Behandlung von Netzwerk- und Sicherheitsaspekten in einem Werkzeug zur verteilten Paarprogrammierung

Sandor Szücs

Institut Mathematik und Informatik - Freie Universität Berlin

15. April 2010

Freie Universität Berlin

Agenda

Next

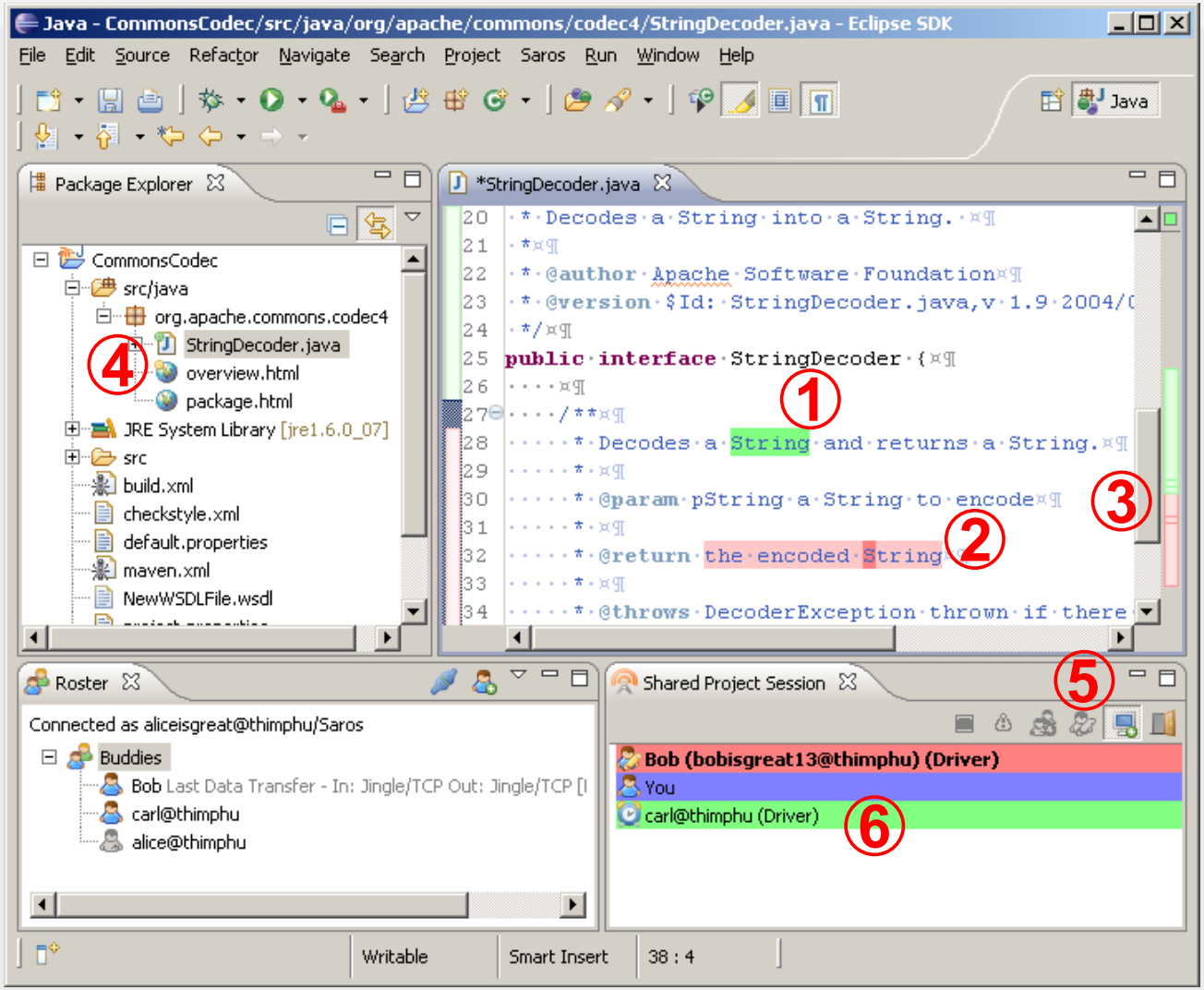
Fehler im Titel!

