



Verbesserung einer XMPP-Bibliothek für den Einsatz in verteilter Paarprogrammierung

Diplomarbeit

Henning Staib

Tutor: Karl Beecher und Stephan Salinger



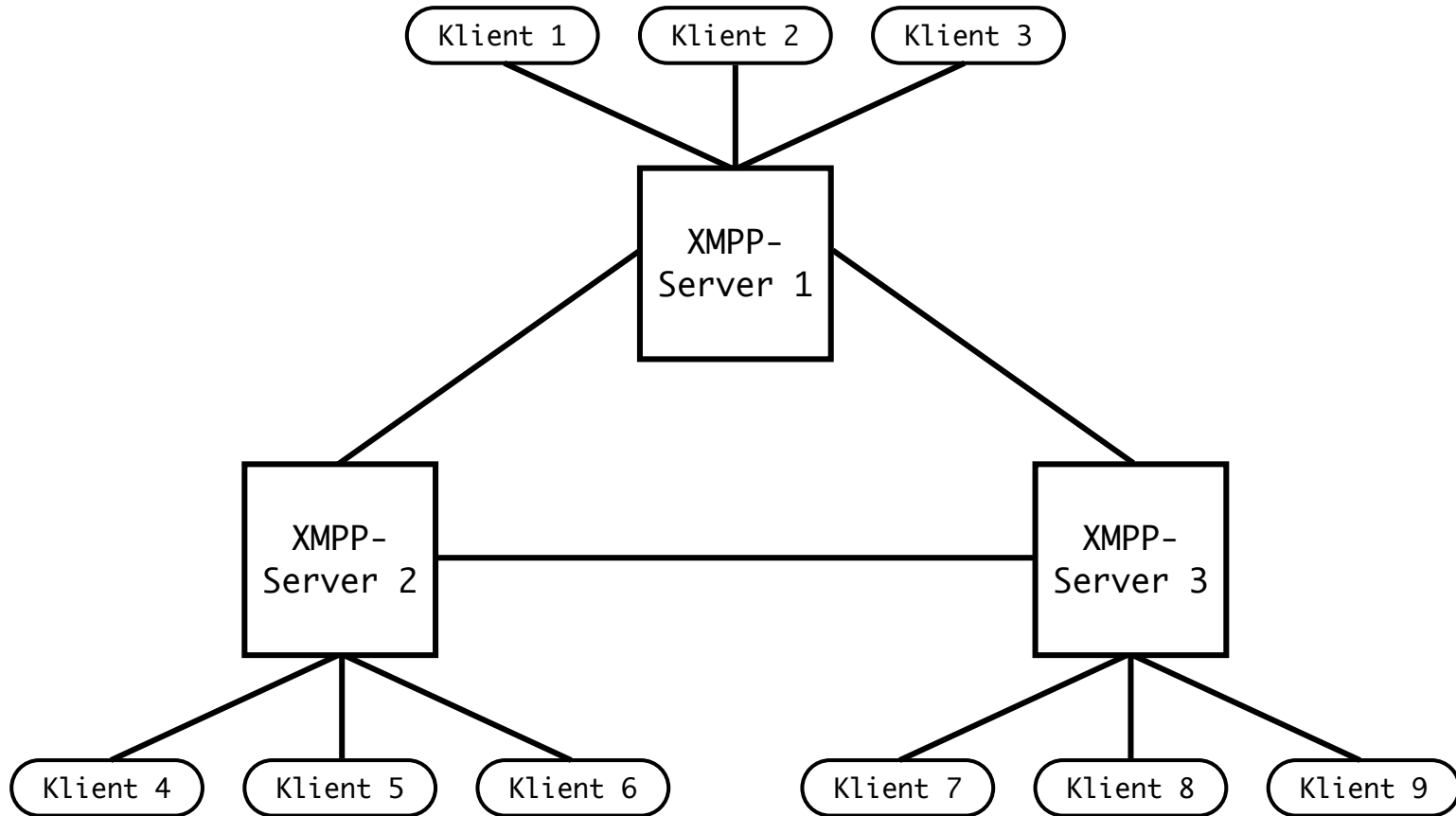
Inhalt

- Ausgangssituation
- Aufgabenstellung
- Lösungsansätze
- SOCKS5 Bytestream Patch
- weitere Patches
- Auswertung

Ausgangssituation

- Kommunikation von Saros über das Netzwerk mittels XMPP
- XMPP (Extensible Messaging and Presence Protocol)
 - XML basiertes Protokoll zum Austausch von Nachrichten, Statusinformationen und Daten zwischen den Teilnehmern des Netzwerkes
 - zusätzliche Dienste definiert in XEPs (XMPP Extension Protocols)
 - Service Discovery zum Auffinden von Diensten
 - Multi-User-Chat
 - Datenaustausch über SOCKS5 Bytestream, In-Band Bytestream, Jingle

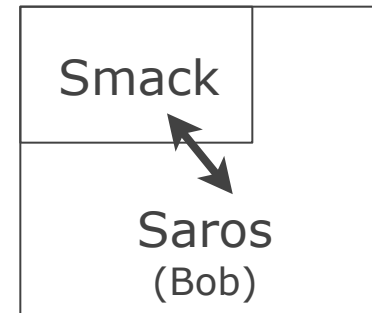
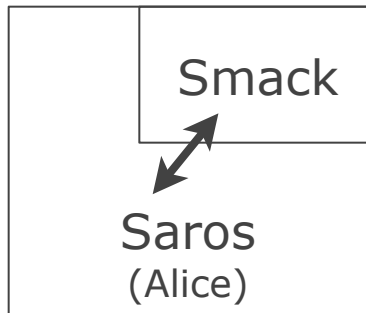
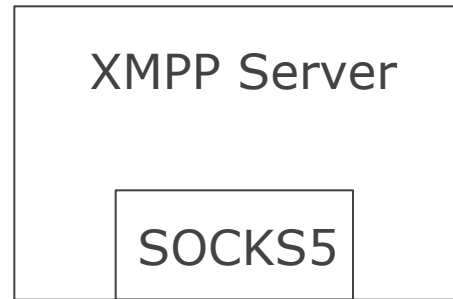
Ausgangssituation



