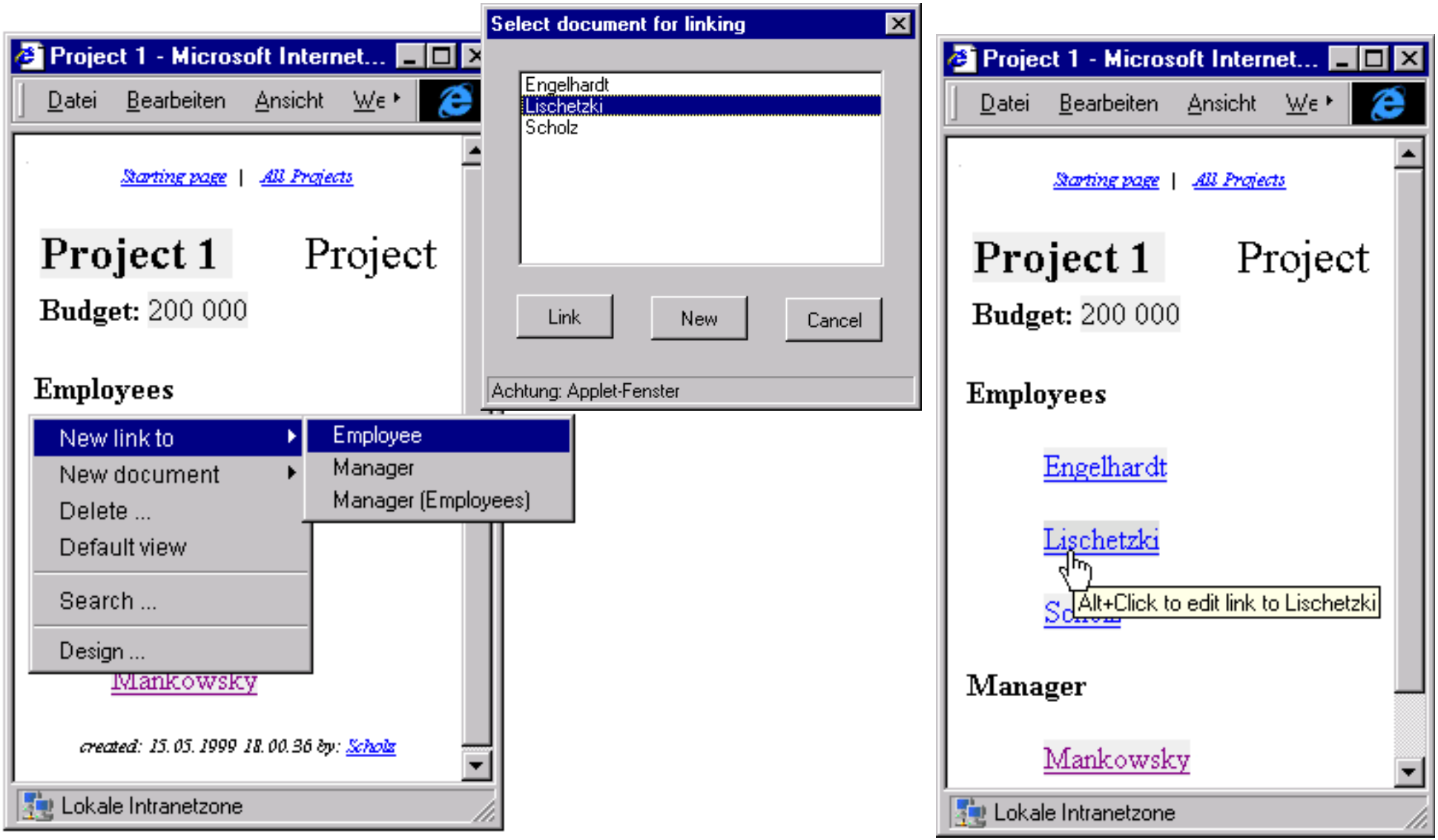
HTML im WYSIWYG-Editor

Bearbeiten von Feldern

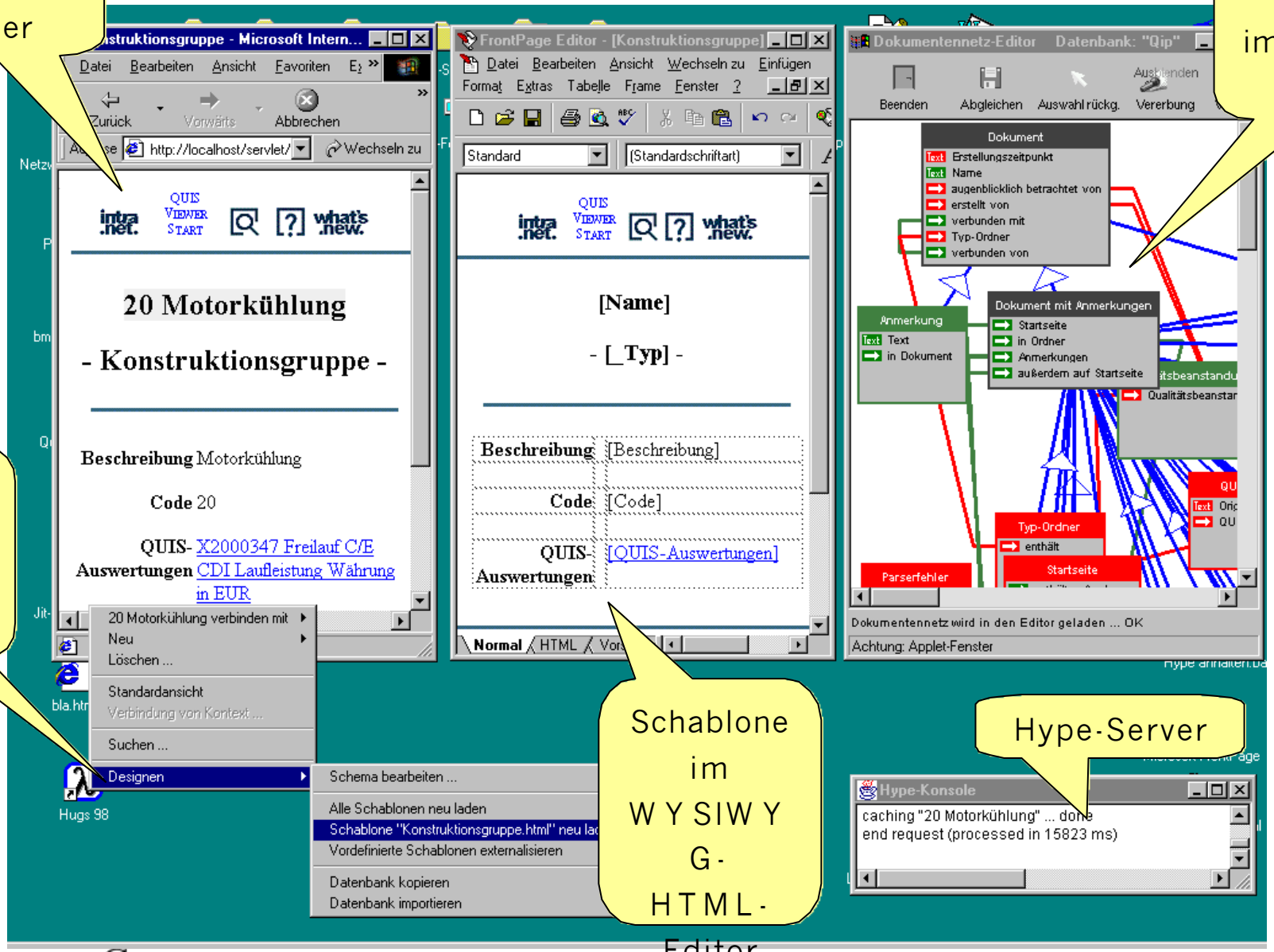




Verbinden von Dokumenten



