Apache James

Der Java Mail Server

Henning P. Schmiedehausen

<henning@apache.org>

KNF Kongreß 2005

Nürnberg

Über den Vortragenden

- KNF (+ IN Franken) Gründungsmitglied
- Schwerpunkt Java & J2EE seit 1999
- Mitarbeit bei der Apache Software Foundation seit 2001
- Mitglied der ASF seit 2005

© 2005 Henning Schmiedehausen

Ein Apache **Mail** Server?

- Die Apache Software Foundation bietet eine komplette Mail Server Lösung?
- Und zwar bereits seit 1999?
- Ein Mailserver, der auf den allermeisten Plattformen funktioniert?

  (auf denen es eine JDK 1.3 kompatible Runtime gibt)
- Der in andere Projekte integrierbar und erweiterbar ist?

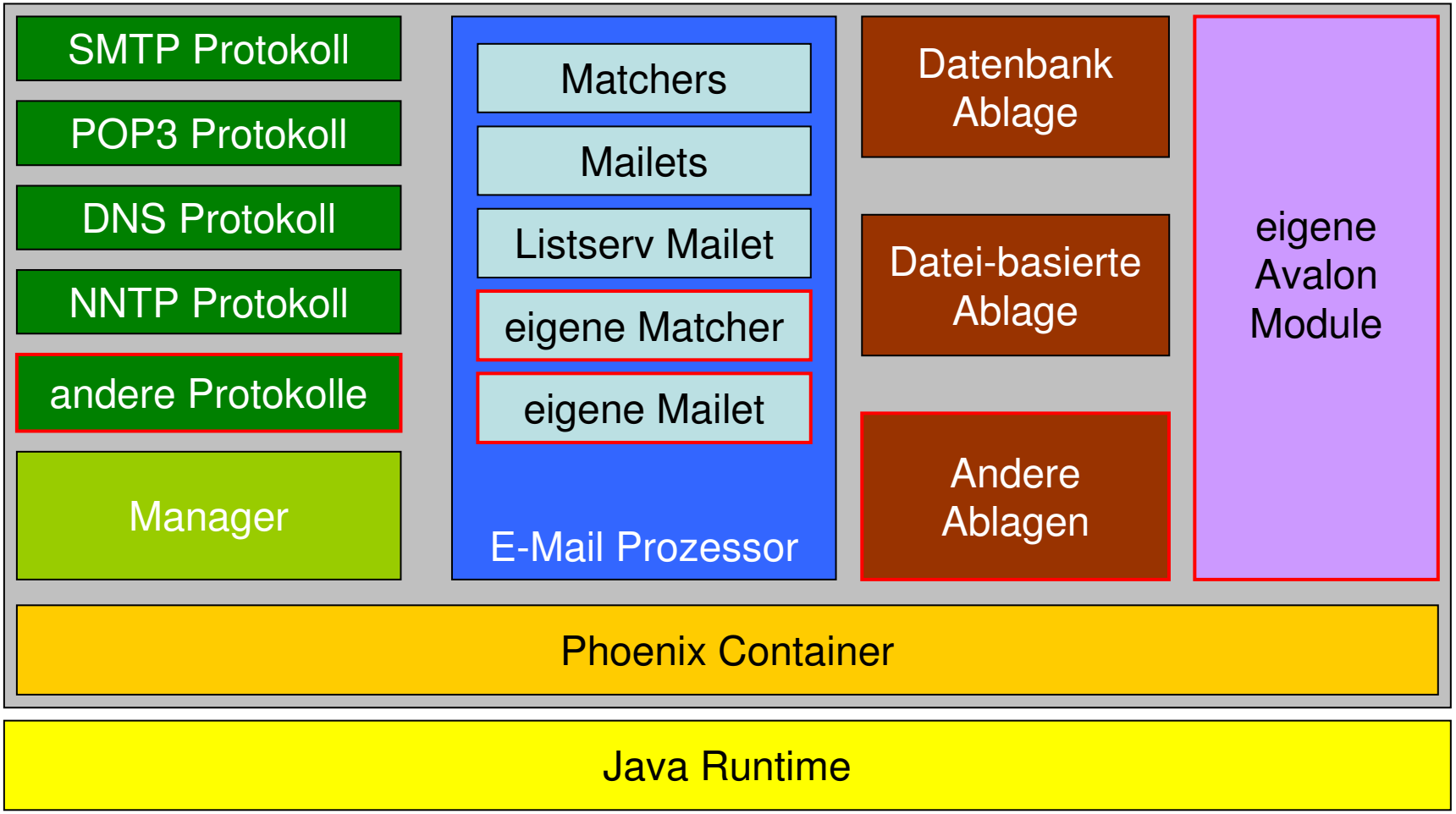
Apache James Projekt

- Gründung 1999
- Top-Level ASF Projekt seit 2002
- “Java Apache Mail Enterprise Server”
- 100% pure Java, JDK 1.3+, JavaMail API
- Unterstützt SMTP, POP3, NNTP und das Abholen von Mails von anderen Servern
- Verwendet JDBC und JNDI für die Benutzerverwaltung und Datenablage