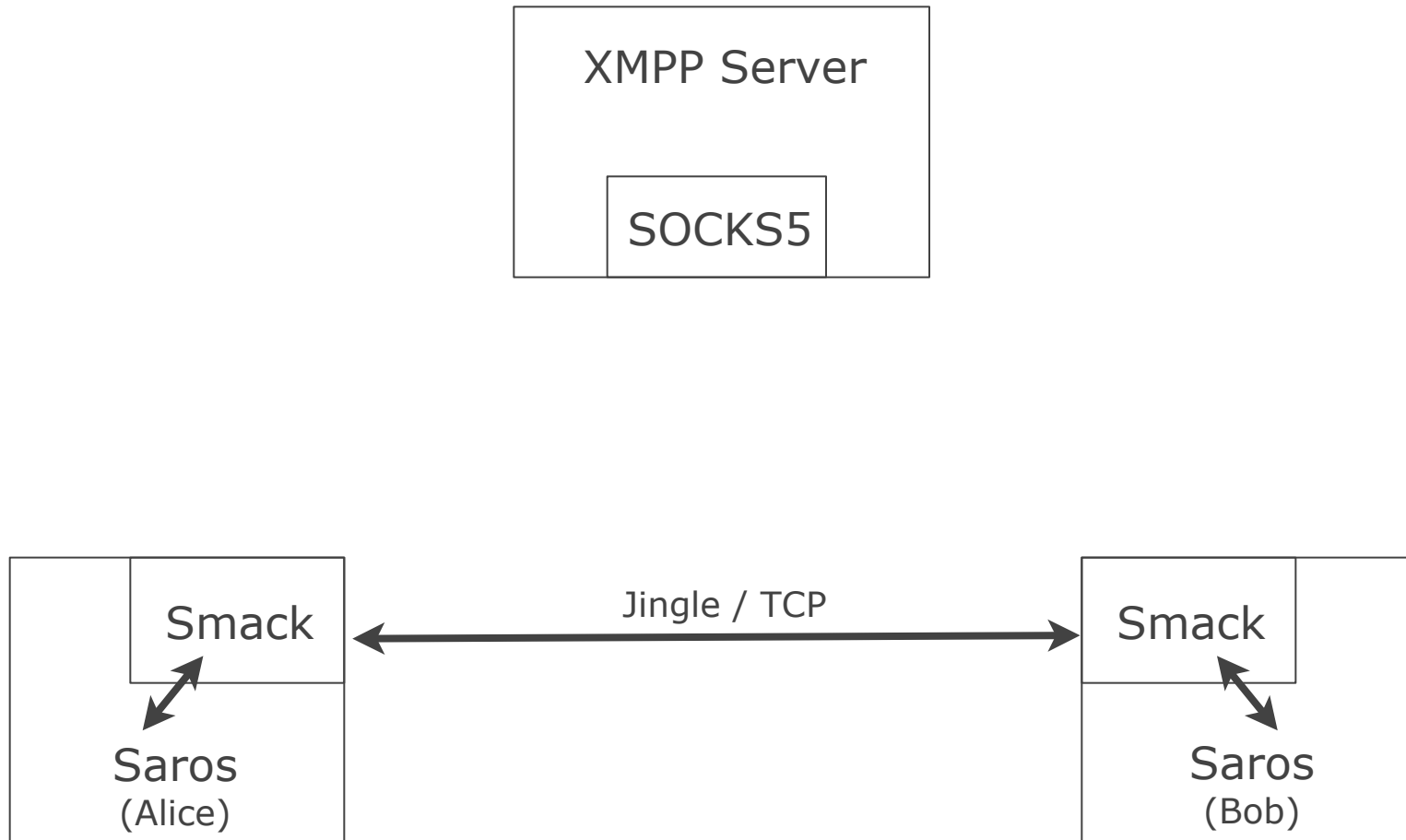
Ausgangssituation

- Saros verwendet XMPP-Bibliothek Smack
- Probleme in Saros die auf Smack zurückzuführen sind
 - Pakete zum Hinzufügen und Entfernen von Kontakten können verloren gehen
 - Saros Projektsynchronisation
 - über Jingle
 - keine Unterstützung für verbindungsorientiertes Protokoll (TCP)
 - fehlerhafte Ermittlung von IP-Adressen
 - lange Initialisierungsphase ($\sim 20s$)
 - FileTransfer
 - kein Stream möglich (nur Datenaustausch mit festgelegter Länge)
 - nicht bidirektional
 - basiert auf SOCKS5 Bytestream und InBand Bytestream die beides der Spezifikation nach unterstützen
 - keine separate Schnittstelle für SOCKS5 Bytestream und InBand Bytestream
 - schlechtes Timeout-Management führt zu Verbindungsabbruchen

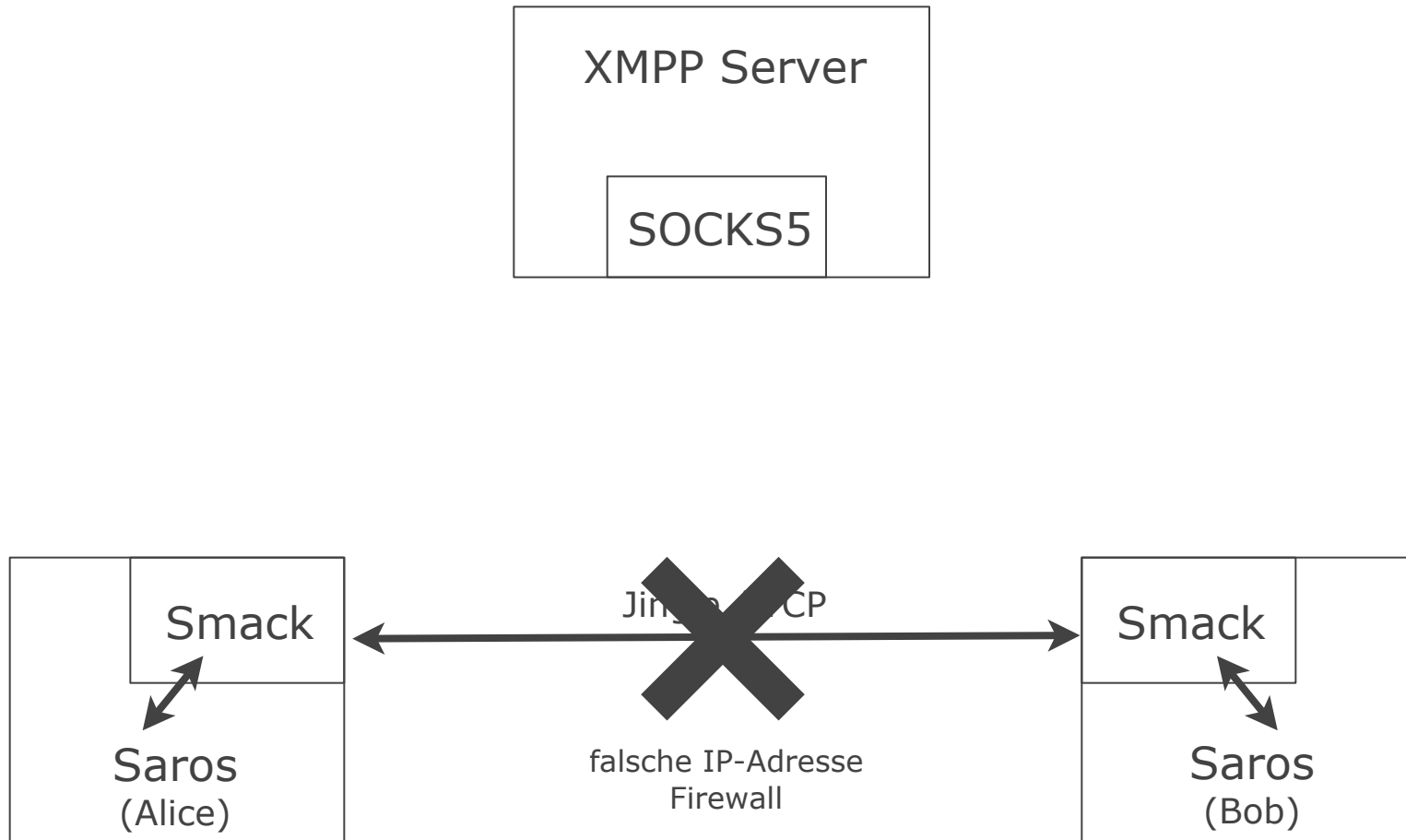
Einschub Probleme Jingle



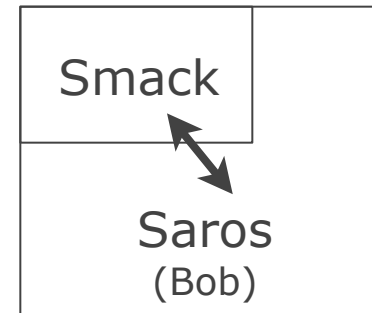
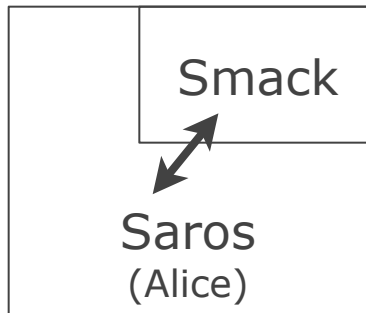
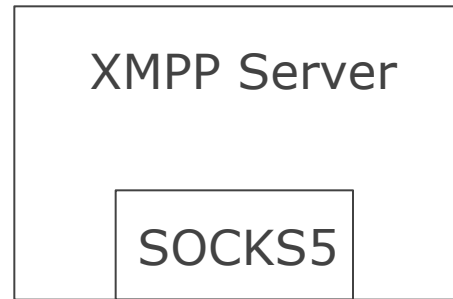
Einschub Probleme Jingle



