

***Die Eclipse Rich Client Plattform - eine
alternative Client-Technologie für
Business Process Management Systeme***

cenit

***Java Forum
Stuttgart, Juli 2006***

Dirk Günther

Teammanager R&D e-business

d.guenther@cenit.de

Agenda

- **Kurzvorstellung CENIT**
- **Einführung BPM**
- **Überblick eWorks**
- **Warum Eclipse RCP**
- **Warum Spring Framework**
- **Tips und Tricks**
- **Diskussion**



Agenda

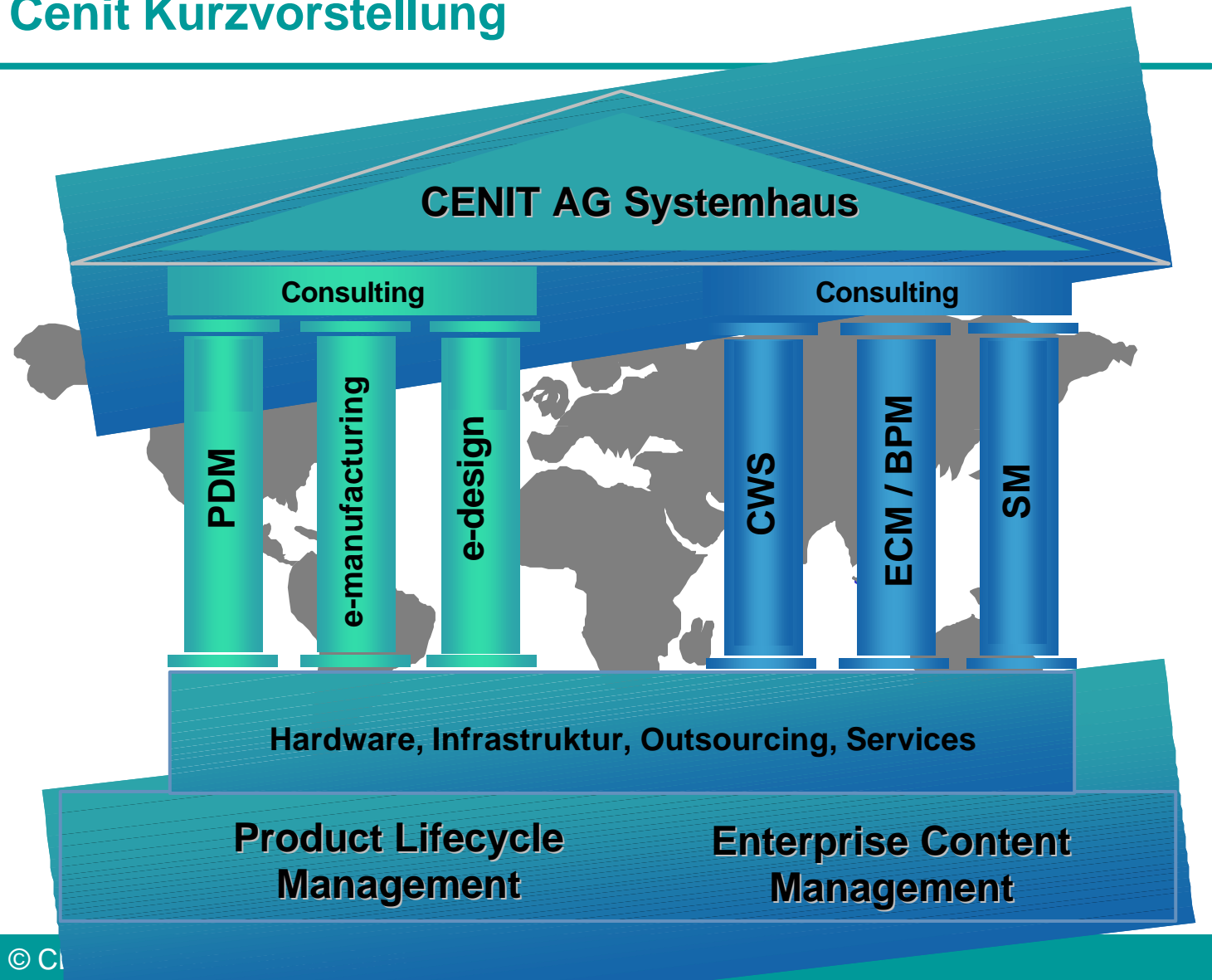
- **Kurzvorstellung CENIT**
- Einführung BPM
- Überblick eWorks
- Warum Eclipse RCP
- Warum Spring Framework
- Tips und Tricks
- Diskussion



