

# Diplomvortrag

Weiterentwicklung einer Eclipse-Erweiterung für  
verteilte Paar-Programmierung im Hinblick auf  
Kollaboration und Kommunikation

**Oliver Rieger**

Freie Universität Berlin, Institut für Informatik  
<https://www.inf.fu-berlin.de/w/SE/ThesisDPPIII>

- Einführung in Saros
- Anforderungen
- P2P-Kommunikation
- Schreib-Parallelität
- Ausblick

- Verteilte Paar-Programmierung
  - Klassischer Paar-Programmierung
    - Besserer Entwurf, Wissenaustausch, Fortlaufende Kontrolle, ...
    - Erfordert räumliche Nähe und passende Arbeitsplätze
  - Ermöglicht:
    - Verteilte Teams
    - Individuelle Arbeitsplätze
    - Mehr Parallelität
- Werkzeug-Unterstützung für Verteilte Paar-Programmierung

- Ansatz des Collaboration Awareness
  - Kollaboration direkt im Werkzeug integriert
  - Eclipse-Erweiterung Saros
- Awareness-Informationen
  - Text-Selektionen
  - Hervorhebung der Änderungen
- Parallelität auf Sicht-Ebene
  - Unabhängiges Sichten der Projekt-Dateien
  - Verfolger-Modus
- Übersetzbarkeit

- Replikations-Ansatz
  - voller Zugriff auf alle Projekt-Dateien
  - ähnliche Projekte bei zwei Teilnehmern
- Extensible Messaging and Presence Protocol (XMPP)
  - dezentraler Server-basierter Architektur (Jabber)
  - Offener Standard
  - XML-Nachrichten
- Im Verlauf des Projekts DPP II
  - Erweiterung von max 2 auf beliebige viele Teilnehmer
  - ein exklusiver Driver

- Replikation des gemeinsamen Projekts
  - vorausgehende Synchronisation aller Projekt-Dateien
  - XMPP-basierte Übertragung
    - begrenzte Übertragungsgrößen
    - hohe Latenzzeit und Versand über dezentralen Serververbund
- Parallelität auf Schreibebeine
  - exklusives Schreibrecht unter vielen Teilnehmer geteilt
  - bisher nur auf Programm- und Sichtebeine
- Kommunikation
  - zentrale Nachrichtenverteilung
    - Jabber-Conference-Room
  - P2P-Verbindung

- P2P Verbindung in Weiterverkehrsnetzen
  - keine global eindeutigen IP-Adressen
  - Kapselung in private Netzwerke mit globaler IP-Adresse
  - Network Address Translation (NAT)
    - Translation der IP-Adressen
    - Umschreiben der Adressen in den IP-Paketen
- NAT-Traversal Problematik
  - fehlende Standardisierung
  - NAT-Typen mit unterschiedlichem Translations-verhalten
    - *Full Cone NAT, (Port) Restricted Cone NAT, Symmetric NAT*
    - verhindern Aufbau einer Direktverbindung