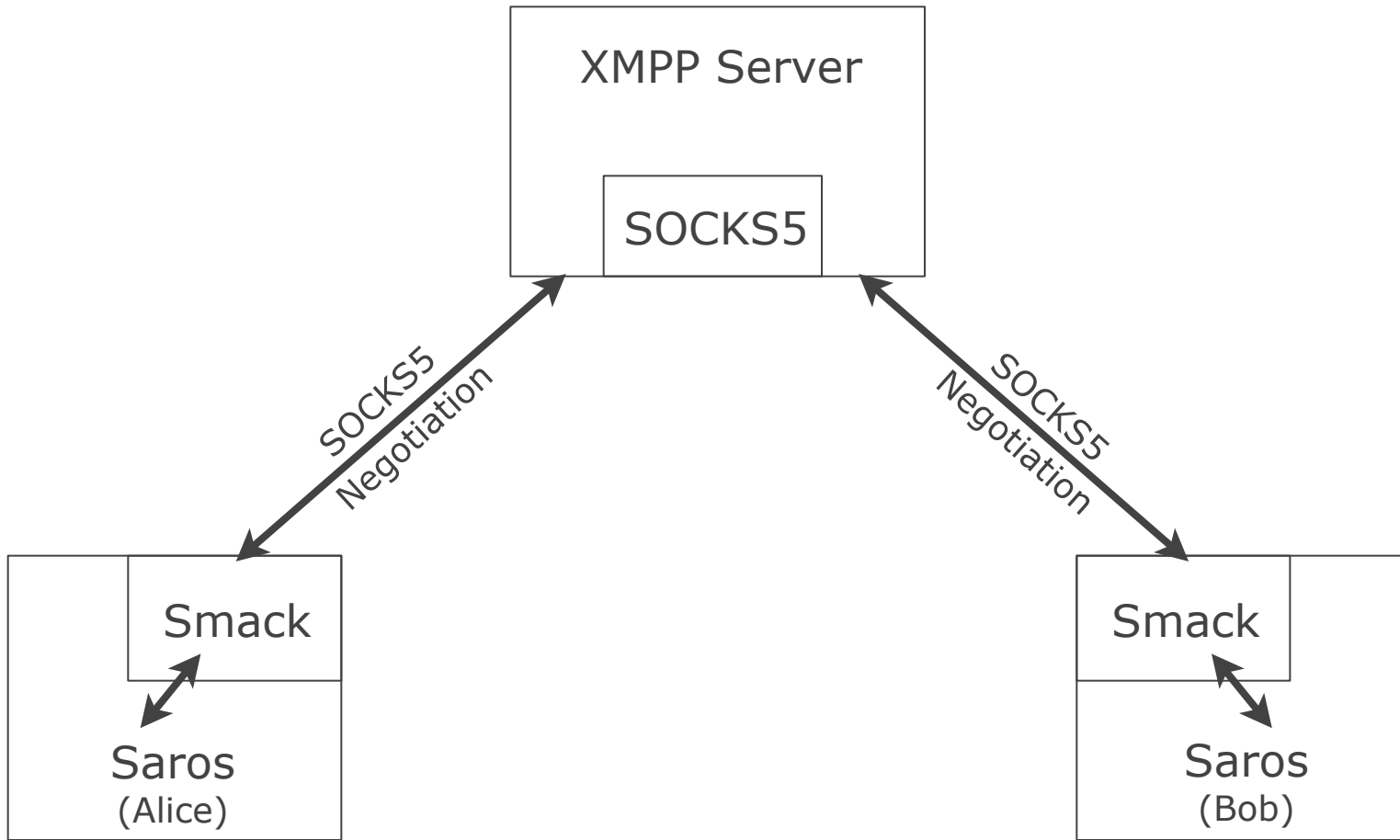
Einschub Probleme Jingle



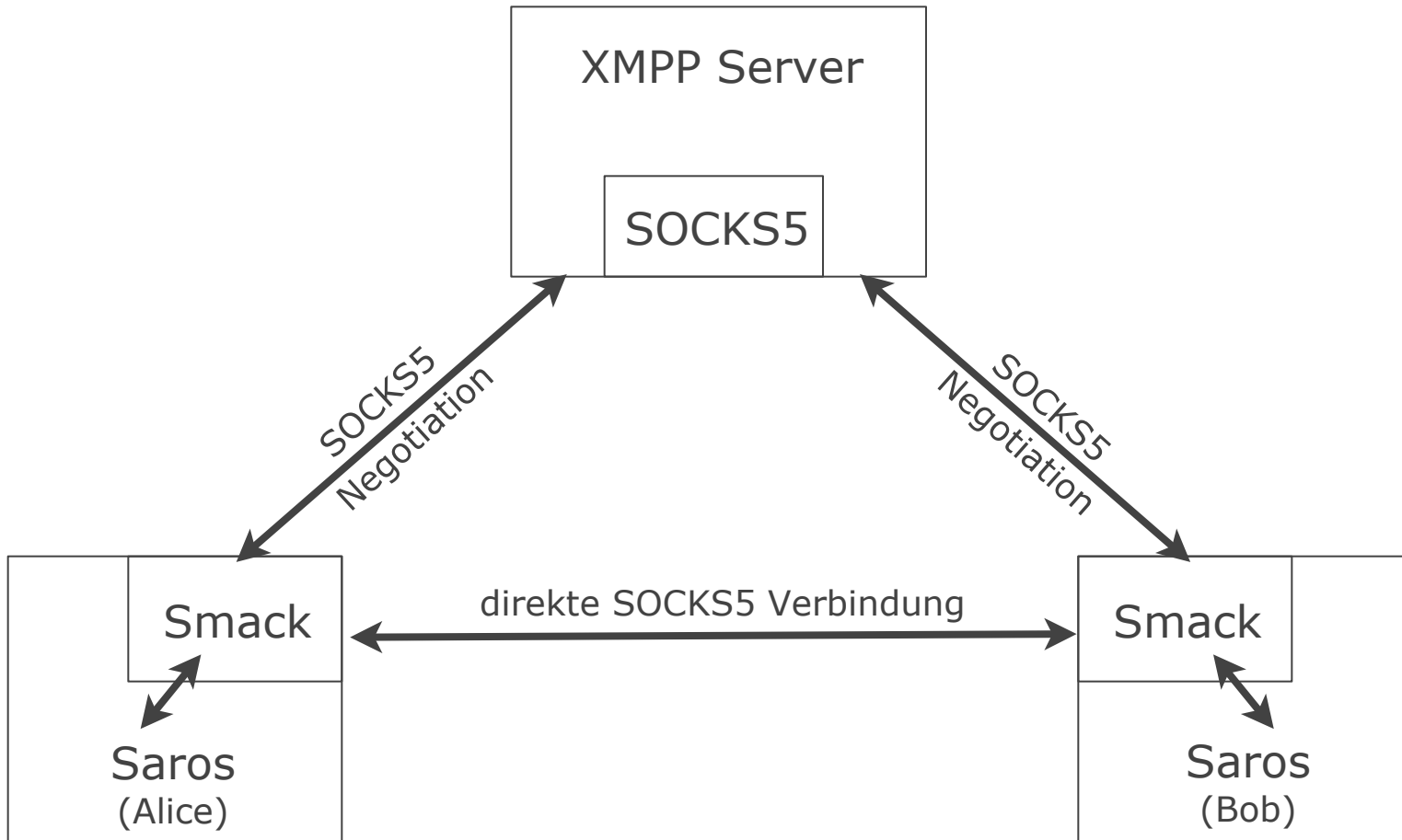
Einschub Probleme Jingle



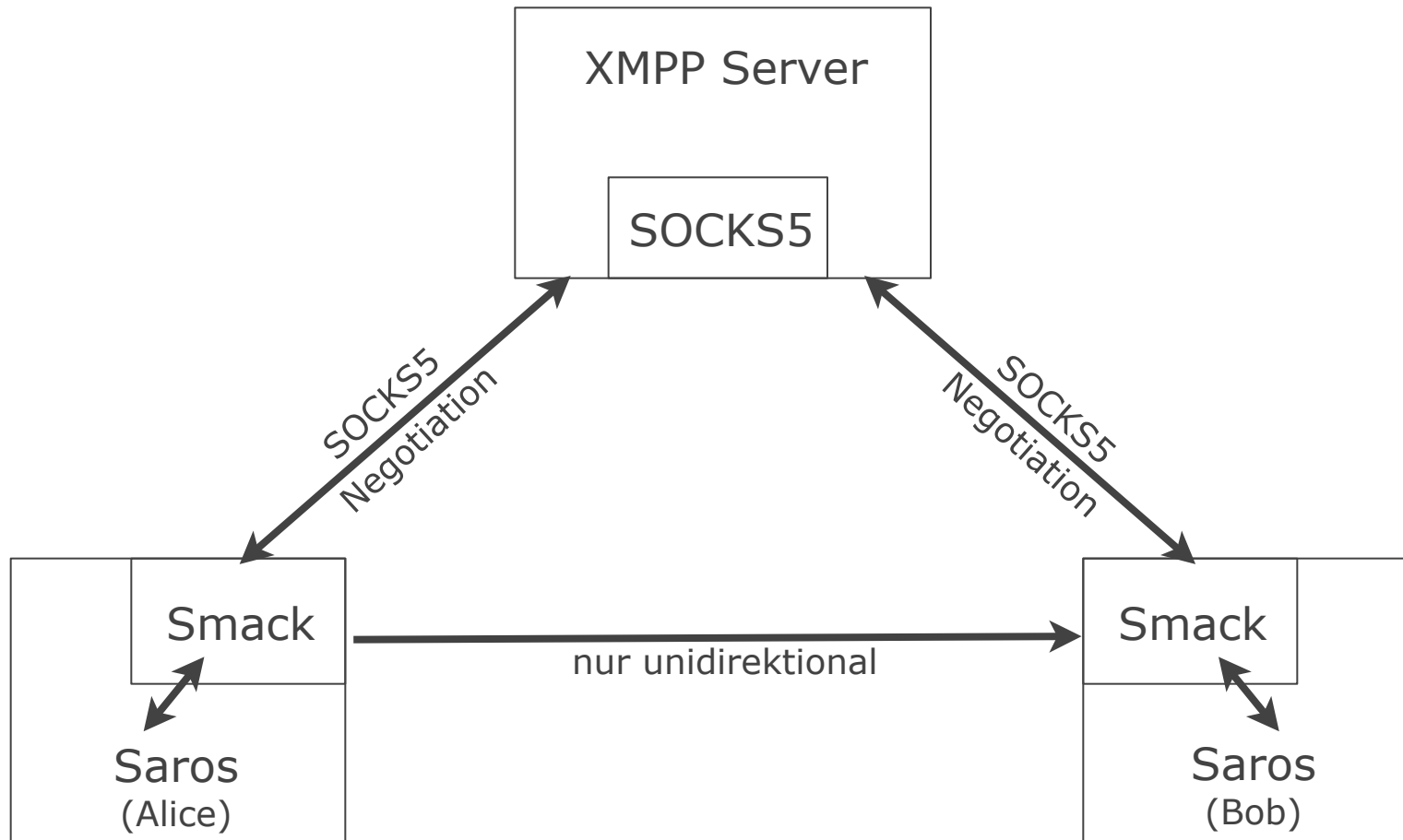
Einschub Probleme SOCKS5