Überblick

- Komponenten-basiert (Apache Avalon)
- Container-basiert (mit Apache Phoenix)
- Läuft als normales Programm
- Kann in Applikationsserver eingebunden werden (z.B. JBoss oder Geronimo)

James Architektur



© 2005 Henning Schmiedehausen

James und Avalon

- Aktuelle Release: 2.2.0, 15. Juni 2004
- Apache Avalon wurde 2004 aufgelöst; hierbei ging viel Dokumentation verloren
- Apache Phoenix ebenfalls eingestellt (Lebt weiter als loom.codehaus.org)
- Die aktuelle Version (2.2.0) verwendet “gut versteckte” Avalon Software
- Dokumentation derzeit unbefriedigend

© 2005 Henning Schmiedehausen

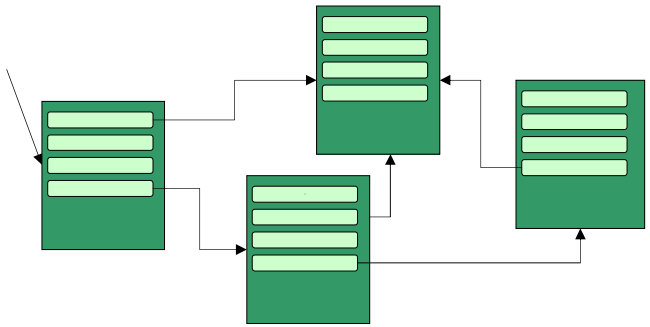
James Anwendungen

- Nur ein weiterer SMTP / POP3 Server?
- James versucht der „Tomcat für Mail Anwendungen“ zu sein
- Universeller Container für Mailprogramme
- Modulare Erweiterungsmöglichkeiten
  - Mailets und Matchers
  - Selbstgeschriebener Code
  - Eigene Komponenten („Avalon Blocks”)