Cenit Kurzzvorstellung

- **Gründung 1988 – CENIT GmbH**
 - **Umwandlung in AG 1998 - CENIT AG Systemhaus**
 - **Börsengang: 06.05.1998, Neuer Markt Frankfurt**
 - **Aktien: 4.183.879 (ISIN: DE0005407100, Prime Standard)**
 - **Hauptsitz: Stuttgart**
 - **Niederlassungen: Berlin, Düsseldorf, Frankfurt, Hamburg, Hannover, München, Saarbrücken**
- **Tochterunternehmen**
 - **CENIT (Schweiz) AG**
 - **CENIT North Amercia Inc.**
- **Mitarbeiter/innen: 522 konzernweit**

Cenit Kurzvorstellung



Agenda

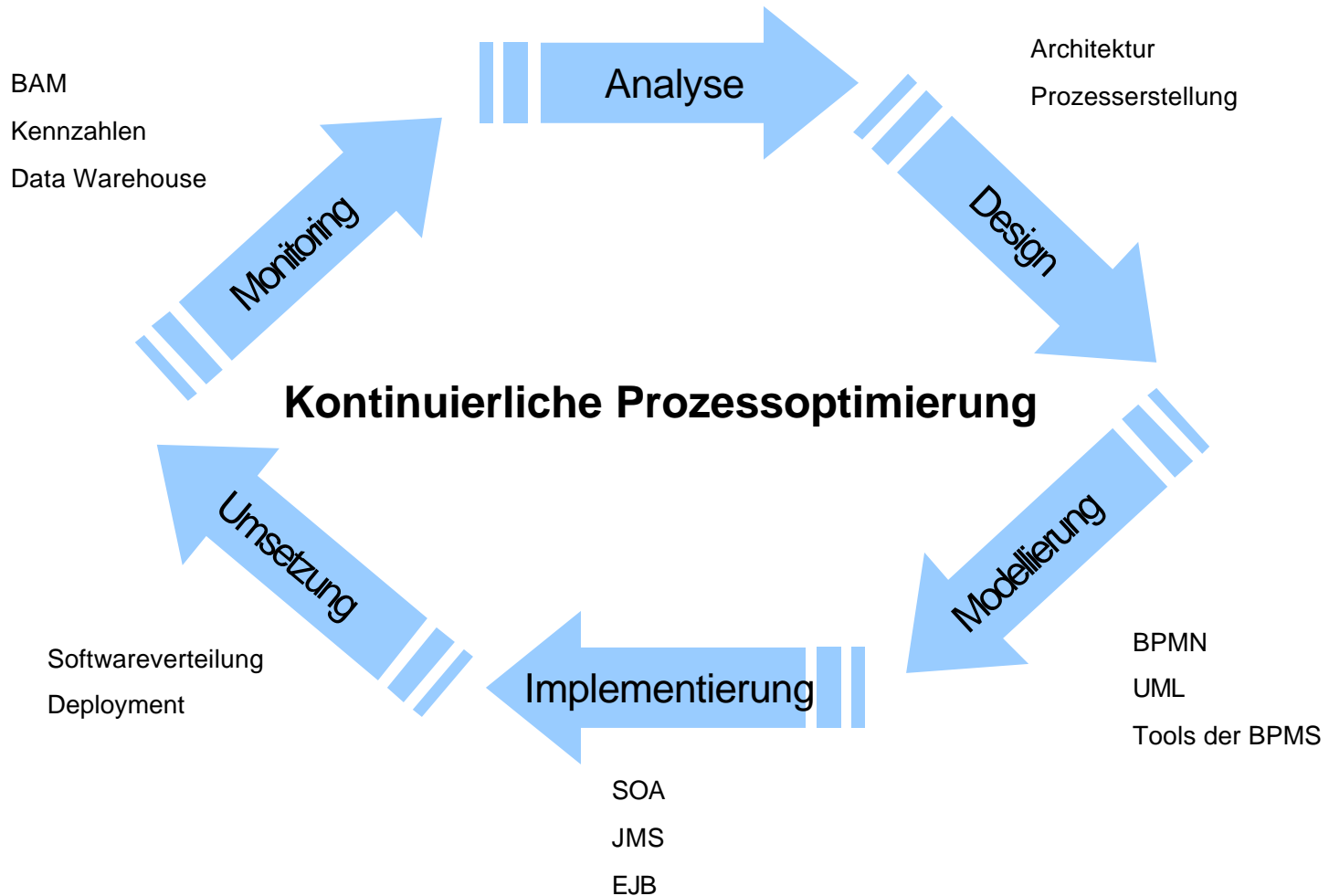
- Kurzvorstellung CENIT
- **Einführung BPM**
- Überblick eWorks
- Warum Eclipse RCP
- Warum Spring Framework
- Tips und Tricks
- Diskussion