Rapid Prototyping mit Hype



Platzhalter in Hype

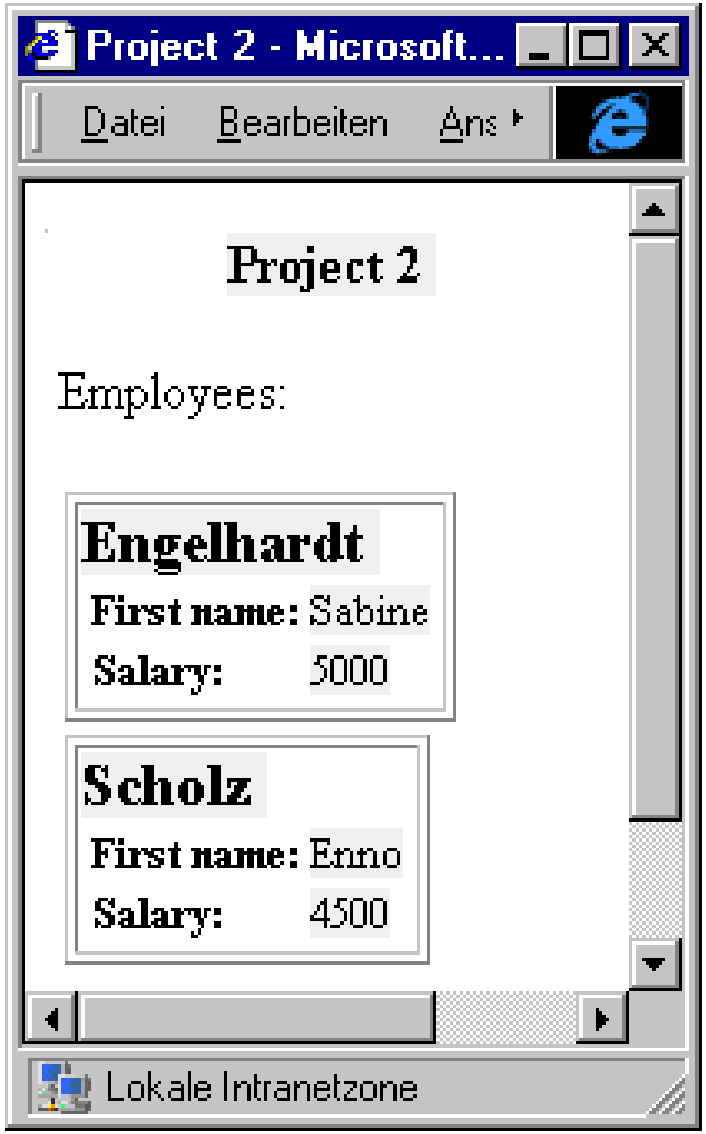
Schachtelung von Ansichten

Project