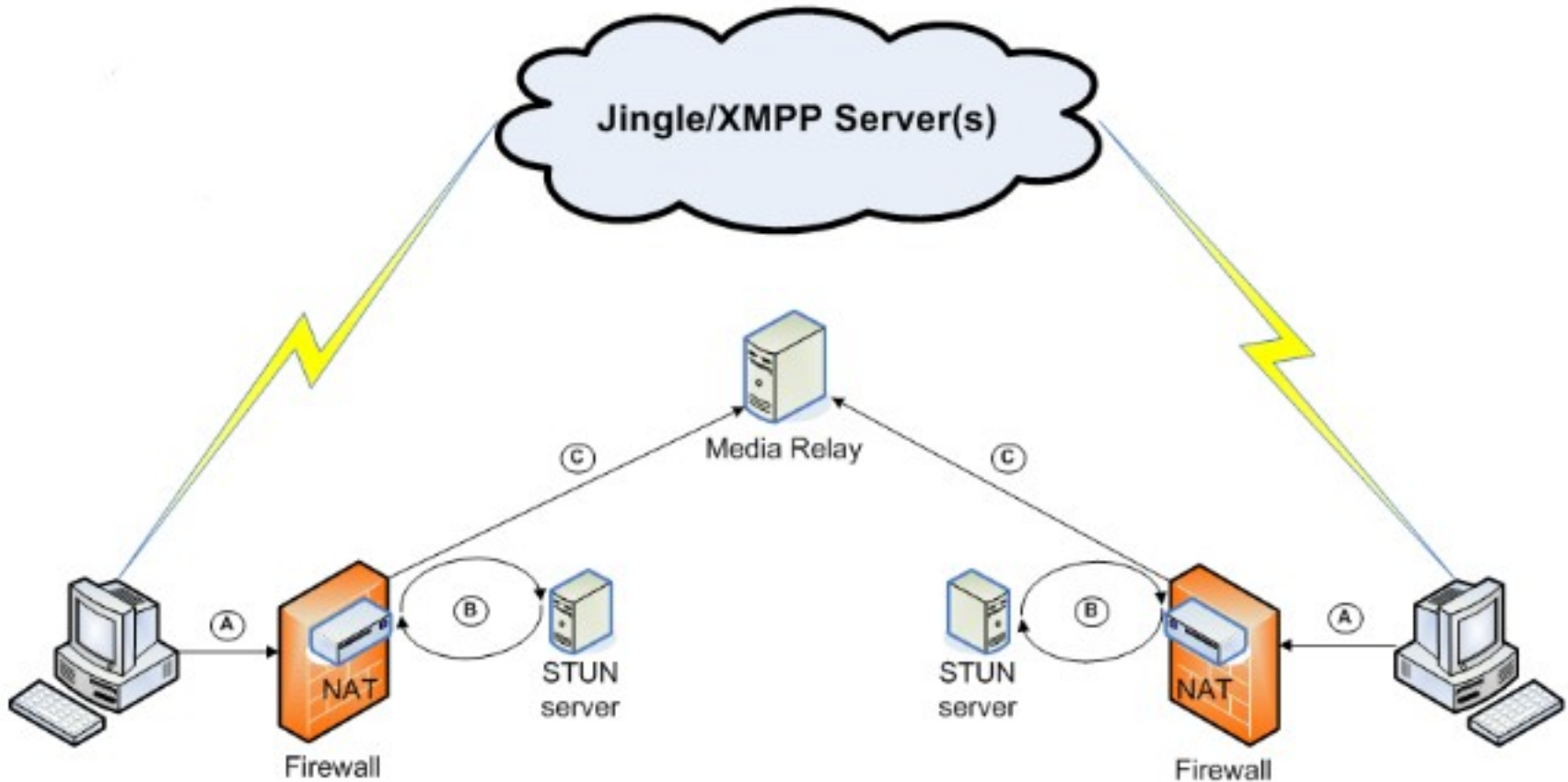
- NAT-Traversal Lösungskonzepte
  - Routerkonfiguration
    - statisches Port-Forwarding
    - Universal Plug-and-Play (UPnP)
  - Simple traversal of UDP through NAT's (STUN)
    - Bestimmung von NAT-Typen, IP-Adressen, Ports
    - nicht für alle Netztopologien geeignet
  - Hole Punching
    - basierend auf STUN Verfahren
    - Öffnen der Verbindungen für P2P Kommunikation
  - Traversal Using Relay NAT (TURN)
    - Datenrelay
    - stellt öffentlichen Netzwerk-Adresse
    - emuliert Restricted Cone NAT



# NAT-Traversal Lösungskonzepte (2)

- Interactive Connectivity Establishment (ICE)
  - universell einsetzbares NAT-Traversal Verfahren
  - Kombination aus STUN und TURN Verfahren
  - Einsatz in VoIP-Software (Skype, Google Talk)
- Jingle
  - Entwicklung in Kooperation mit Google
  - universeller Verbindungsaufbau mit STUN und ICE Verfahren
  - XMPP-Protokoll-Erweiterung für Multi-Media-Dienste
  - Verwaltung über XMPP
- Smack 3.0.x
  - Implementierung der gegenwärtigen XMPP-Protokolle
  - modularer Aufbau und Schnittstellen zur Erweiterung