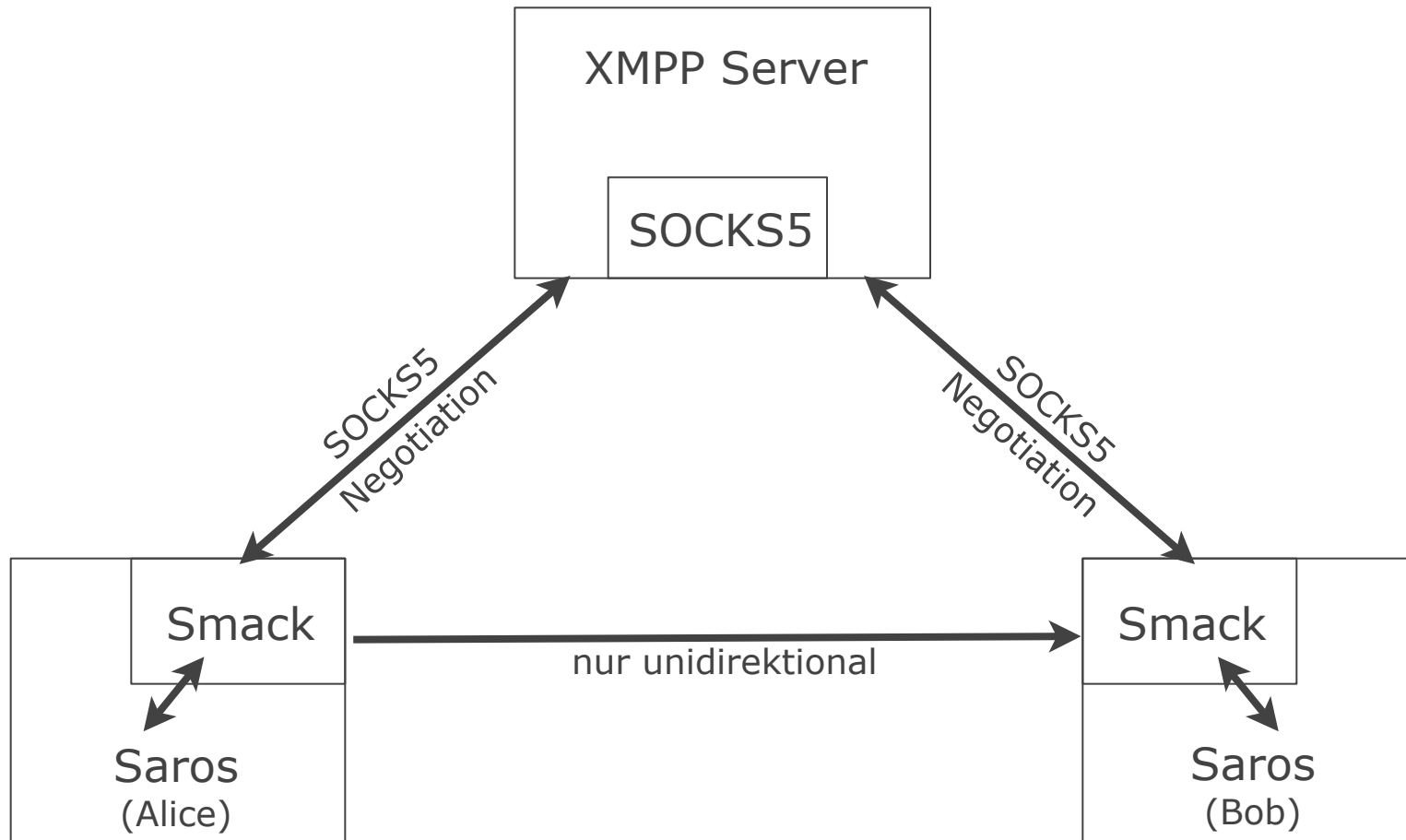
Einschub Probleme SOCKS5



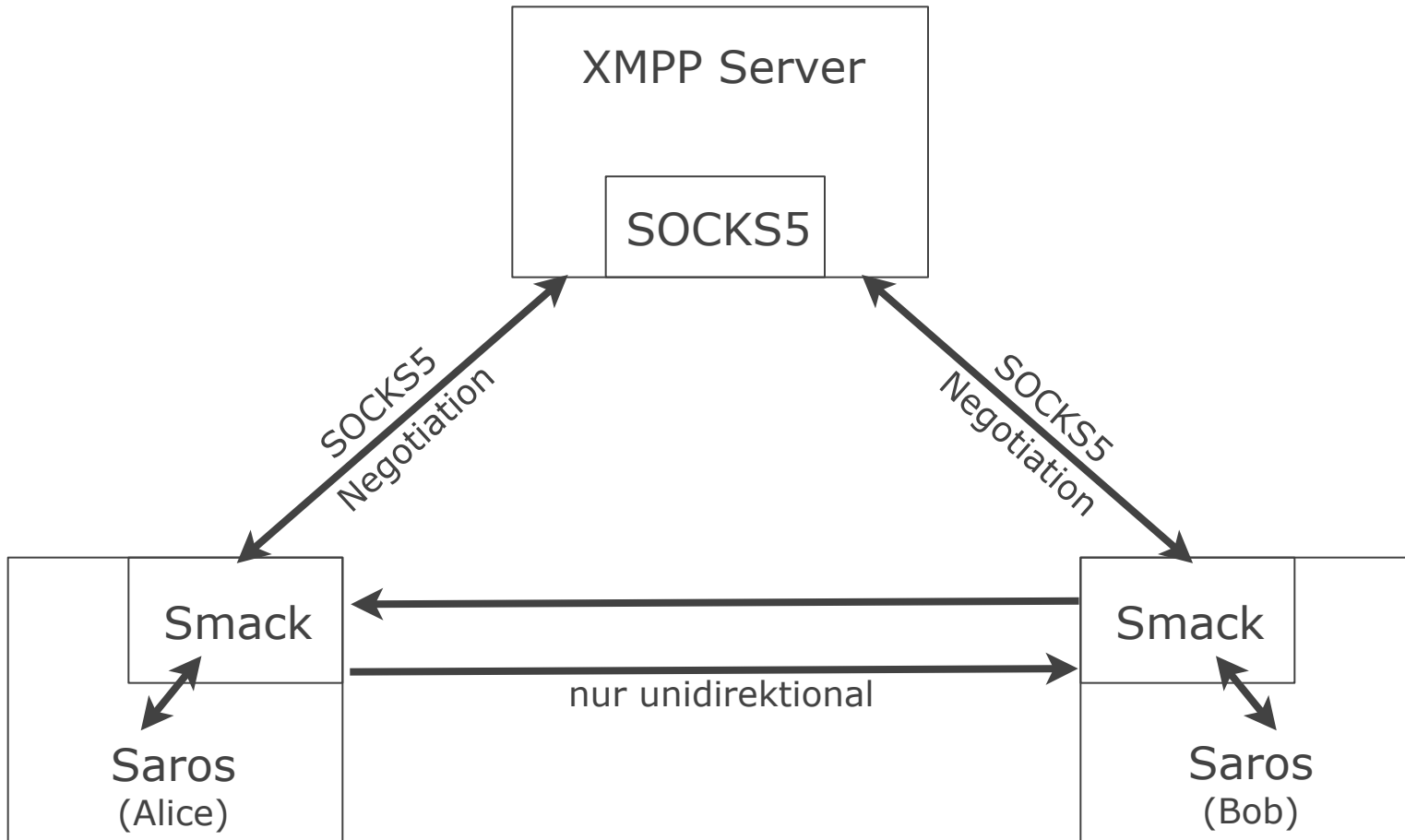
Einschub Probleme SOCKS5



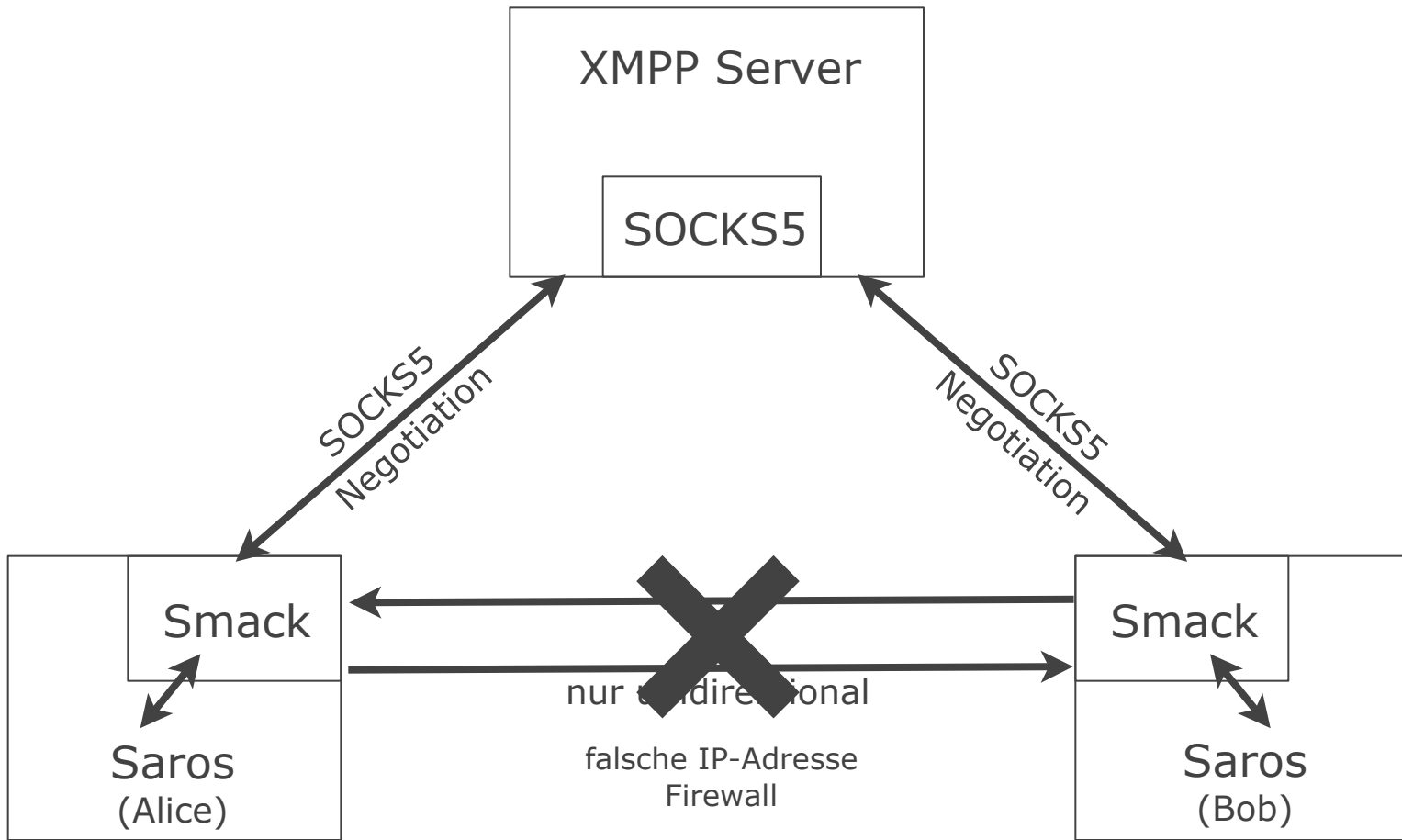
Einschub Probleme SOCKS5