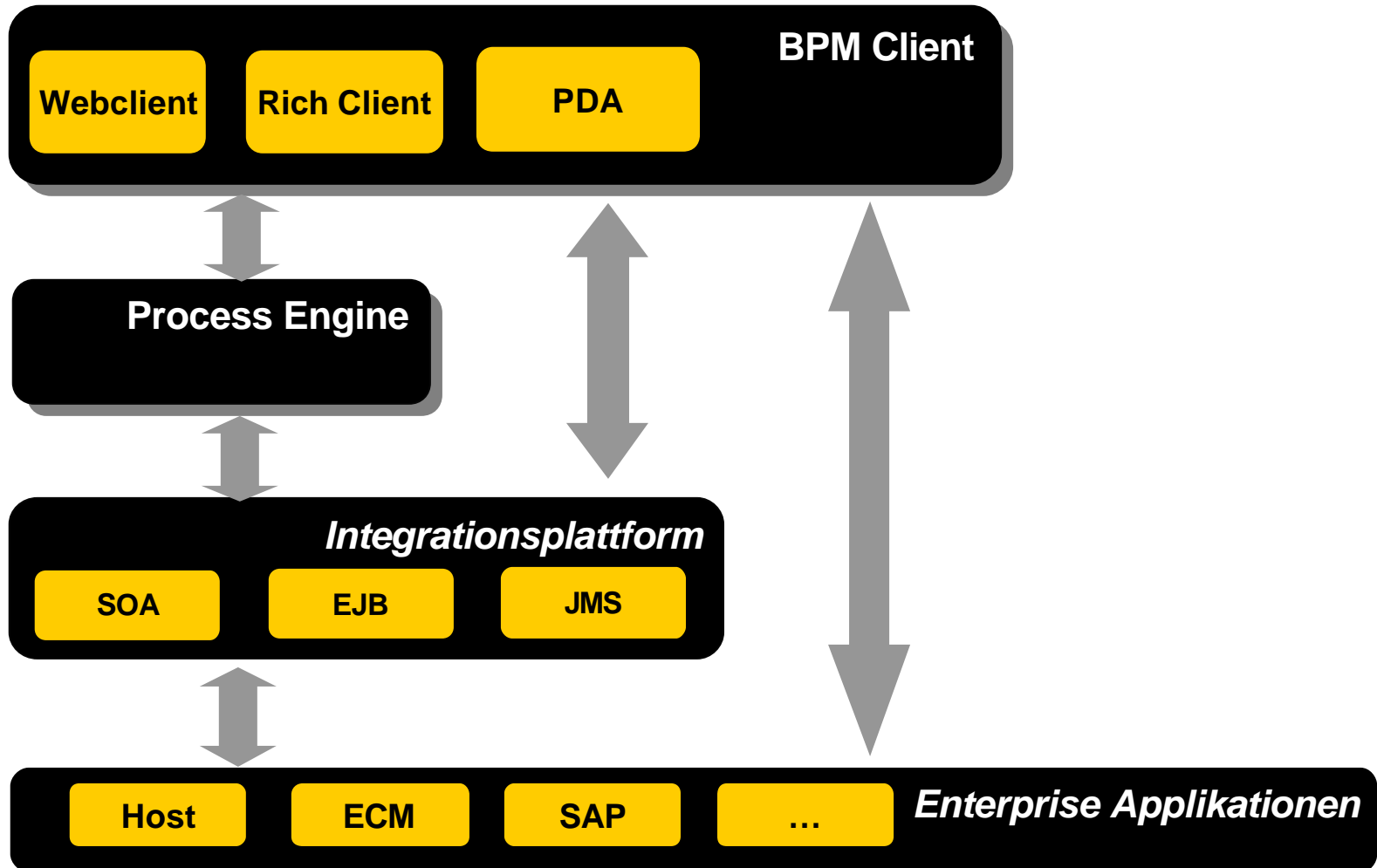
Übersicht Business Process Management (BPM)