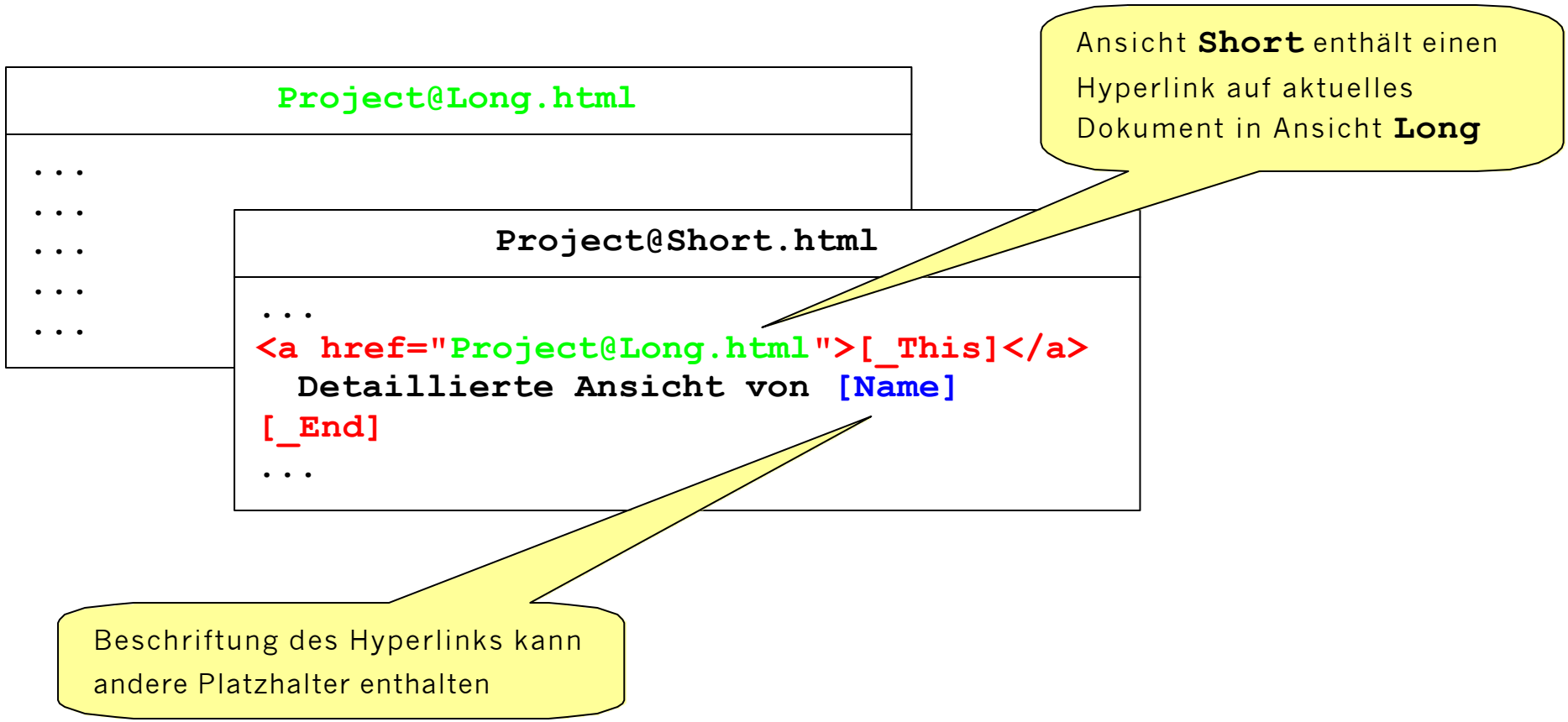
Name: Project 1
Budget: 200 000
Employees: Engelhardt, Scholz
Manager: Mankowsky

```
<html>
  <body>
    <p align="center">
      <b>[Name]</b>
    </p>
    <p>Employees:</p>
    <p>
      <a href="Employee@Short.html">
        [Employees+]
      </a>
    </p>
  </body>
</html>
```