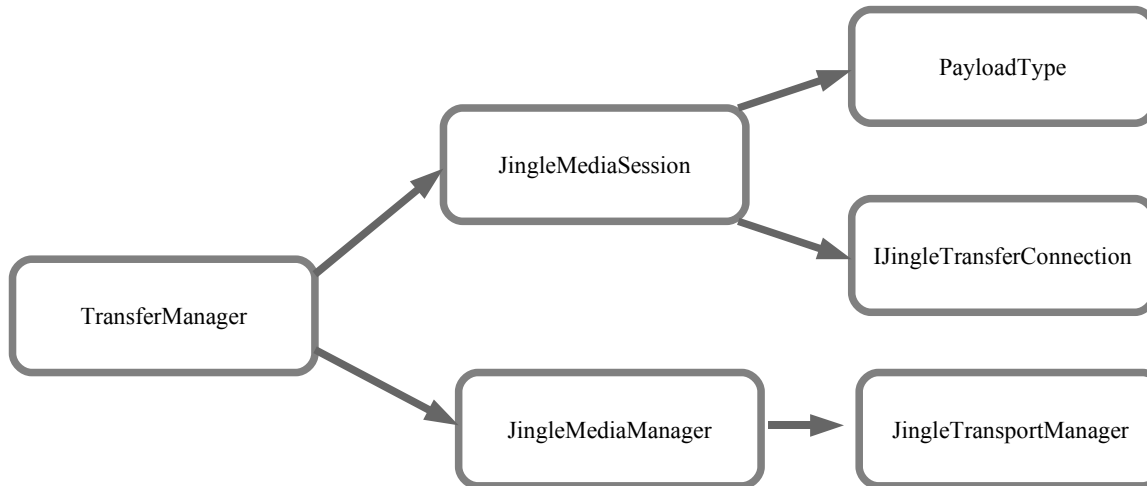
# Jingle P2P-Verbindung



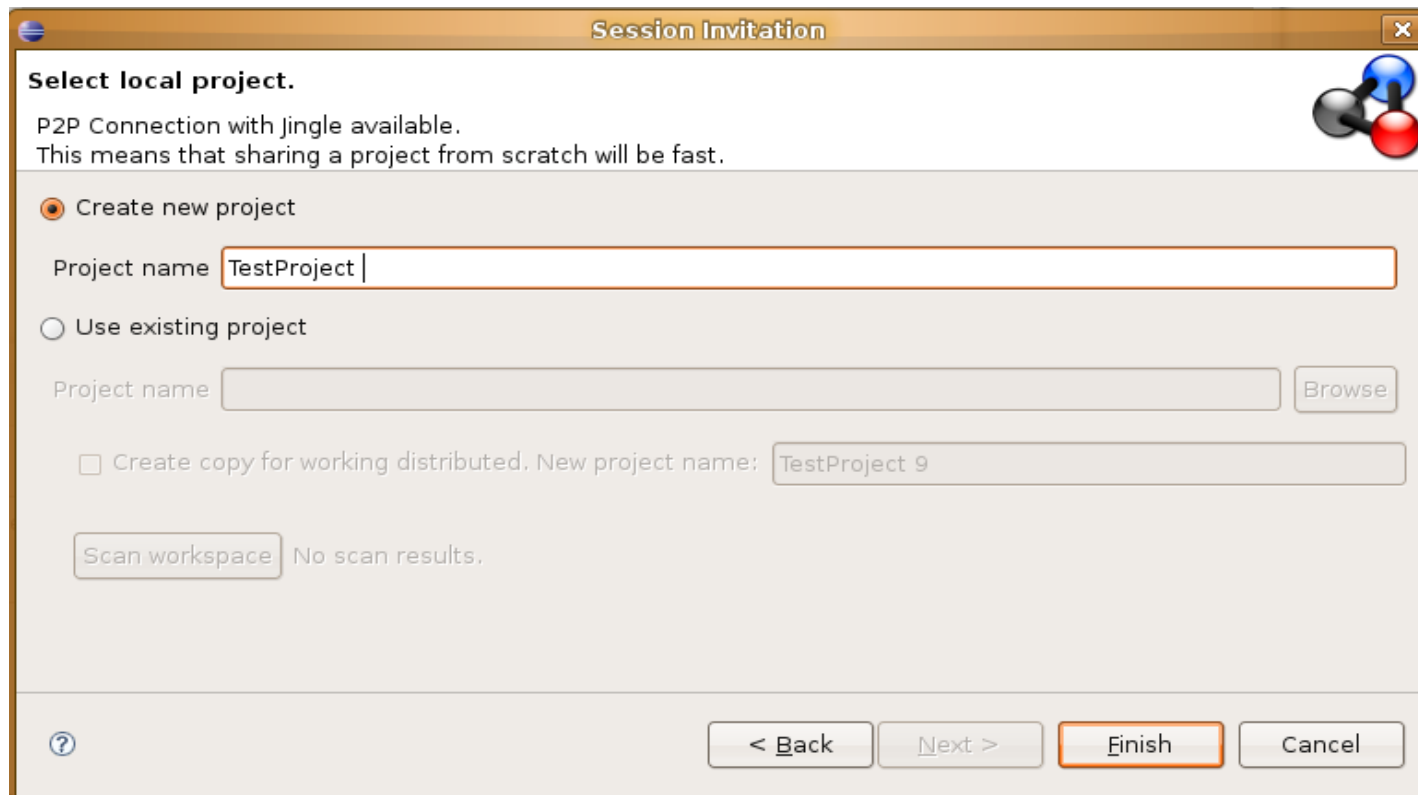
- (A) Lokale IP-Adressen (P2P)
- (B) Globale IP-Adressen (P2P mediated by NAT)
- (C) Media Relay