- **Definition Prozess:**
Ein Prozess ist eine strukturierte und gesteuerte Folge von Aktivitäten, die angibt, wie aus einem vorgegebenen Input ein bestimmtes Ergebnis ("Output") erzeugt werden soll [1].
- **Definition Geschäftsprozess:**
Ein Geschäftsprozess ist ein dem Unternehmenszweck dienender Prozess.
- **Definition BPM:**
Methode als auch Verfahren zur kontinuierlichen Verbesserung von Geschäftsprozessen unterstützt durch IT-Systeme.

Übersicht BPM



Übersicht BPM

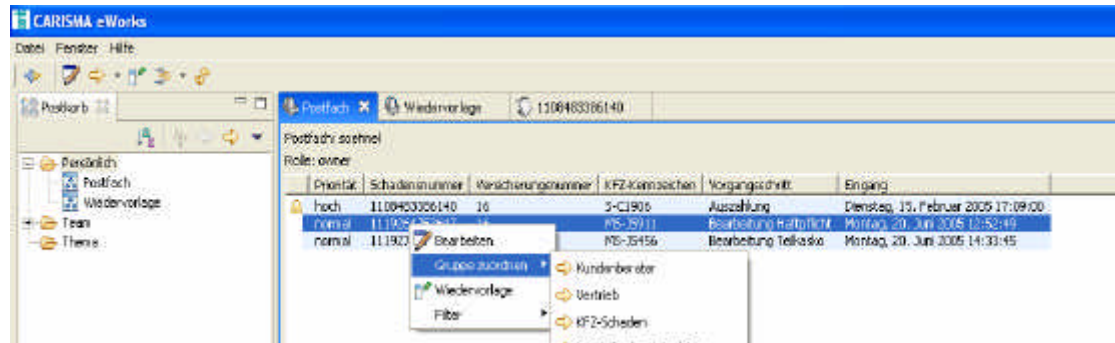


Anforderungen an einen BPM Client

- **Flexibilität**
- **Die Anwendungsmasken sind Prozess-abhängig**
- **Integration von Desktop Anwendungen**
- **Integration von Enterprise Applikationen**
- **Hohe Performanz**
- **Einfache Bedienung**
- **Erweiterbarkeit**
- **Rollenfunktionalität**

Grundsätzliche Funktionalität eines BPM Client

- **Postkorb (Worklist)**
 - Liste der Aufgaben eines Benutzers und/oder eines Teams
 - Verteilung der Aufgaben
 - Priorisierung der Aufgaben
 - Wiedervorlage
- **Prozessabhängige Bearbeitungsmasken**
 - Eigenständige Formulare
 - Integration von anderen Applikationen (z.B. Office, SAP-Gui)
 - Notizen anlegen
 - Vorgangsmappe
 - Anzeige Akten