- Behandlung von Netzwerk- und Sicherheitsaspekten in einem Werkzeug zur verteilten Paarprogrammierung
- Behandlung von Netzwerk- und Testaspekten in einem Werkzeug zur verteilten Paarprogrammierung

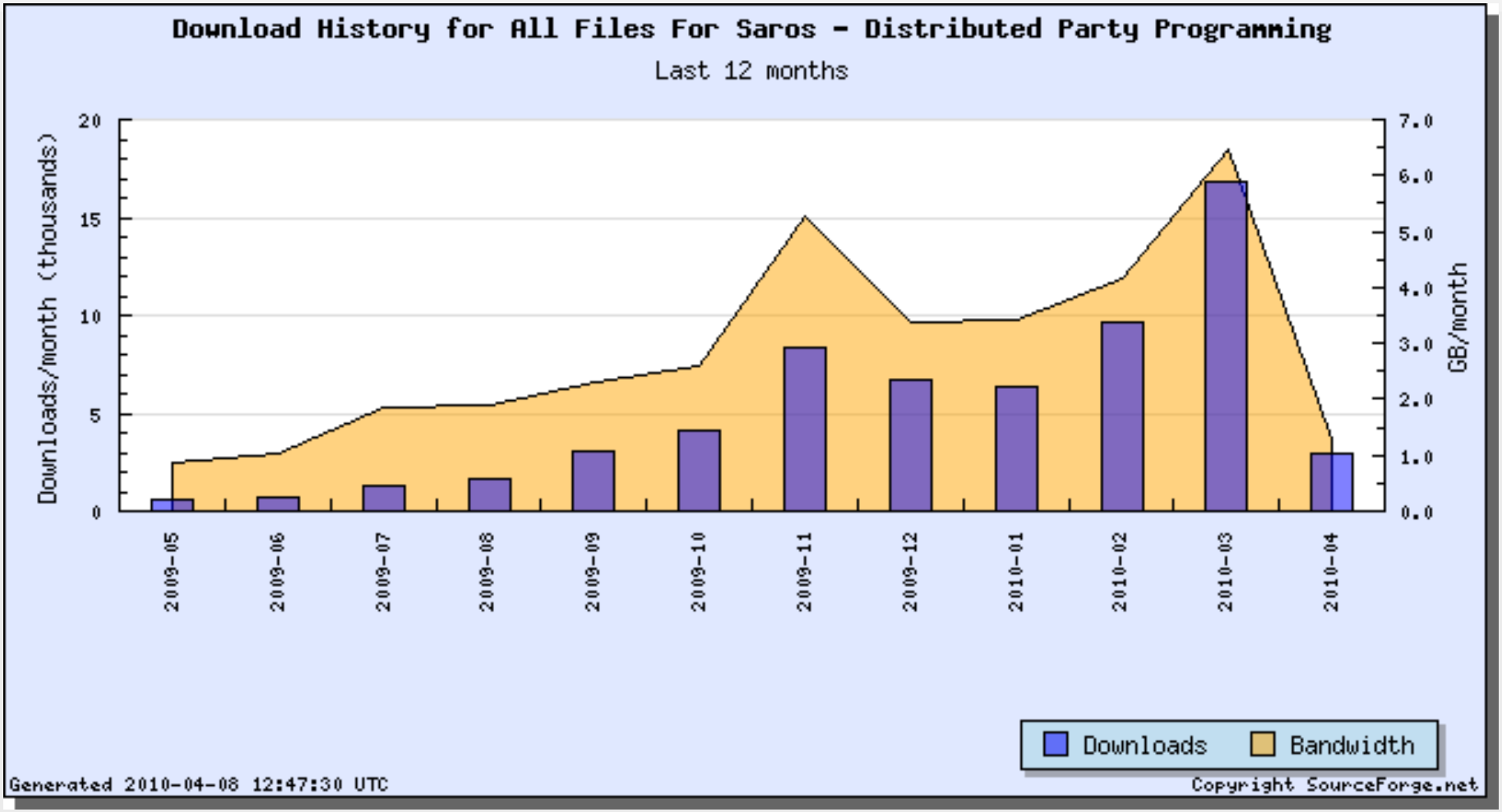
Fehler im Titel!