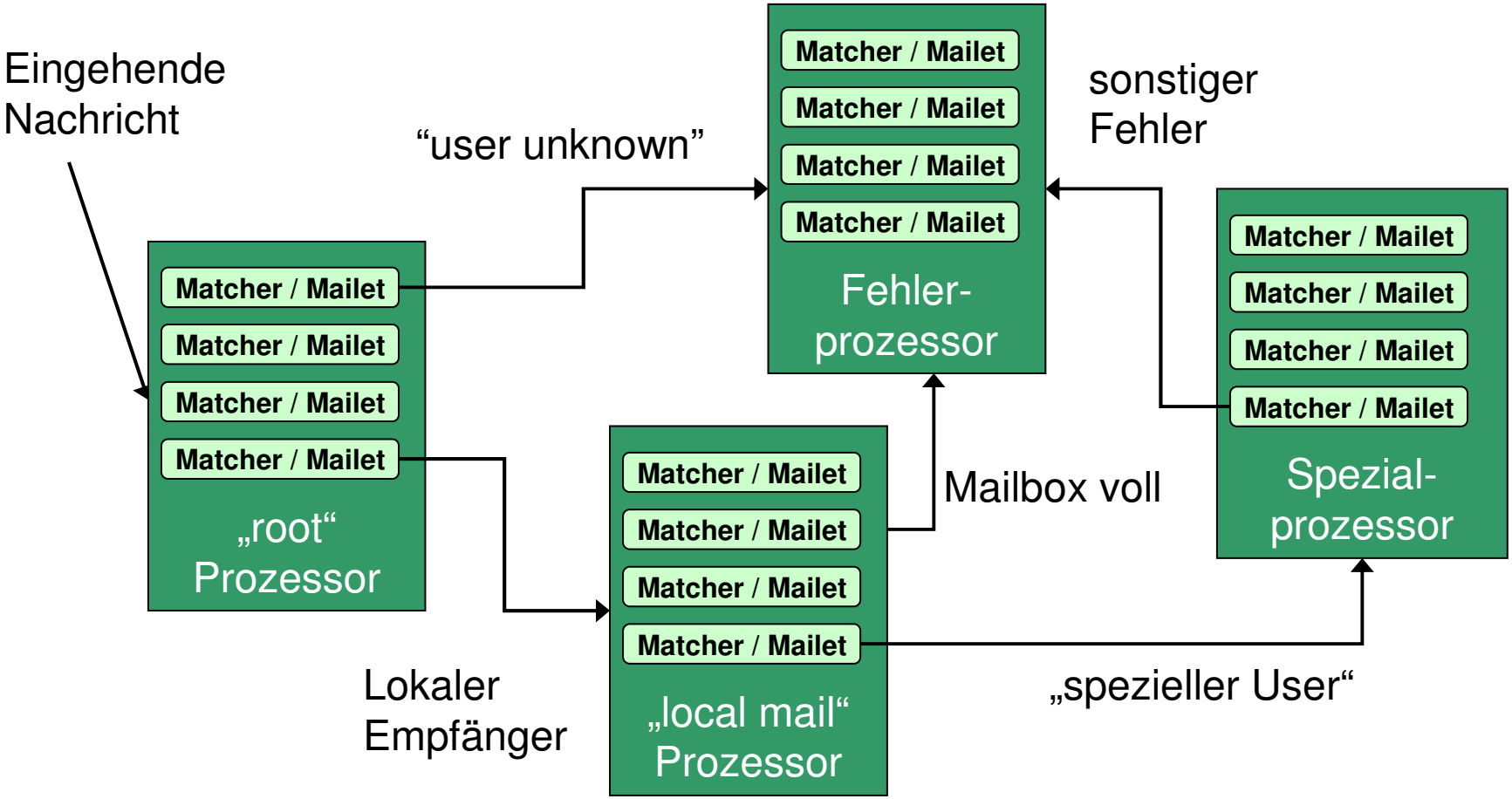
© 2005 Henning Schmiedehausen

James Nachrichtenverarbeitung

- E-Mail durchläuft einen “Prozessorbaum”
- Jeder Prozessor enthält ein oder mehrere “Mailet/Matcher” Paare
- Nachrichten können für verschiedene Empfänger aufgeteilt werden
- Spezielles Mailet (`ToProcessor`) um Mail zwischen Prozessoren zu transportieren



Nachrichtenverarbeitung



© 2005 Henning Schmiedehausen

James Demo

- Start und Stopp von der Kommandozeile
- Benutzer durch Manager Konsole hinzufügen
- Senden und Empfangen von E-Mail

James Programmierung

Mailet/Matcher API

- Mailet (Mail Servlet) & Matcher
- Werden immer paarweise verwendet

- Matcher – Nachrichtenauswahl
- Mailet – Nachrichtenbearbeitung

- Aktuelles API ist bei Version 1.0, eine zweite Version ist bereits geplant

© 2005 Henning Schmiedehausen

Matcher API

```
public interface Matcher {
    void init(MatcherConfig cfg);
    void destroy();
    Collection match(Mail mail);

    MatcherConfig getMatcherConfig();
    String getMatcherInfo();
};
```

Vorhandene Matcher

- Matcher Beispiele
  - **`All`** – trifft alle Nachrichten
  - **`HasAttachment`** – multipart/mixed Mails
  - **`HasHeader`** – Findet eine Kopfzeile
  - **`HostIs`**, **`HostIsLocal`** – prüft Hostnamen
  - **`RecipientIs`**, **`UserIs`** – prüft Usernamen
- Die aktuelle James Distribution beinhaltet mehr als 25 verschiedene Matcher

Mailet API

```
public interface Mailet {
    void init(MailetConfig cfg);
    void destroy();
    void service(Mail mail);

    MailetConfig getMailetConfig();
    String getMailetInfo();
};
```