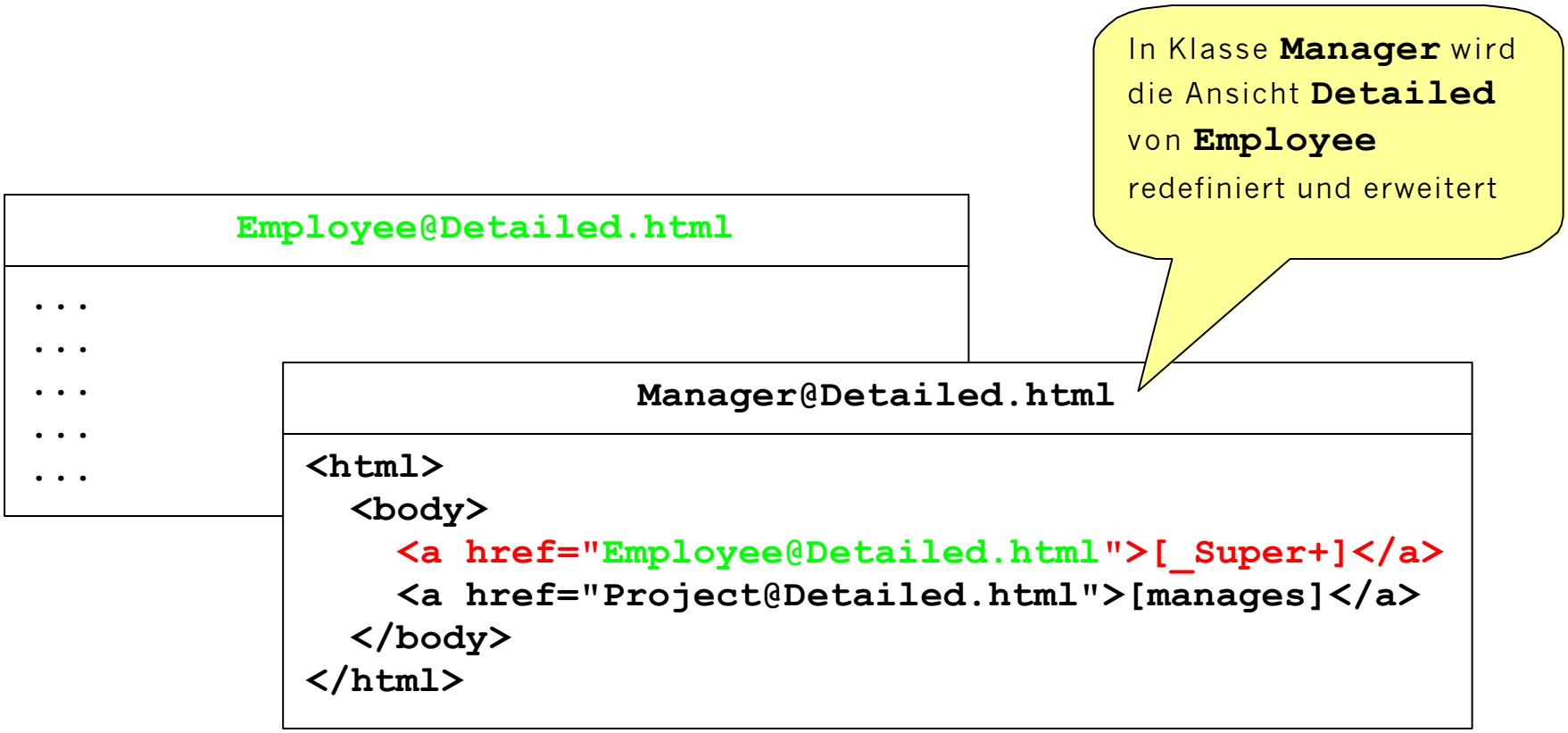
Einzige Änderung im Vergleich zu früherer Folie!