# Integration von Jingle

- Versionswechsel der Smack API
- Integration der Jingle-Erweiterung bietet Grundlage P2P Kommunikation
- Multi-Media-Erweiterung für Datei-Übertragung auf Basis von Jingle
  - auf dem ICE-Verfahren basierender Transport-Manager
  - TCP-Übertragungskanal für bidirektionales Senden



- Adaption des Replikations-Ansatzes
  - Aushandeln der Verbindungsart
  - XMPP-basierte Dateiübertragung als Ausweichprotokoll



- Rechnergestützte Gruppenarbeit
  - Anforderungen
    - Echtzeit
    - Verteilt
    - Uneingeschränkt
- Nebenläufigkeitskontrollverfahren
  - Koordination der Aktionen konkurrierender Prozesse
  - Wahrung der Konsistenz
  - Sperrverfahren
    - Pessimistischen Sperrverfahren
    - Optimistischen Sperrverfahren
  - Operational Transformation (OT)
    - Konsistenzerhaltung durch Konfliktauflösung
    - Transformation von Konfliktoperationen zur Konsistenzerhaltung

- Operational Transformation
  - für eng gekoppelte Gruppenarbeit
    - Zugriff auf kleine Bausteine bzw. einzelne Zeichen und Wörter
    - Änderungen werden sofort lokal ausgeführt (*Echtzeit-Anforderung*)
    - Kommunikation der Änderungs-Operationen (*Verteilt-Anforderung*)
    - blockierungsfreies Arbeiten, da Konsistenz beim Empfang entfernter Operation überprüft wird (*Uneingeschränkt-Anforderung*)
  - Aktuelle Forschungsarbeiten zum Themengebiet CSCW
  - Echtzeit-Gruppeneditor im Vergleich
    - Bsp: IRIS, SubEthaEdit, ECF, ACE
    - zur kollaborativen synchronen Echtzeitbearbeitung
    - basieren auf Verfahren der Operation Transformation