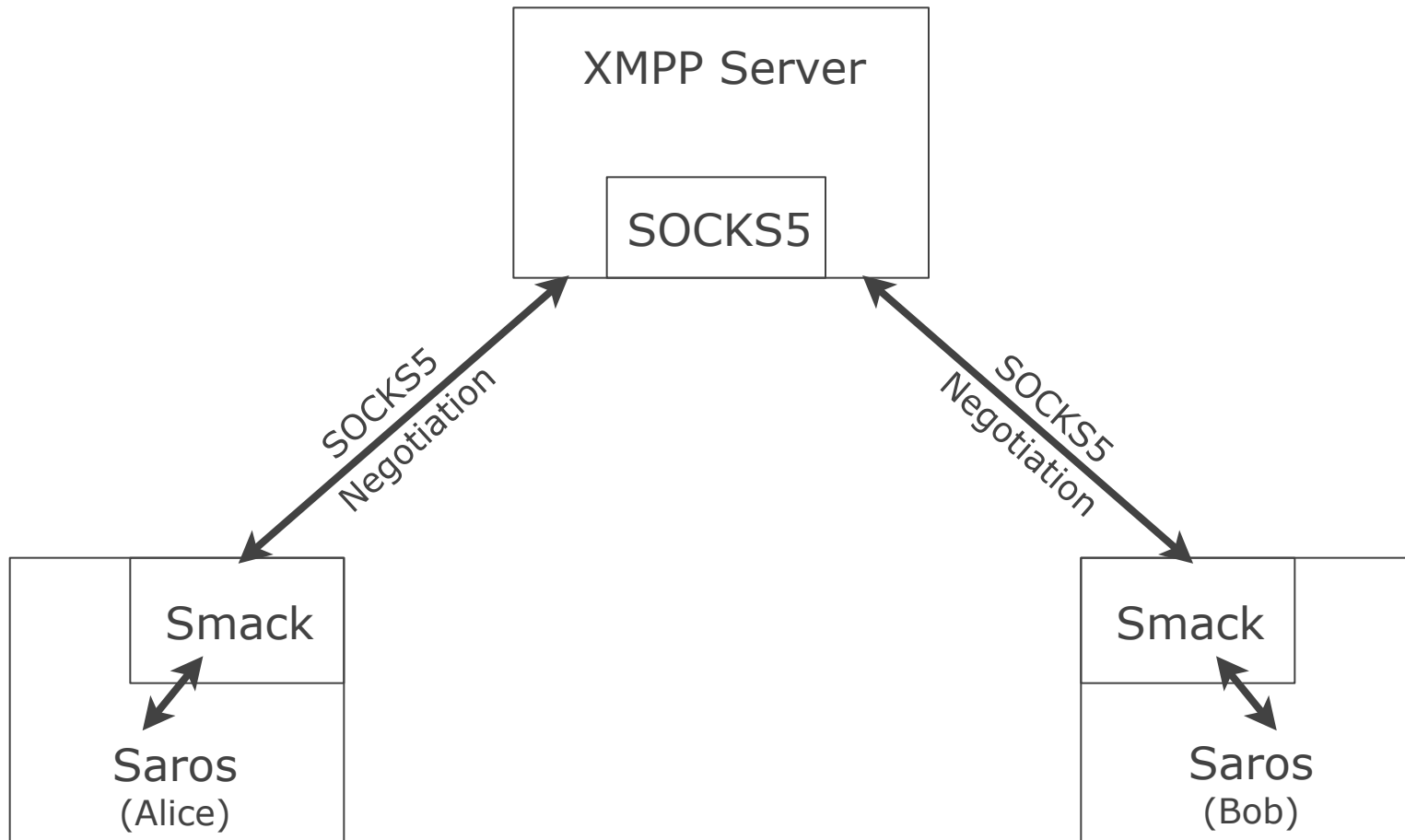
Einschub Probleme SOCKS5



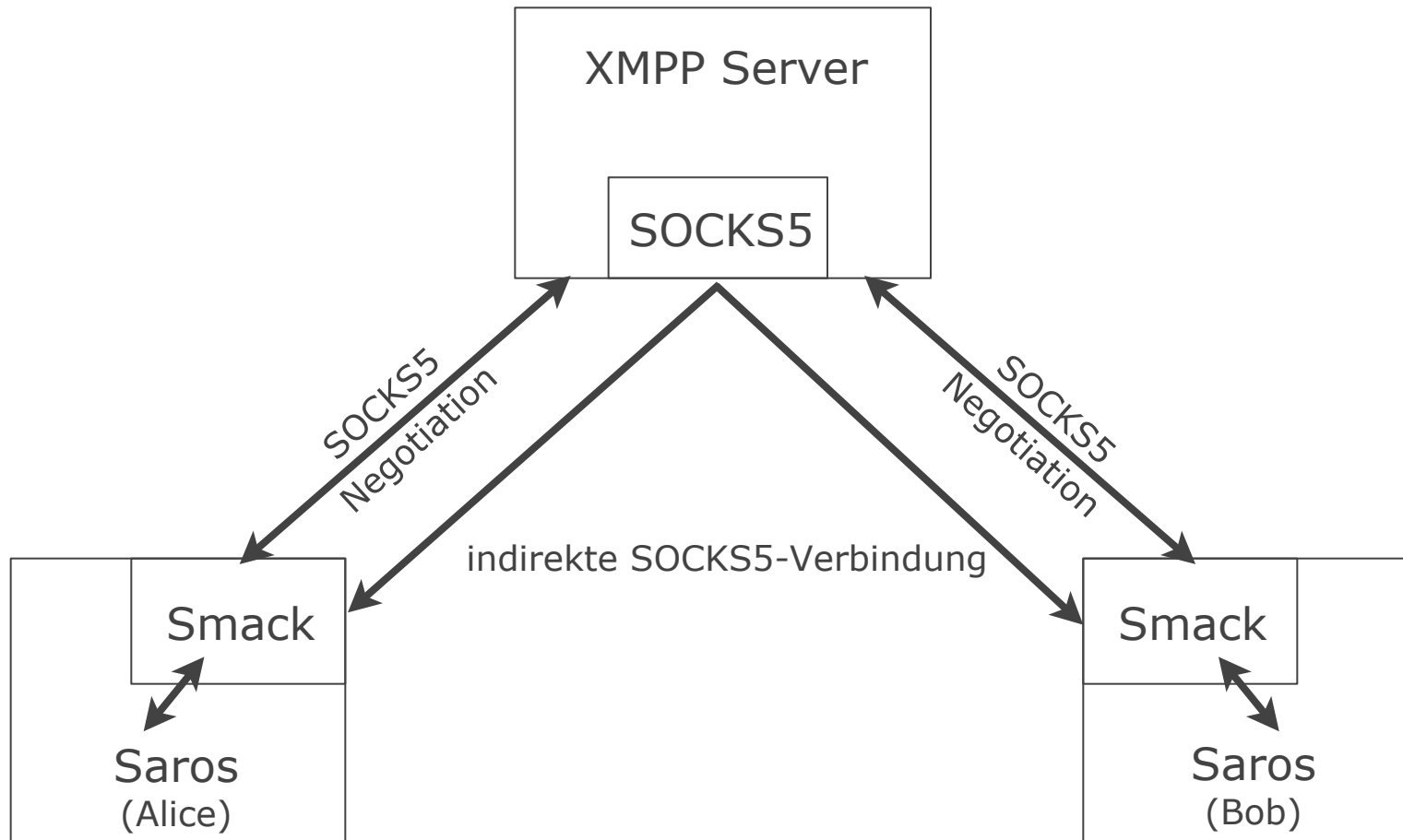
Einschub Probleme SOCKS5



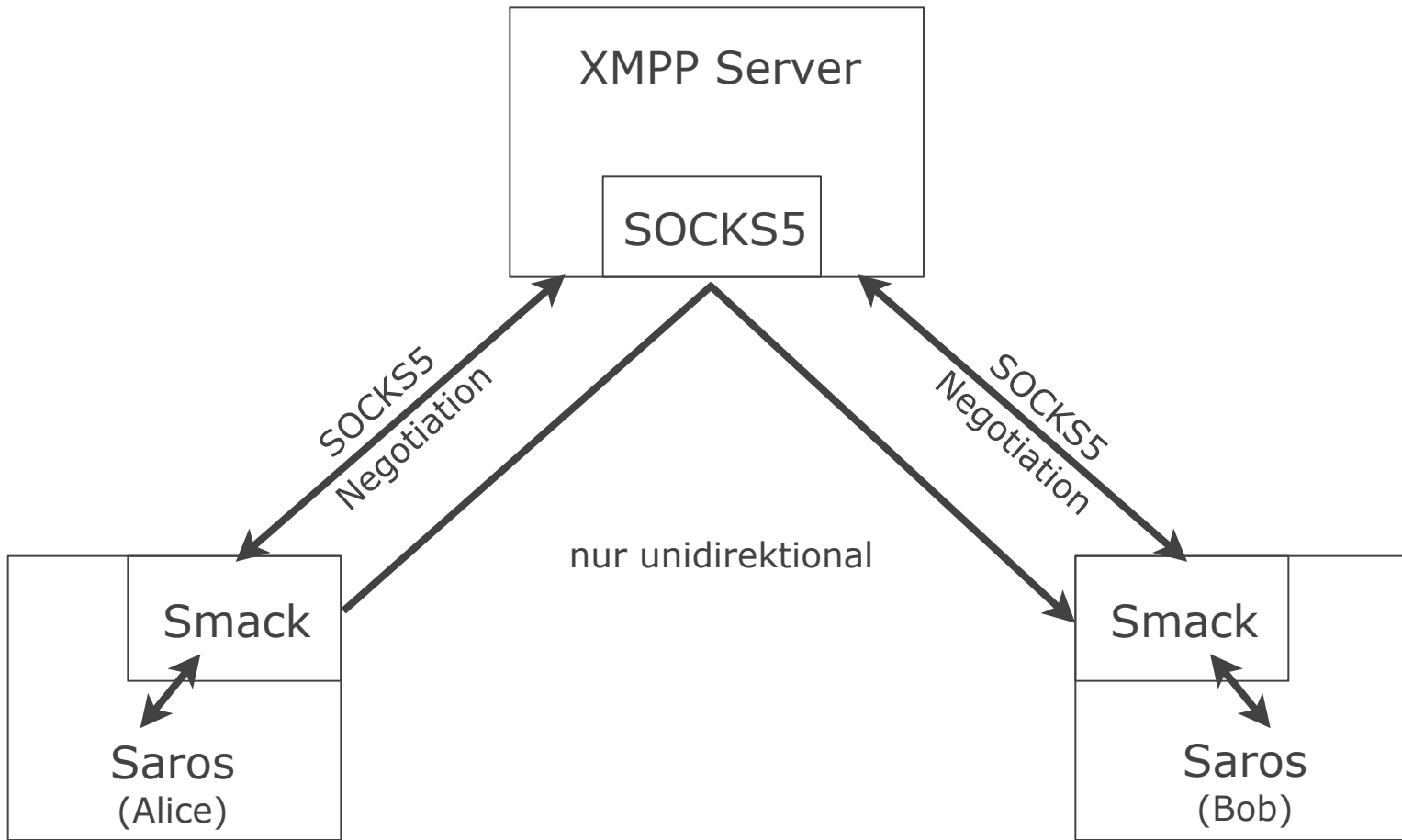
Einschub Probleme SOCKS5