Verweise auf andere Ansichten des aktuellen Dokuments



Vererbung zwischen Ansichten

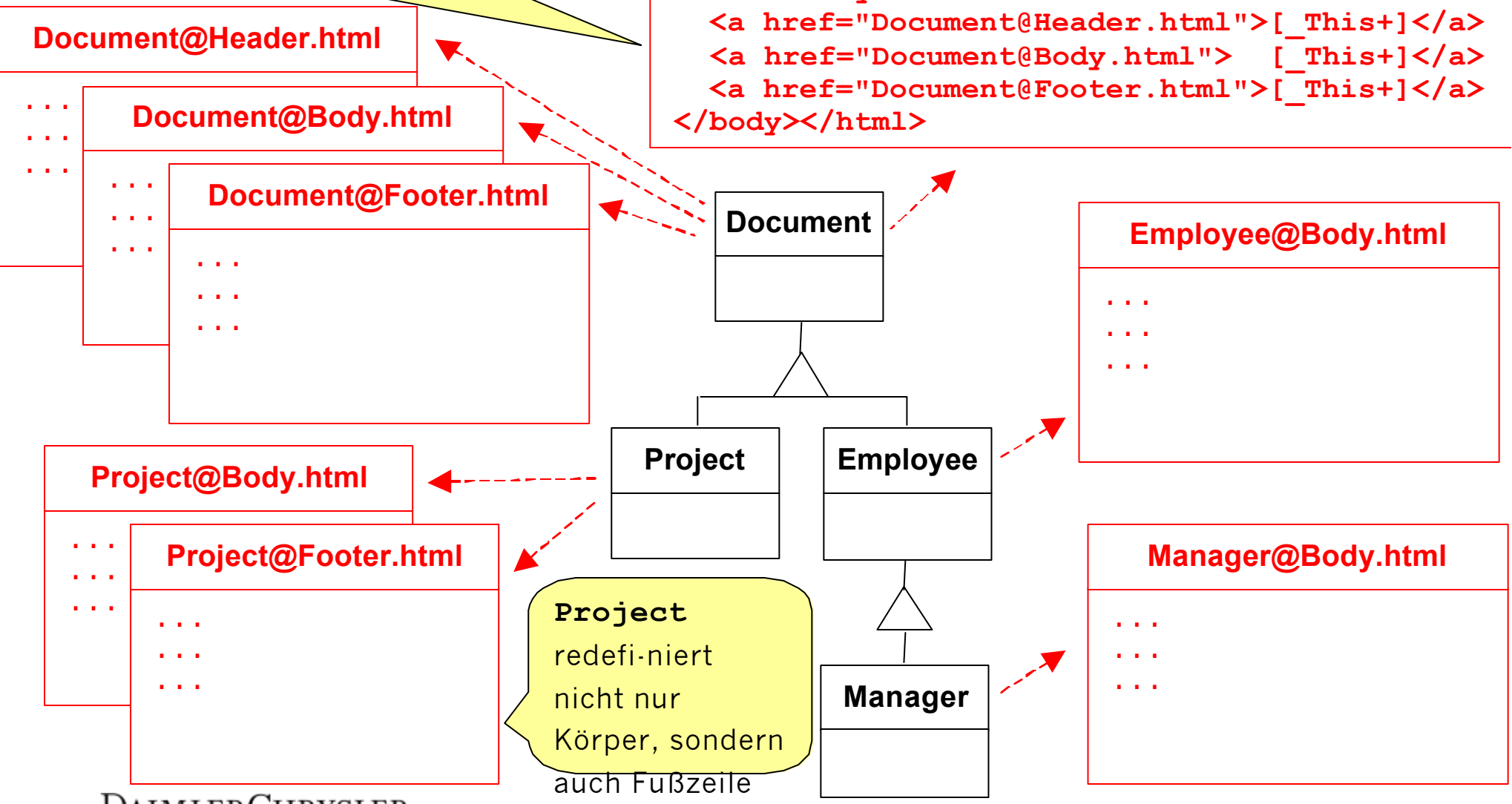


Anwendung: Einheitliche Kopf- und Fußzeile

Ansicht **CI** wird von allen Subklassen von **Document** geerbt

Document@CI.html

```
<html><body>
  <a href="Document@Header.html">[_This+]</a>
  <a href="Document@Body.html">  [_This+]</a>
  <a href="Document@Footer.html">[_This+]</a>
</body></html>
```



Vor- und Nachteile von Platzhaltern in Hype

Vorteile:

- Reduzierung des Programmieraufwandes für dynamisches HTML
- durch statische Überprüfungen können Laufzeitfehler und *dangling hyperlinks* vollständig (!) ausgeschlossen werden
- Syntax optimiert für Verwendung mit WYSIWYG-HTML-Editoren
- objektorientierte Konzepte (Vererbung, Redefinition, Aggregation) erleichtern Strukturierung komplexer Anwendungen
- statische Analyse der Schablonen ermöglicht Vorhergenerierung aller Ansichten eines Dokuments → Beschleunigung von Lesezugriffen auf Dokumente

Nachteile:

- begrenzte Mächtigkeit der Platzhaltersprache

Die letzten 10 %: Einbettung von Java-Code

Lesender Zugriff auf Objekte

Project@Detailed.html

```
<html>
  ...
  <b>
    [_Java]
    float remainingBudget = this.Budget();
    EmployeeVector employees = this.Employees();
    for(int i = 0; employees.size(); i++) {
      Employee e = employees.elementAt(i);
      remainingBudget -= e.Salary();
    }
    out.print(remainingBudget);
    [_End]
  </b>
...
</html>
```