Agenda

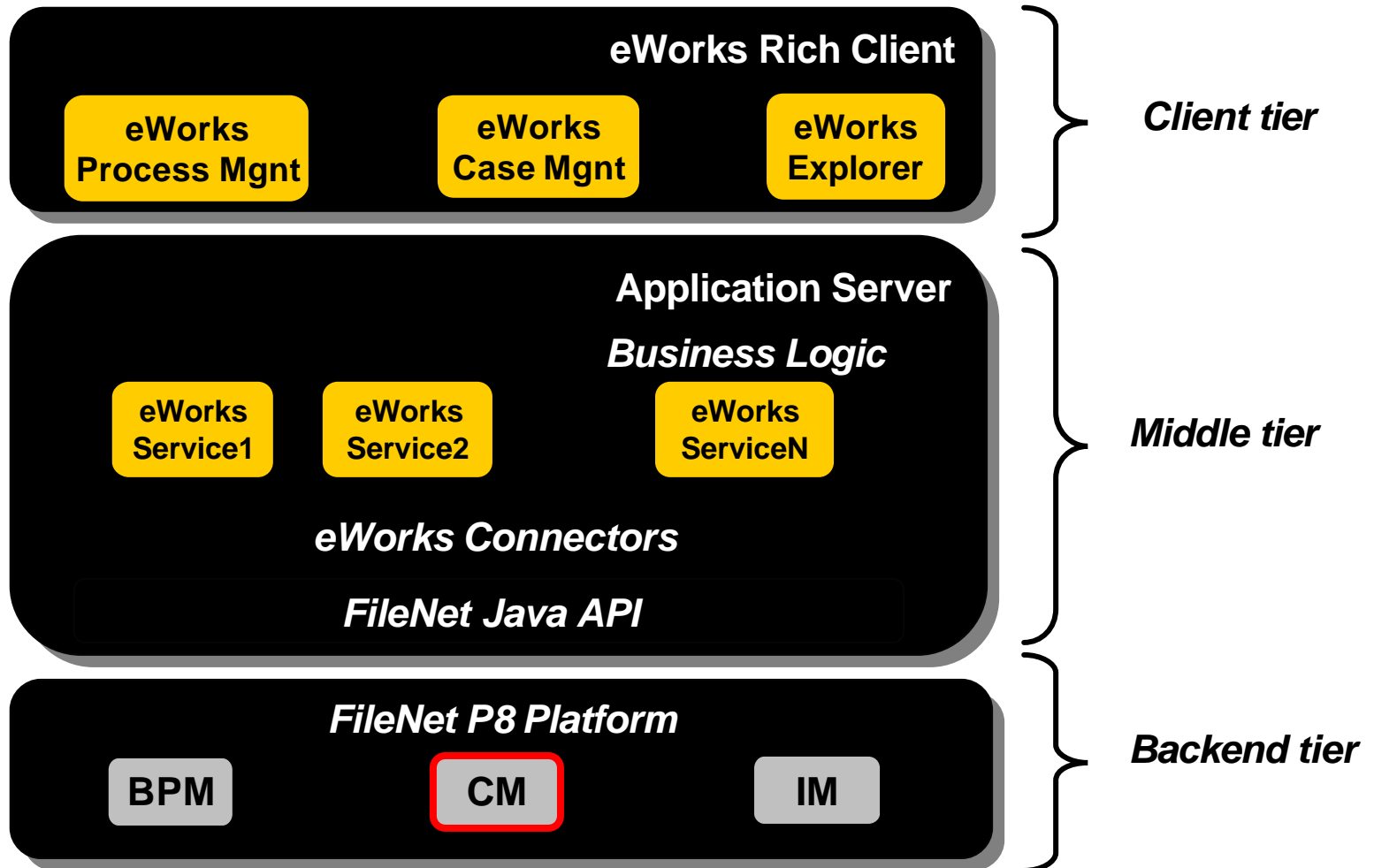
- Kurzvorstellung CENIT
- Einführung BPM
- **Überblick eWorks**
- Warum Eclipse RCP
- Warum Spring Framework
- Tips und Tricks
- Diskussion



CENIT eWorks – Überblick

- **CENIT eWorks Rich Client**
 - **Branchenunabhängiger BPM Client auf Basis FileNet P8**
 - **Ein konfigurierbares Produkt mit Standardfunktionalitäten**
 - Postkorb
 - Aktenverwaltung
 - „Explorer“ für Enterprise Content Management
 - **Erweiterbar für kundenspezifische Anforderungen**
- **Technologische Basis**
 - **FileNet P8 als Plattform (CM, BPM, IM)**
 - **3 Schicht Architektur**
 - Client basierend auf Eclipse RCP
 - Middle Tier mit Services für Logik und Zugriff
 - Datenhaltung in FileNet P8

CENIT eWorks – Architektur



Agenda

- Kurzvorstellung CENIT
- Einführung BPM
- Überblick eWorks
- **Warum Eclipse RCP**
- Warum Spring Framework
- Tips und Tricks
- Diskussion



Warum Eclipse RCP

- **Vorteile**
 - **Sehr gute Erweiterbarkeit durch Plugins über Extension Points**
 - **Intelligenter Update Mechanismus**
 - **Bietet viele wiederverwendbare Komponenten**
Help UI, Update, Intro, Cheat Sheets, Forms etc.
 - **Solide Architektur**
 - **Läuft auf dem Benutzer PC**
 - **Möglichkeit der Offline Benutzung**
- **Nachteile**
 - **Hoher Einarbeitungsaufwand**
 - **Komplexität**