Vorhandene Mailets

- Mailet Beispiele
  - **`AddHeader`**, **`AddFooter`** – Mails ändern
  - **`Forward`** – Nachricht an Empfänger schicken
  - **`ToProcessor`** – mit anderem Prozessor weiterbearbeiten
  - **`LocalDelivery`**, **`RemoteDelivery`**
  - **`JDBCAlias`** – Mail Aliases aus Datenbank
- James enthält ~ 20 verschiedene Mailets

Beispiel: Ein eigenes Mailet

Aufgabe: „*Das Mailsystem soll jede Mail an einen lokalen Benutzer mit einer Bestätigungsmail beantworten*“

- Matcher: **`RecipientIsLocal`**
- Eigenes Mailet

Ein eigenes Mailet

```
public class NotifyMailet
  extends GenericMailet
  implements Mailet
{
  public void service(Mail mail) {
    MailAddress sender    = mail.getSender();
    Collection rcpts      = mail.getRecipients();
    MailetContext context = getMailetContext();

    for (Iterator it = rcpts.iterator(); it.hasNext(); ) {
      MailAddress rcpt = (MailAddress) it.next();
      MimeMessage msg = createNotifyMsg(sender, rcpt);
      context.sendMail(msg);
    }
  }
}
```

Mailet aktivieren

config.xml (James Hauptkonfiguration)

```
[…]
<processor name="root">
[…]
<mailet match="RecipientIsLocal"
   class="de.intermeta.james.mailet.NotifyMailet"/>
[…]
</processor>
[…]
```

Mailet Demo

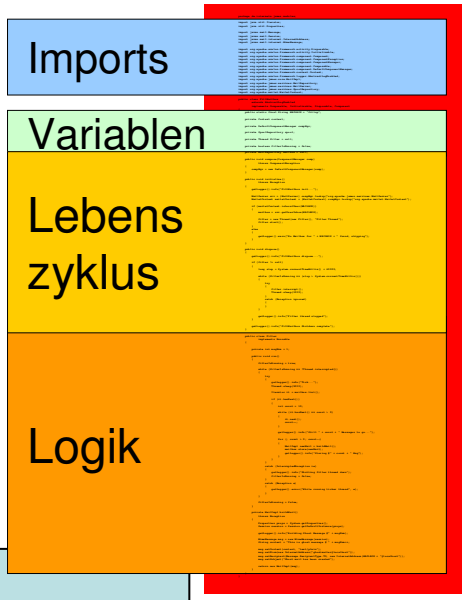
James erweitern

Aufgabe: „*Wenn die Mailbox nicht leer ist, dann sorge dafür, daß immer zehn Nachrichten in der Mailbox sind*“

- Eigene Komponente(Avalon „Block“)
- Wird in James eingebettet
- Phoenix lädt und konfiguriert den Block
- Muß als Jar-File installiert werden

Eine eigene James Erweiterung

- 200 Programmzeilen
- **`Composable`**, **`Initializable`**, **`Disposable`** Lebenszyklus
- Wird durch “assembly.xml” mit James verbunden




```
<block name="fillmailbox"
    class="de.intermeta.james.modules.FillMailbox">
  <provide name="James"
    role="org.apache.mailet.MailetContext"/>
  <provide name="James"
    role="org.apache.james.services.MailServer"/>
 </block>
```

Erweiterungs-Demo

Die Zukunft von James

- James V3 ist in der Entwicklung
  - IMAP Unterstützung
  - Bessere SPAM Filterung
  - “fast fail” Unterstützung für SMTP
- Entwicklung geht langsam aber kontinuierlich voran
- Neue Release soll noch 2005 kommen

© 2005 Henning Schmiedehausen

(Meine) James Wunschliste

- IMAP Unterstützung
- RSS Feed Protokolle (lesen/schreiben)
- Kein HTTP Protokoll?
- Instant Messaging Zugriff?
- STARTTLS für SMTP
- Nicht-Phoenix Container (Loom, Excalibur)
- (bessere) J2EE Integration

© 2005 Henning Schmiedehausen

…und von hier nach…?

- Apache James Homepage
  - http://james.apache.org/
- Dieser Vortrag (Folien & Programme)
  - http://henning.schmiedehausen.org/james/
- Sonstige James Informationen
  - http://www.java201.com/resources/browse/31-all.html

Fragen?

Vielen Dank für die Aufmerksamkeit!