Aktuelles Objekt vom Typ **Project**

Typsichere Lese-Methode

Aktueller HTML-Ausgabestrom vom Typ **PrintWriter**

Schreibender Zugriff auf Objekte

Project@RaiseSalaryForm.html

```
<html>
 <body>
  <p align="center">[Name]</p>
  <form action="
    [_Java]
    for(int i = 0;
        i < Employees.size();
        i++) {
      Employee e =
        Employees()
         .elementAt(i);
      e.Salary(e.Salary() + amount);
    }
    [_End]">
   <p>Raise salary by:</p>
   <p><input type="text" name="amount"></p>
   <p><input type="submit"></p>
  </form>
 </body>
</html>
```

Typsichere Schreib-Methode

Java-Objekt vom Typ **String** repräsentiert Texteingabe-feld mit Name **amount**



Ergänzung und Redefinition der Zugriffsmethoden

Project@RaiseSalaryForm.html

```
<html><body>
  <p>[Name]</p>
  <form action="[_Java]
                raiseSalaries(amount);
                [_End]">
  <p>Raise salary by:
     <input type="text" na
     <input type="submit">
  </form>
</body>
```

Einzige Änderung im V ergleich zu früherer Folie!

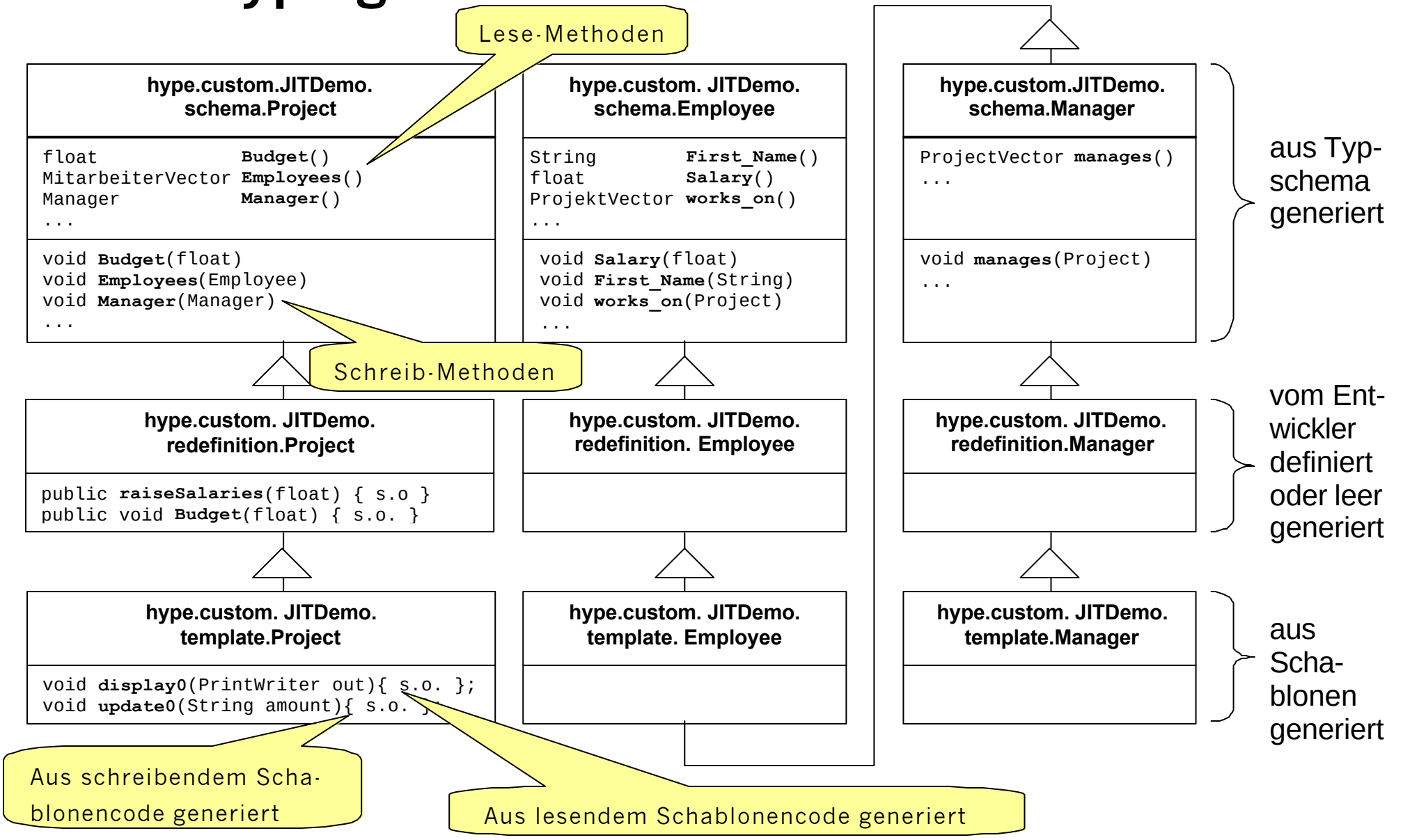
Zusätzliche Methode in Klasse **Project**