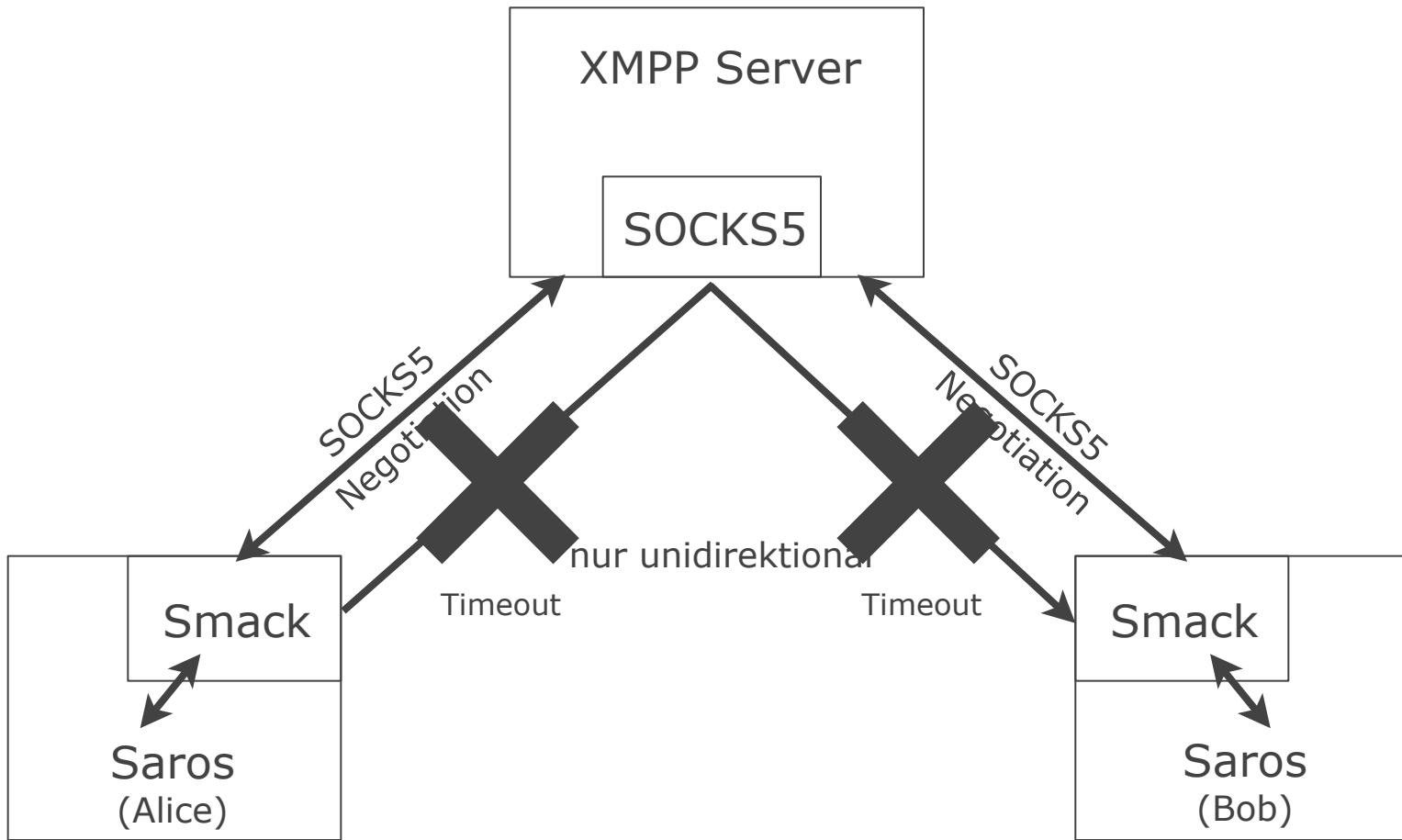
Einschub Probleme SOCKS5



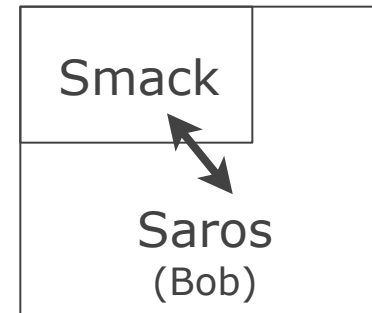
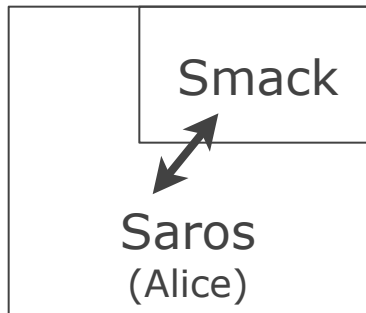
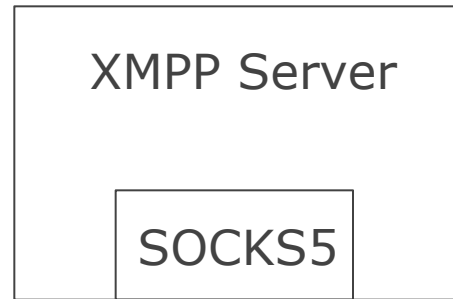
Einschub Probleme SOCKS5