- Divergenz
  - Ausführung der Operationen in verschiedener Reihenfolge
  - unterschiedliche Dokumentenzustände
- Kausalitätsverletzung
  - Operationsausführung abhängiger Operationen außerhalb ursprünglicher Reihenfolge
- Intentionsverletzung
  - Ausführung der Operation hat ein anderes Ergebnis, als bei der Erzeugung beabsichtigt
  - Bsp: Ausgang: ABCDE
    - $O_1 = \text{delete}(2, \text{CD}) \rightarrow \text{ABE} \rightarrow \text{A12BE}$
    - $O_2 = \text{insert}(1, 12) \rightarrow \text{A12BCDE} \rightarrow \text{A1CDE}$
  - auch durch Serialisierung nicht auflösbar

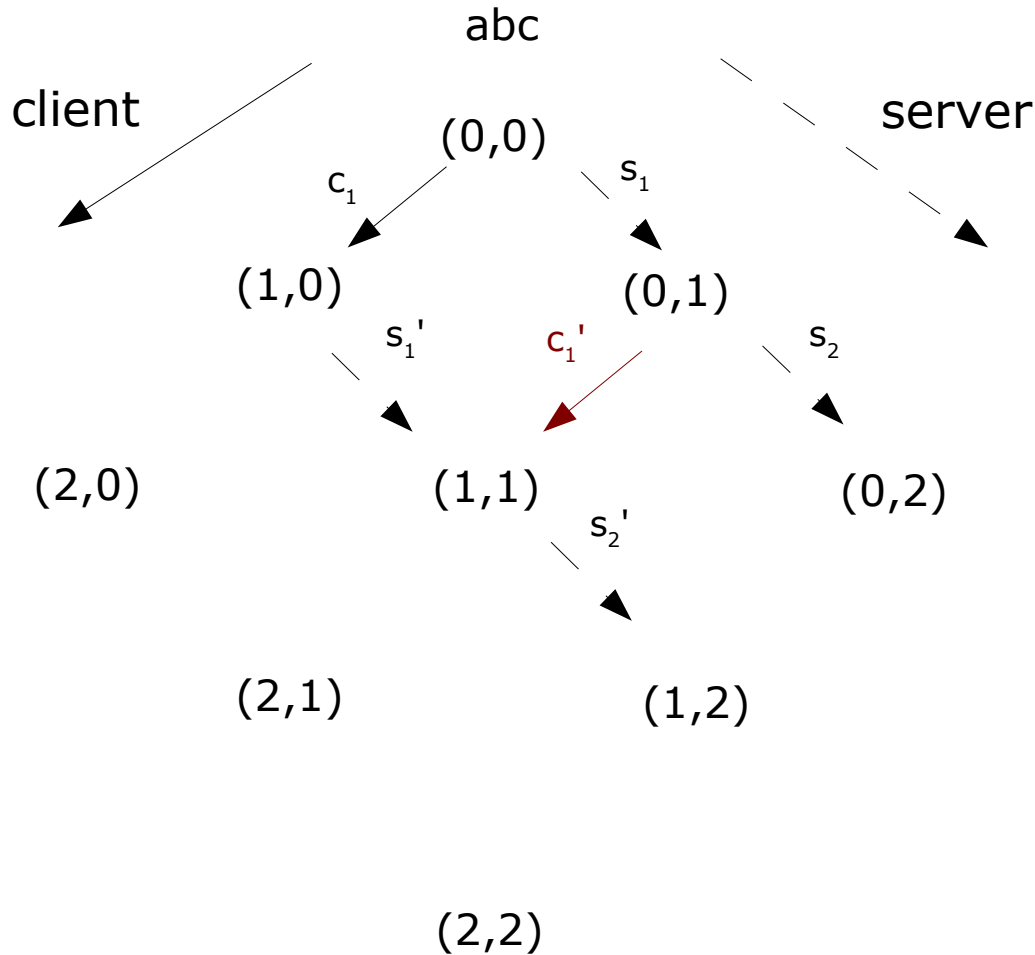
- Jupiter Algorithmus
  - 2-Wege Protokoll zur Sicherung der Konvergenz, Kausalität und Intention
  - zentraler Server zur Konsistenzsicherung
    - Adaption des 2-Wege Jupiter-Protokolls
    - geringer Kommunikationsaufwand
- 2-Wege Protokoll bestehend aus:
  - Transformationsfunktion
  - Zweidimensionaler Vektor-Raum
  - Nachrichtenwarteschlange
- Jupiter-Request für Kommunikation der Operationen