- Behandlung von Netzwerk- und Sicherheitsaspekten in einem Werkzeug zur verteilten Paarprogrammierung
- Behandlung von Netzwerk- und Testaspekten in einem Werkzeug zur verteilten Paarprogrammierung

Was ist Saros?



Saros wird populärer



Tätigkeiten im Projekt

- APs lösen (== mehr Funktionalität)
- Bugfixing (== weniger Defekte)
- Outreach (== Werbung)
- STF (== Testmöglichkeit schaffen)
- Kaffee kochen

Tätigkeiten im Projekt

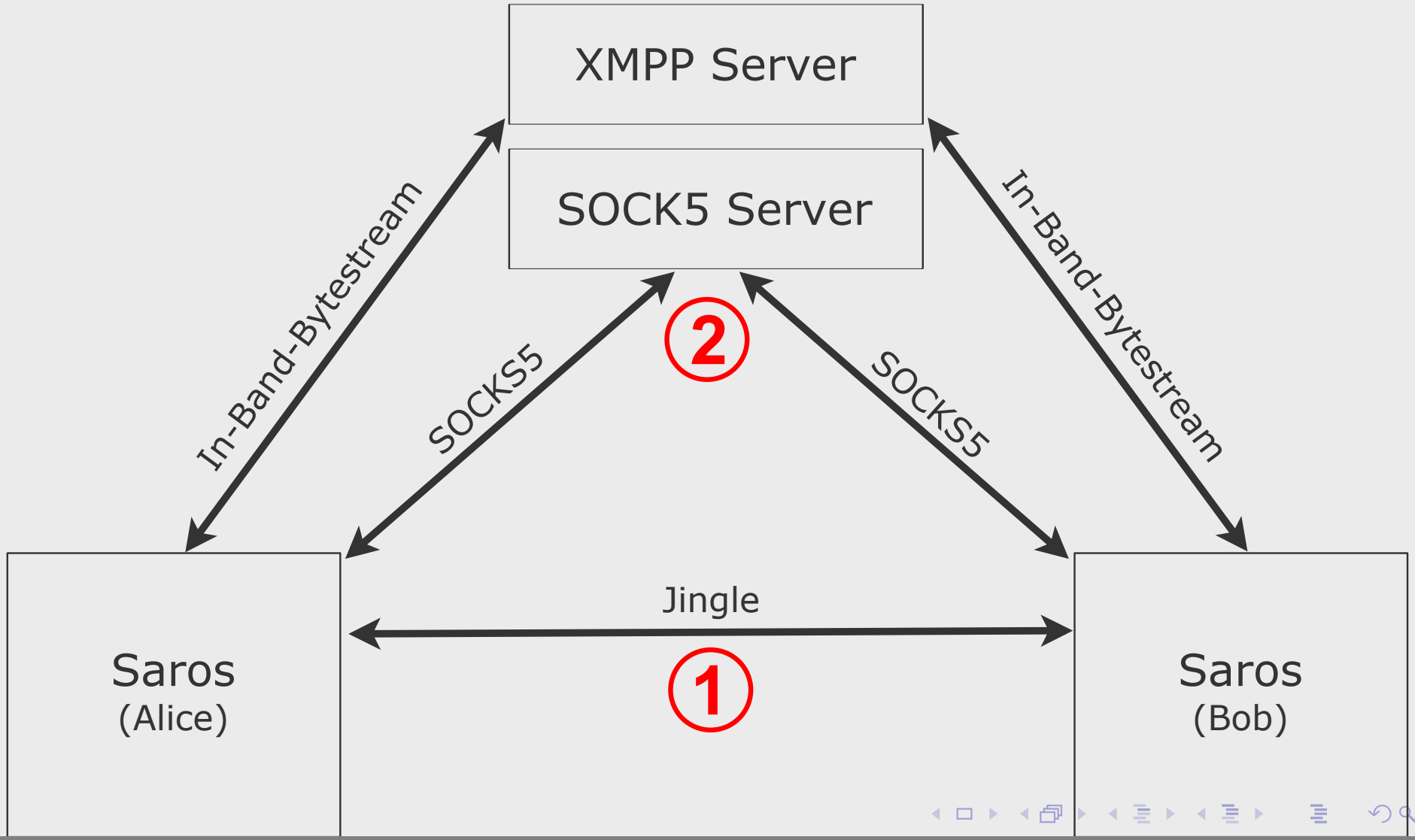
- APs lösen (== mehr Funktionalität)
- Bugfixing (== weniger Defekte)
- Outreach (== Werbung)
- STF (== Testmöglichkeit schaffen)
- Kaffee kochen