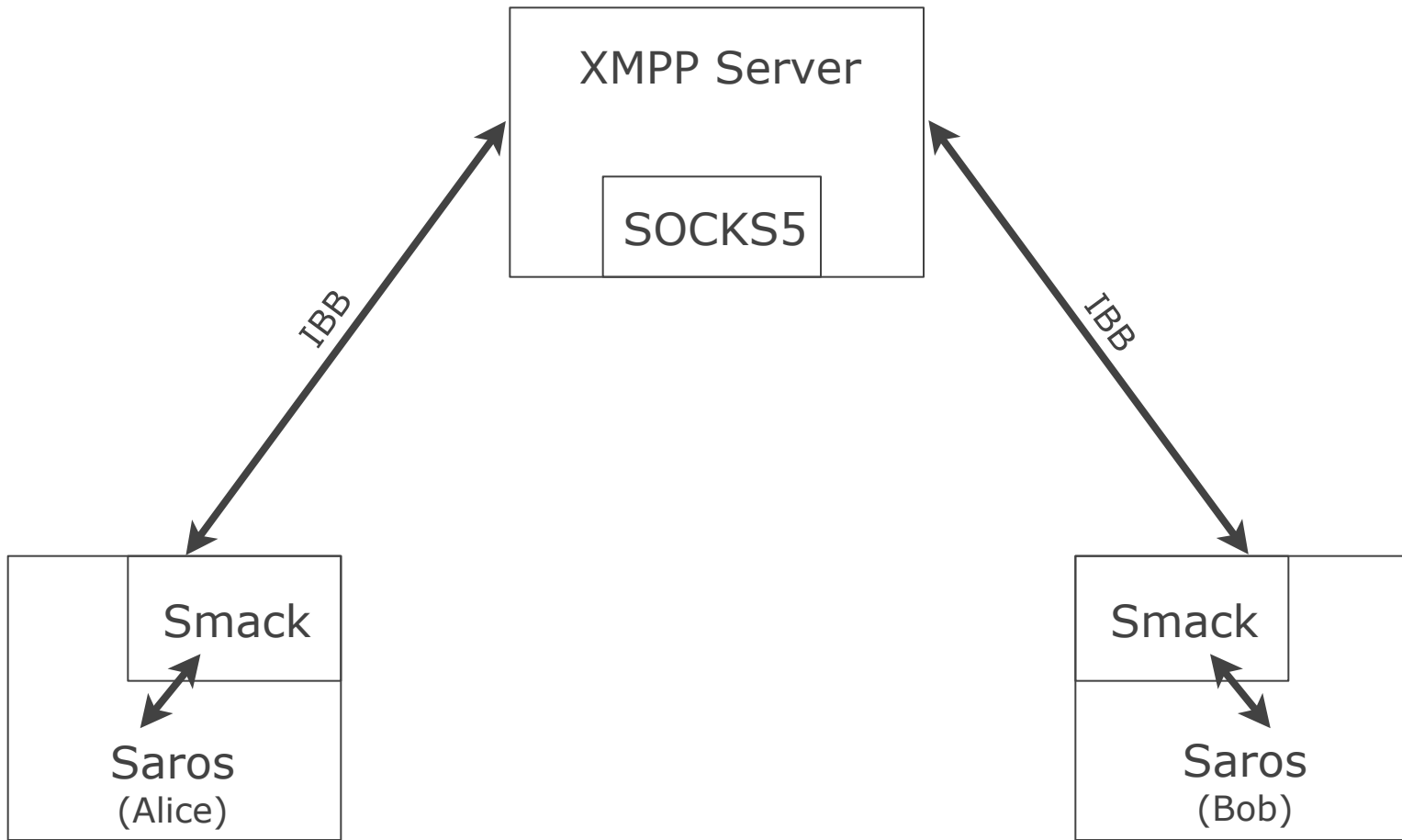
Einschub Probleme SOCKS5



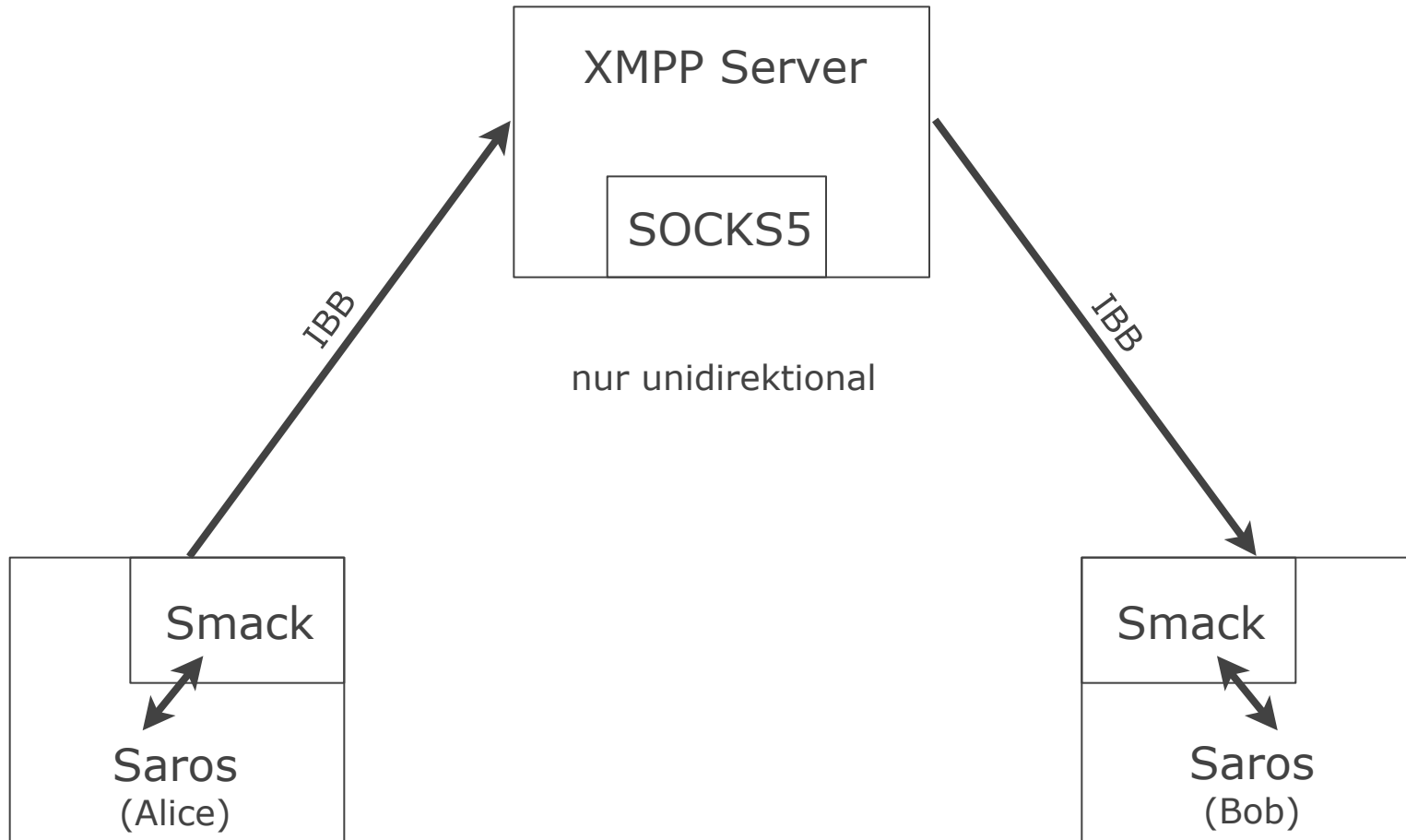
Einschub Probleme IBB



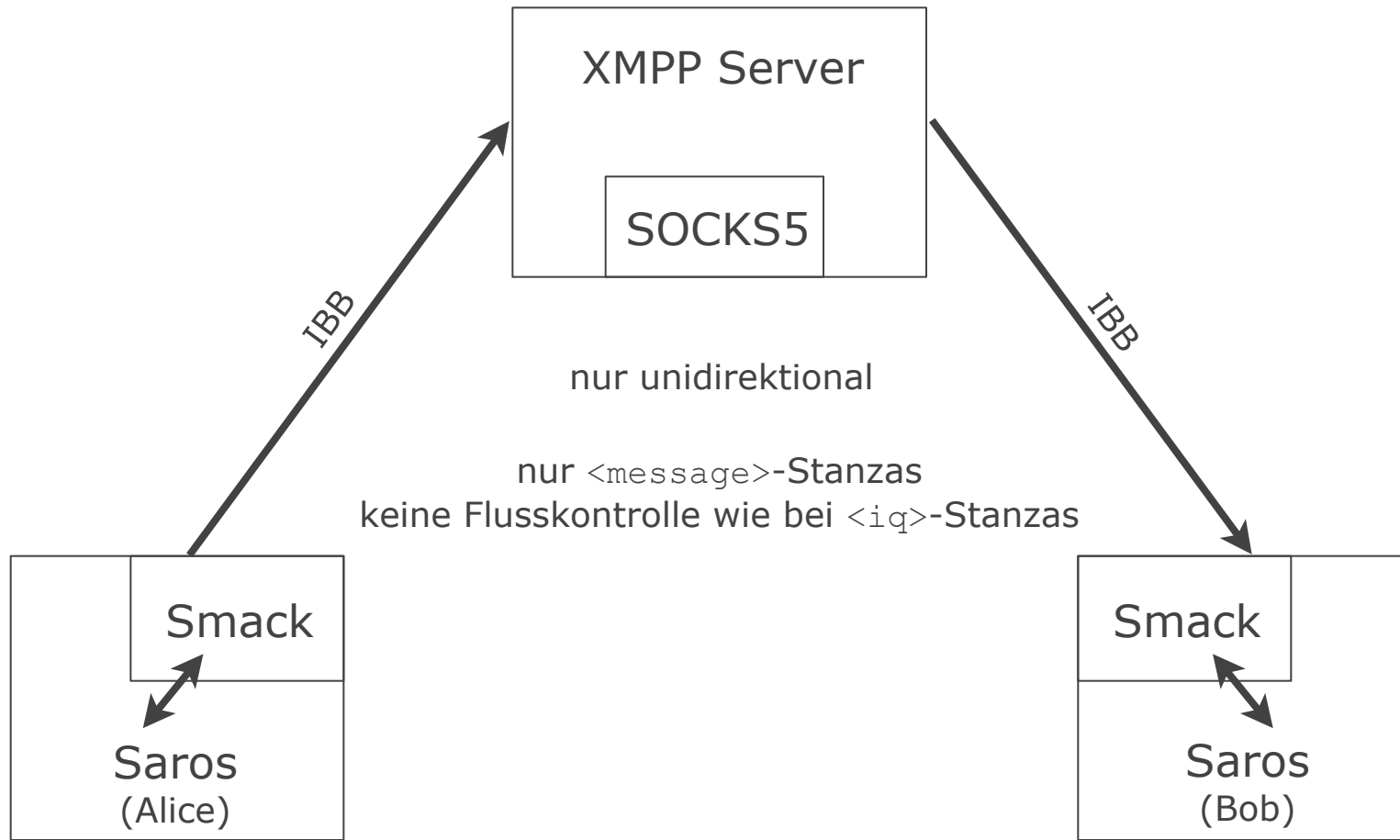
Einschub Probleme IBB



Einschub Probleme IBB



Einschub Probleme IBB



Ausgangssituation

- Probleme beim Open Source Projekt Smack
 - letztes offizielles Release 2008
 - zunächst hauptsächlich von Jive Software Entwicklern vorangetrieben, seit 2008 der Community überlassen
 - wenig Aktivität der bestehenden Entwickler
 - bestehende Patches der Community werden ignoriert
 - restriktiver Zugang zur Entwicklergruppe
 - teilweise kein Wissen / Verantwortlichkeit für bestimmte Komponenten
- über 70 offene Probleme im Bugtracker
- Spezifikation bereits implementierter Komponenten hat sich geändert
 - dadurch veraltete bzw. fehlerhafte Implementierungen
- mangelnde Code-Dokumentation
- inkonsistente Strukturierung / Formatierung
- verhältnismäßig Unit-Tests mit Abhängigkeit zu einem XMPP-Server

Aufgabenstellung

- Verbesserung Smack
 - SOCKS5 Bytestream
 - Roster
 - Evaluation Jingle Startup
 - aufgrund der Komplexität und des Zeitrahmens verworfen
 - stattdessen IBB
 - InBand Bytestream
- Smack-Community
 - Wie kann die Community eines durch eine Firma kontrollierten Open-Source-Projektes reaktiviert werden?
 - Welche Erfolgchancen hat eine Abspaltung des Projektes?
 - Wie kann die Qualität der Entwicklerbeiträge einer Open-Source-Community gesteigert werden?
- Mitarbeit in Saros
 - Releases
 - Tests
 - Bugfixes
 - Reviews

Lösungsansätze

- Entwicklung von Patches für Smack
 - zielgerichtet auf Probleme von Saros
 - genügen höheren Qualitätskriterien als bestehende die ignoriert wurden
 - Beachtung des Kontextes eines Problems
 - Ursache des Problems lösen, nicht Symptome
 - Unit-Test um zukünftige Fehler und Regressionen zu vermeiden
- Aufnahme in die Smack-Entwicklergruppe
- Impulse zur Reaktivierung der Community geben

SOCKS5 Bytestream Patch

- Extrahieren der Schnittstelle für SOCKS5 Bytestream
 - bietet Funktionalität über die Anwendung zum Dateitransfer hinaus
- Verbesserung der Funktionalität
 - basierend auf aktuellem Stand der XEP-0065 Spezifikation
 - Konfigurierbarkeit
 - lokale IP-Adressen
 - Port
 - Verfügbarkeit des lokalen Proxy
 - Timeouts
 - Usability der Schnittstelle
- Unit-Tests
 - testen der Funktionalität
 - Verbesserung der Wartbarkeit
 - einfacheres Einpflegen von Änderungen
 - Erstellung der Test-Schnittstelle `Protocol`
 - unabhängig von XMPP-Server-Implementierung
 - erlaubt Tests anhand der Protokollspezifikationen
- Verbesserungsvorschläge für Spezifikation (XEP-0065)

Einschub Protocol

@Test

```
public void shouldNegotiateSocks5BytestreamAndTransferData() throws Exception {
```

```
// get Socks5BytestreamManager for connection
```

```
Socks5BytestreamManager byteStreamManager = Socks5BytestreamManager.getBytestreamManager(connection);
```

```
// create responses in the order they should be queried according to XEP-0065
```

```
// build discover info that supports the SOCKS5 feature
```

```
DiscoverInfo discoverInfo = createDiscoverInfo(targetJID, initiatorJID);
```

```
discoverInfo.addFeature(Socks5BytestreamManager.NAMESPACE);
```

```
// return that SOCKS5 is supported if target is queried
```

```
protocol.addResponse(discoverInfo, Verification.correspondingSenderReceiver,  
    Verification.requestTypeGET);
```

```
...
```

```
// build used stream host response
```

```
Bytestream streamHostUsedPacket = createBytestreamResponse(targetJID, initiatorJID);
```