Redefinierte Methode in Klasse **Project**

Project.java

```
public class Employee extends
hype.custom.JITDemo.schema.Employee {
   void raiseSalaries(float amount) {
      ...
   };
   void Budget(float newBudget) {
      if(newBudget > this.Budget())
         super(newBudget);
   };
}
```

Von Hype generiertes Persistenz-Framework



Status der Implementierung, Vergleich, Zusammenfassung

Status der Implementierung

- > 360 Klassen Java
- JDK: 1.1.7a
- Webserver: Apache und JavaWebServer auf NT 4.0
- Datenbank: MS Access via JDBC (demnächst auch: Oracle 8i oder MS SQL Server)
- HTML-Editor: beliebig
- Browser:
  - zum Ansehen von Dokumenten: Netscape Communicator und Internet Explorer ab Version 3.0
  - zum Editieren von Dokumenten: Internet Explorer 4.0 und 5.0
- Kompromisse bei Plattformunabhängigkeit:
  - Verwendung von IE4.0-spezifischen HTML-Tags (z.B. absolut positionierte DIV-Tags),
  - *keine* Verwendung von ActiveX Components, Plugins, RMI, IE5.0-Kontextmenüs, MS-proprietären Java-APIs

Einsatz von Hype bei DaimlerChrysler

- Intranet-Hypertextsystem im Qualitätsmanagement
- stark vernetzte qualitätsrelevante Dokumente
- wächst ständig (menschliche und maschinelle Quellen)
- 45 Dokumenttypen:
  - "Konstruktionsgruppe"
  - "Schadensteil"
  - "Qualitätsbericht" ...
- ca. 1 Mio Objekte
- ermöglicht Erstellung von Qualitätsberichten direkt im Browser

ca. 100-seitige, druckreife "Multimedia-Bücher" im A4-Format

Vergleichbare Ansätze

Verbindung von HTML mit Platzhalter-/Skriptsprachen:

- Hyperwave • Java Server Pages • Active Server Pages • PHP
- DxML ...

Dokumentenmanagement-Systeme:

- Livelink • Documentum • Lotus Notes • Fabasoft Components
- BSCW • Hyperwave ...

objektorientierte Datenbankmanagementsysteme:

- Versant • Jasmine ...

Metamodellierungs-Software:

- Alfabet

Web-Frontends für relationale DBMS:

- Oracle WebDB ...

Zusammenfassung

- Werkzeug für die schnelle Entwicklung von webbasierten 3-Schicht-Anwendungen für das Dokumentenmanagement
- sowohl Benutzer als auch Anwendungsentwickler benötigen nur einen Webbrowser
- Typkonzepte erleichtern die Anwendungsentwicklung
- ermöglicht Anwendungsentwicklung nicht nur für Software-Ingenieure, sondern auch für Power-User
- Datenbankschema und HTML-Darstellung von Objekten kann im laufenden Betrieb geändert werden