Next

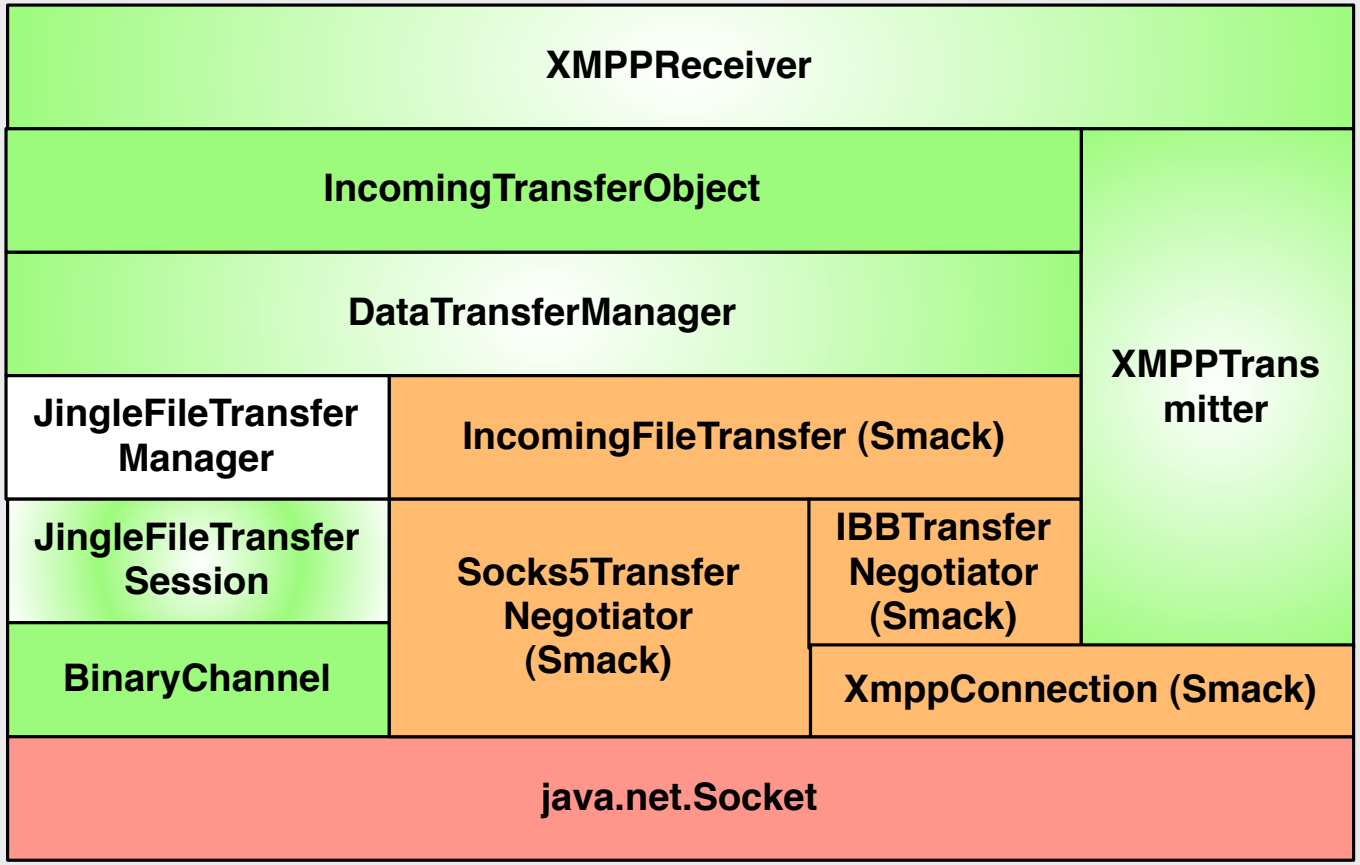
Netzwerk Protokoll - XMPP

- Message - unidirektional
- IQ - bidirektional / Frage-Antwort
- XEP - Erweiterungen

Netzwerk Protokolle - XEP 3



Saros Netzwerkschicht



Was wird transferiert?

- Projektsynchronisierung
- Benutzeraktivitäten (Textselektion, Texteinfügen, Sichtfeld, ...)
- **neu** Datenströme (Audio,Video,..) - Danke an Stephan Lau

Wie wird transferiert?

- Projektsynchronisierung, Datenströme und Aktivitäten größer als 16 kB:
  - Jingle/TCP
  - Jingle/UDP
  - Socks5-Bytestreams
  - IBB
- XMPP Message: alle Daten einer Sitzung

Dateisynchronisation optimieren

Netzwerk	Verbindung	Typ	Durchsatz
Internet	Jingle/TCP	Dateien	4992 kB/s
Internet	Jingle/TCP	Archiv	3994 kB/s
Internet	Jingle/UDP	Dateien	974 kB/s
Internet	Jingle/UDP	Archiv	1377 kB/s
Internet	Socks5	Archiv	79 kB/s
Teles VPN	Jingle/TCP	Dateien	1288 kB/s
Teles VPN	Jingle/TCP	Archiv	1664 kB /s
Teles VPN	IBB Teles	Archiv	4 kB/s

Cancelation und Progress-Support von Dateiübertragungen

- Übertragungen sind abbrechbar und ablehnbar
- realistischer Fortschrittsbalken

Benutzeraktivität via Peer-to-Peer