```
streamHostUsedPacket.setSessionID(sessionID);
```

```
streamHostUsedPacket.setUsedHost(proxyJID);
```

Einschub Protocol

```
// return used stream host info as response to the bytestream initiation
protocol.addResponse(streamHostUsedPacket, new Verification<Bytestream, Bytestream>() {

    public void verify(Bytestream request, Bytestream response) {
        assertEquals(response.getSessionID(), request.getSessionID());
        assertEquals(1, request.getStreamHosts().size());
        StreamHost streamHost = request.getStreamHost();
        assertEquals(response.getUsedHost().getJID(), streamHost.getJID());
    }

}, Verification.correspondingSenderReceiver, Verification.requestTypeSET);

...

// call the method that should be tested
BytestreamSession session = byteStreamManager.establishSession(targetJID, sessionID);

...

// verify protocol flow
protocol.verifyAll();
}
```

weitere Patches

- InBand Bytestream Implementierung mit extrahierter Schnittstelle analog zum SOCKS5 Bytestream Patch
- Frühzeitiges Registrieren des RosterListeners
- RosterListener nur bei Änderungen informieren

Behebung seit langem bestehender Probleme in Smack

- Verbesserung des Delayed Delivery Parser
- Verbesserung der Robustheit des Message-Parsers und I18N
- I18N für Message Subjects

Auswertung

- Reaktion der Smack-Entwickler auf die Patches
 - Nutzerkonto für Bugtracker
 - Beginn von Code-Durchsichten
 - später Aufnahme in die nicht-öffentliche Gruppe „Community Planning“
- Code-Durchsicht für SOCKS5 Bytestream Patch wurde nicht abgeschlossen
- Patch für RosterListener durchgesehen, aber dennoch nicht dem SVN hinzugefügt
- weitere Patches führten zu keinen Reaktionen und auch keinen weiteren Code-Durchsichten
- die beiden aktiven Entwickler waren anderweitig beschäftigt

Auswertung

- Reaktivierung der Community
 - Analyse bestehender Abspaltungen (Forks)
 - ein bestehender Fork seit Juli 2009 nicht mehr weiterentwickelt
 - neuer Fork vom Februar 2010 kaum Aktivität und keine neuen Entwickler
 - Erstellung eines Dokumentes mit Richtlinien für Smack-Entwickler
 - enthält Hinweise zu
 - Code-Formatierungen
 - der Package-Struktur
 - Qualitätskriterien für Patches
 - bewährten Entwicklungsmethoden
 - Erstellen von Unit-Tests
 - Erzeugung von Patch-Dateien
 - wurde mit Smack-Entwicklern abgestimmt aber nicht veröffentlicht
 - Aufnahme neuer Mitglieder durch Mentorship-Programm geregelt
 - seit Oktober 2009 keine weiteren Entwickler hinzugekommen

Auswertung

- Entwicklungen seit Juni 2010
 - Umstellung der Plattform-Software von Ignite Realtime
 - Verschiebung nicht-öffentlicher Dokumente in die öffentliche Gruppe „Contributors“
 - dadurch alle Dokumente unfreiwillig veröffentlicht
 - Erstellung eines Vorschlags für eine Roadmap zur Weiterentwicklung
 - wenig Reaktionen
 - keine Diskussion und Weiterentwicklung der Roadmap
 - im August 2010 Zugriff auf SVN
 - alle entwickelten Patches eingecheckt
 - alle damit verbundenen Bugtracker-Einträge geschlossen

Auswertung

- Verbesserungen in Saros
 - Entwicklung von Patches für kleinere Bugfixes
 - Code-Durchsichten

 - Umbau des Netzwerkmoduls von Saros
 - Umstellung der Verwendung von Jingle/FileTransfer auf die Verwendung von SOCKS5 Bytestream / InBand Bytestream
 - Zuverlässigkeit beim Herstellen von Verbindungen erhöht
 - Geschwindigkeit der Verbindungsherstellung erhöht
 - Reduzierung der Code-Komplexität

- Verbesserungen an der SOCKS5 Bytestream Spezifikation
 - gefundene und gemeldete Inkonsistenzen und Fehler behoben



Diskussion

Danke