Agenda

- Kurzvorstellung CENIT
- Einführung BPM
- Überblick eWorks
- Warum Eclipse RCP
- **Warum Spring Framework**
- Tips und Tricks
- Diskussion



Warum das Spring Framework

- **Vorteile**
 - **Leichtgewichtige Komponenten (POJOs)**
 - **Es können unterschiedliche Remote Services und Protokolle verwendet werden**
 - **Mit Acegi existiert eine Security Infrastruktur**
 - **Gute Architektur**
- **Nachteile**
 - **Einarbeitungsaufwand**
 - **Konfiguration kann schnell komplex werden**

Agenda

- Kurzvorstellung CENIT
- Einführung BPM
- Überblick eWorks
- Warum Eclipse RCP
- Warum Spring Framework
- **Tips und Tricks**
- Diskussion



Tipps und Tricks

- **ClassLoader**
 - Jedes Bundle (Plugin) hat seinen eigenen Classloader
 - Das Einbinden bestehender Libraries kann schwierig werden (z.B. JDIC, Spring)
 - Classloading Mechanismus muss verstanden werden
- **Unterscheidung zwischen Compile und Runtime Umgebung**
 - Es gibt eine strikte Trennung zwischen Compile und Runtime Umgebung
 - Die Compile Umgebung wird über die Standard Preference Page “Java build path” konfiguriert
 - Die Runtime Umgebung über den Plugin-Manifest Editor

Einbindung Spring Framework

- **CorePlugin**
 - **Stellt zentrale Dienste zur Verfügung u.a. den Zugriff auf Spring**
 - **Bean Konfiguration liegt im Plugin Verzeichnis**

```
public Object getBean(String beanName) {
    ClassLoader loader = Thread.currentThread().getContextClassLoader();
    try {
        if (m_ApplicationContext == null) {
            Thread.currentThread().setContextClassLoader(getDefault().getClass().getClassLoader());
            m_ApplicationContext = new ClassPathXmlApplicationContext(BEANS_XML);
        }
        return m_ApplicationContext.getBean(beanName);
    }
    finally {
        Thread.currentThread().setContextClassLoader(loader);
    }
}
```

Rollenfunktionalität

- **Anforderung:** In Abhängigkeit der Rollenzugehörigkeit stehen dem Benutzer andere Funktionen zur Verfügung
- **Lösung:** IActionDelegate2 um die Funktion canEnable erweitert
 - In der selectionChanged Methode wird canEnable aufgerufen und die Action enabled oder disabled
 - canEnabled kann auch dazu benutzt werden, Kontext-abhängig Funktionen zu de- oder aktivieren

```
protected boolean canEnable(IStructuredSelection selection) {  
    Iterator iterator = selection.iterator();  
    while (iterator.hasNext()) {  
        Document document = (Document)iterator.next();  
        if (document.isCheckedOut()) {  
            return false;  
        }  
    }  
    return true;  
}
```

Update Mechanismus

- **Der integrierte Update Mechanismus von Eclipse ist für den normalen Anwender zu komplex**
- **Die IT möchte sicherstellen, das alle Anwender mit den gleichen Versionen arbeiten**
- **Wenn alle Benutzer zur gleichen Zeit das Update holen, kann es zu Beeinträchtigungen des Netzwerkes kommen**
- **Lösung: Implementierung eines eigenen Update Mechanismus auf Basis des Plugins `org.eclipse.update.core`**

Messages an den Client

- **Anforderung: Der Client muss von außen über Messages benachrichtigt werden, z.B. neuer Eintrag im Postkorb, oder Bearbeitungsende in Office**
- **Lösung: Aufbau eines Event Mechanismus für externe Ereignisse**
 - **Unterstützung für verschiedene Event Publisher (JMS, Filesystem, DDE)**

Zeit für Fragen

Quellen

- [1] Einführung in Business Process Management (BPM) und BPM Systeme (BPMS), Dr. Christoph Strnadl, <http://www.bpm-guide.de/articles/55>
- [2] Eclipse Foundation <http://www.eclipse.org/>
- [3] Spring Framework <http://www.springframework.org>

Vielen Dank für Ihre Aufmerksamkeit ...

©CENIT AG Systemhaus
Max-Planck-Strasse 17
D – 40699 Erkrath

Dirk Günther
+49 211 530652 32
d.guenther@cenit.de

Weitere Informationen unter <http://www.cenit.de>