- falls Daten $>$ 16 kB, Verwendung: Cut and paste
- Overhead: Verbindungsaufbau
- zu lösende Probleme:
  - Smacks FileTransfer schließt die Verbindung nach einer Dateiübertragung
  - richtungsorientierter Verbindungsaufbau
- Aktuell: Verbindungsaufbau versagt bei NAT

Benutzeraktivität via Peer-to-Peer

- falls Daten $>$ 16 kB, Verwendung: Cut and paste
- Overhead: Verbindungsaufbau
- zu lösende Probleme:
  - Smacks FileTransfer schließt die Verbindung nach einer Dateiübertragung
  - richtungsorientierter Verbindungsaufbau
- Aktuell: Verbindungsaufbau versagt bei NAT

Benutzeraktivität via Peer-to-Peer

- falls Daten $>$ 16 kB, Verwendung: Cut and paste
- Overhead: Verbindungsaufbau
- zu lösende Probleme:
  - Smacks FileTransfer schließt die Verbindung nach einer Dateiübertragung
  - richtungsorientierter Verbindungsaufbau
- Aktuell: Verbindungsaufbau versagt bei NAT

Next

Motivation

1. Problem: Testbarkeit der Netzschicht
   - Peer-to-Peer Software braucht mindestens 2 Teilnehmer
   - unterschiedliche Instanzen eines Singleton-Objekts notwendig
   - Neal Ford in “10 Ways to Improve Your Code”:

     *singleton is bad because: ..., **untestable**, ...*

2. Problem: Verbindungsaufbau testen
   - Netzwerkeigenschaften: NAT, Firewall, Unzuverlässigkeit
   - als Einzelperson testen?
   - Wiederholbarkeit?
   - Socks5 und localhost
3. Problem: realistische Tests
   - Funktions- und Integrationstests != Unittest

Motivation

1. Problem: Testbarkeit der Netzschicht
   - Peer-to-Peer Software braucht mindestens 2 Teilnehmer
   - unterschiedliche Instanzen eines Singleton-Objekts notwendig
   - Neal Ford in “10 Ways to Improve Your Code”:

     *singleton is bad because: ..., **untestable**, ...*
2. Problem: Verbindungsaufbau testen
   - Netzwerkeigenschaften: NAT, Firewall, Unzuverlässigkeit
   - als Einzelperson testen?
   - Wiederholbarkeit?
   - Socks5 und localhost
3. Problem: realistische Tests
   - Funktions- und Integrationstests != Unittest

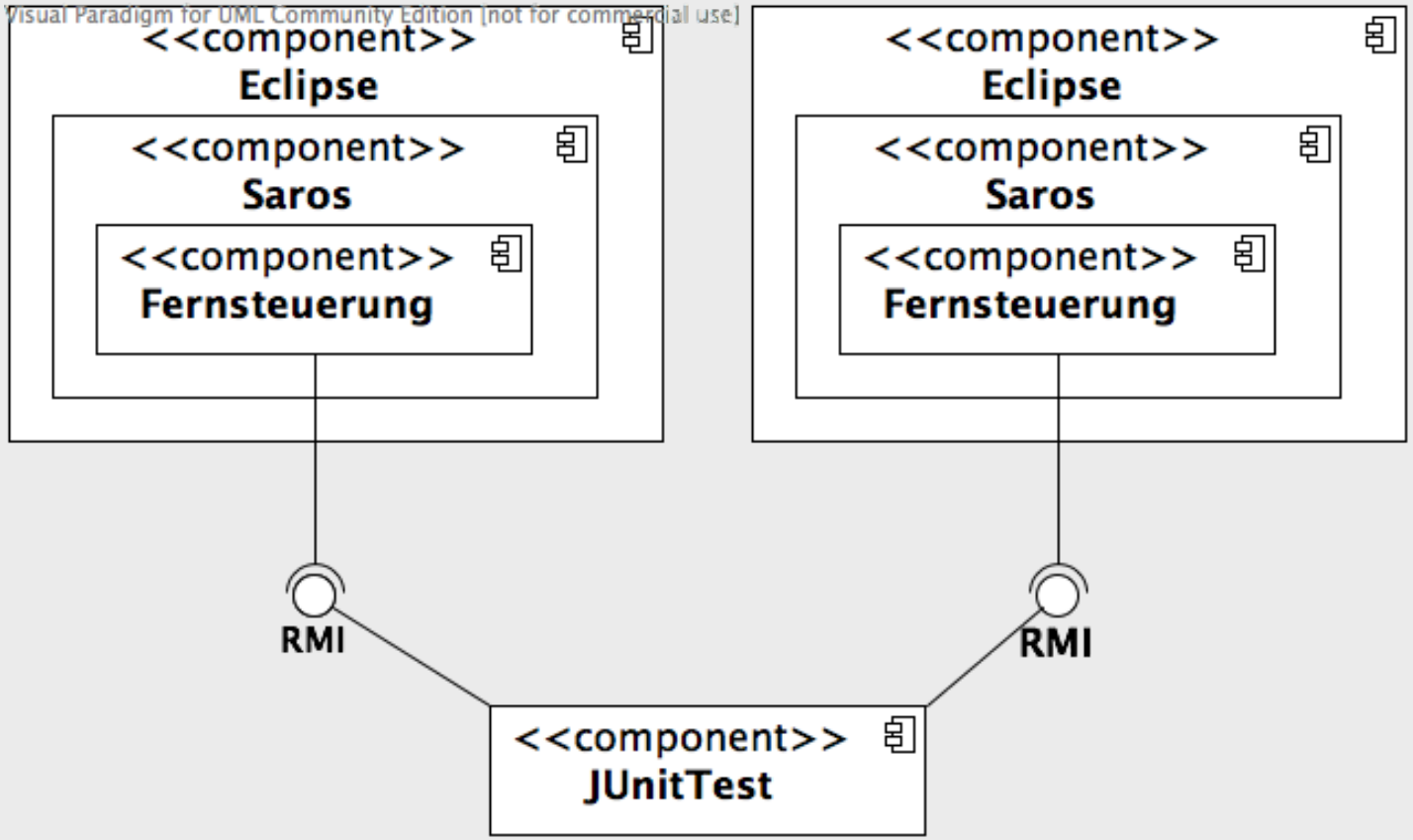
Motivation

1. Problem: Testbarkeit der Netzschicht
   - Peer-to-Peer Software braucht mindestens 2 Teilnehmer
   - unterschiedliche Instanzen eines Singleton-Objekts notwendig
   - Neal Ford in “10 Ways to Improve Your Code”:

     *singleton is bad because: ..., **untestable**, ...*
2. Problem: Verbindungsaufbau testen
   - Netzwerkeigenschaften: NAT, Firewall, Unzuverlässigkeit
   - als Einzelperson testen?
   - Wiederholbarkeit?
   - Socks5 und localhost
3. Problem: realistische Tests
   - Funktions- und Integrationstests != Unittest

gewählter Ansatz - Steuerung

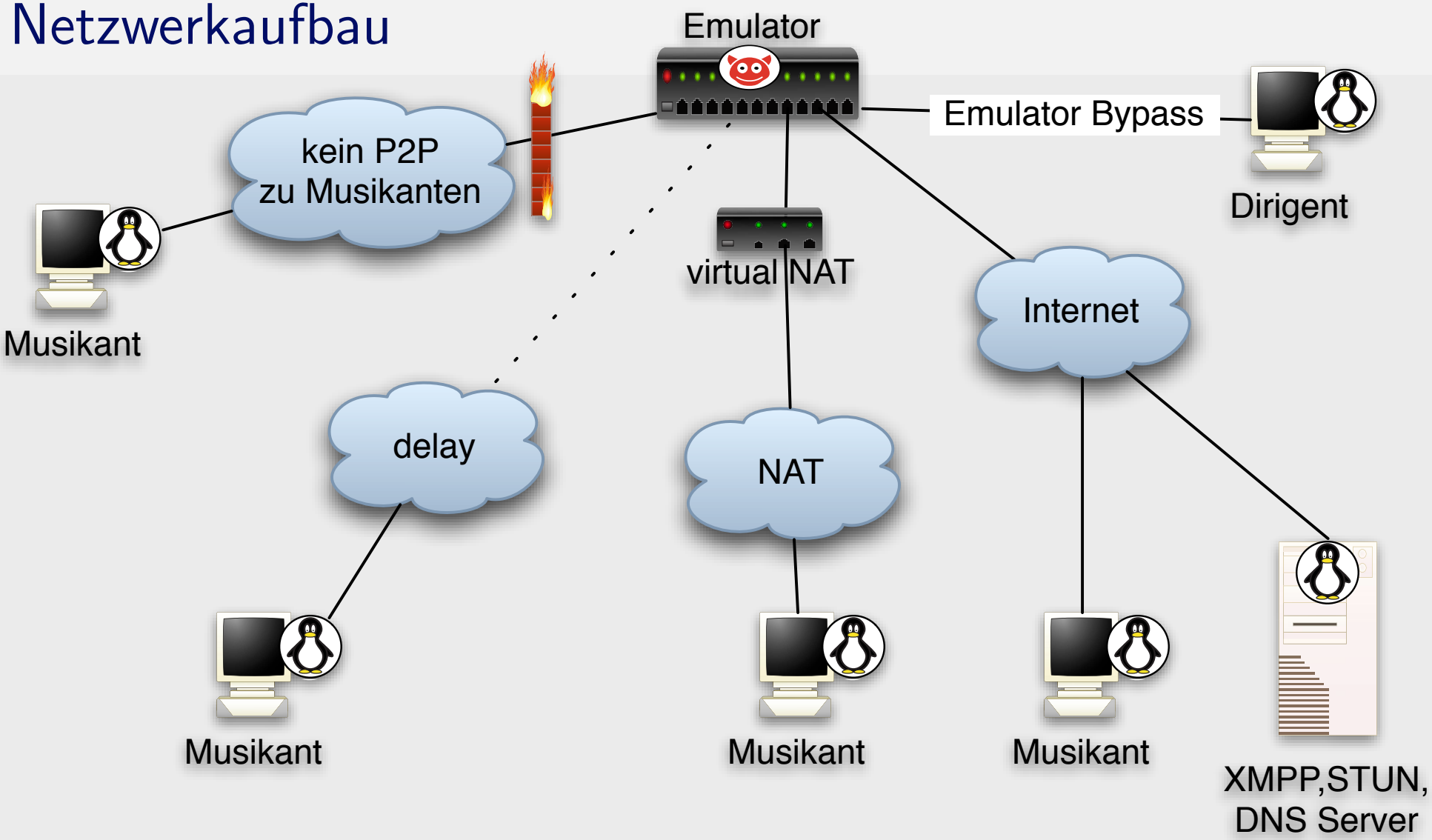
- SWTBot erweitern → Fernsteuerung von Eclipse über RMI



gewählter Ansatz - Netzwerk

- Infrastruktur in einem VM Netz (XMPP, STUN, DNS, Saros,..)
- Emulator - erzeugt Netzwerkeigenschaften (Delay, NAT, ..)

Netzwerkaufbau



Anwendung

- Entwickler schreiben Tests und testen diese lokal
- Verifikation und Wiederholbarkeit im Emulator

Beispiel: Testgleichheit lokal/remote

```
@Before
public void configureInvitee() {
  invitee = new Musician(new JID(
    BotConfig.JID_ALICE),
    BotConfig.PASSWORD_ALICE,
    BotConfig.HOST_ALICE,
    BotConfig.PORT_ALICE);
  invitee.initRmi();
  invitee.openSarosViews();
  invitee.xmppConnect();
}
```

Relevanz

- intern
  - Defekte in einer kontrollierten Umgebung finden
  - weitere Möglichkeit: manuelle Tests - selten bis unnötig (Testtag)
- extern - Interesse an der Funktionalität vom Swtbot Entwickler

Next

Offene Fragen

- Verbindungsaufbau
  - richtungsorientiert
  - nicht persistent
  - Jingle/ICE scheint defekt (?)
  - Timeouts in Smack FileTransfer sind/waren defekt (Danke Henning)
- Wie helfen STF-Tests der Stabilität von Saros?

Offene Fragen

- Verbindungsaufbau
  - richtungsorientiert
  - nicht persistent
  - Jingle/ICE scheint defekt (?)
  - Timeouts in Smack FileTransfer sind/waren defekt (Danke Henning)
- Wie helfen STF-Tests der Stabilität von Saros?

Zusammenfassung

- Hauptthemen
  - Optimierung
  - Abbrechbarkeit und Fortschritt
  - Testbarkeit
- Nebenbaustellen
  - Defektbeseitigung (verschiedene XMPP Clients parallel)
  - Benutzersupport (E-Mail, IRC)

Zusammenfassung

- Hauptthemen
  - Optimierung
  - Abbrechbarkeit und Fortschritt
  - Testbarkeit
- Nebenbaustellen
  - Defektbeseitigung (verschiedene XMPP Clients parallel)
  - Benutzersupport (E-Mail, IRC)

QA

Danke!