# Transformationsfunktion

- Inclusion Transformation
  - bindet den Effekt einer Operation in eine andere Operation ein
- Transformationsfunktion:
  - $xform(c,s) = c', s'$
  - gemeinsamer Ausgangszustand
  - gemeinsamer Endzustand
- Operationen
  - Zeichenketten beliebiger Länge
  - Einfügen, Löschen
  - zusätzlich: Leere Operation, Geteilte Operation
- Priorisierung der Server-Operationen

# 2-Wege Protokoll



$$s_1 = \text{ins}(1,b)$$

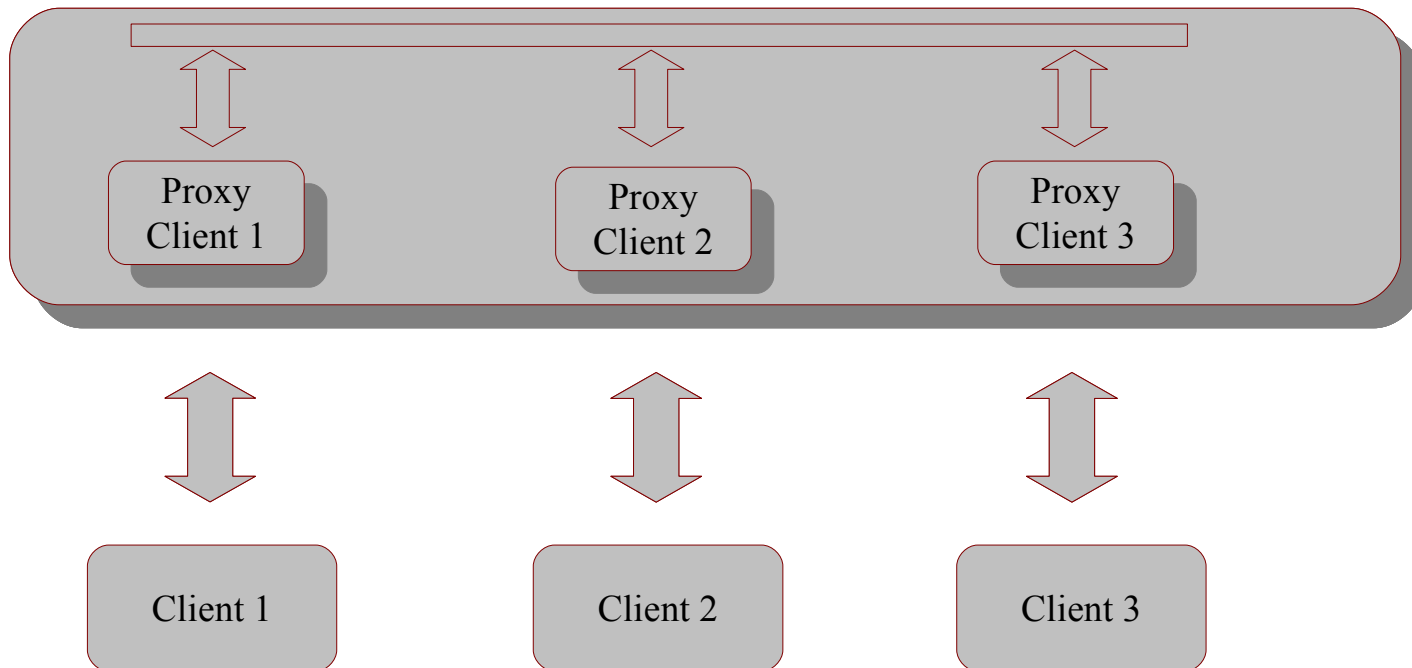
$$c_1 = \text{ins}(0,a)$$

$$s_1' = \text{ins}(2,b)$$

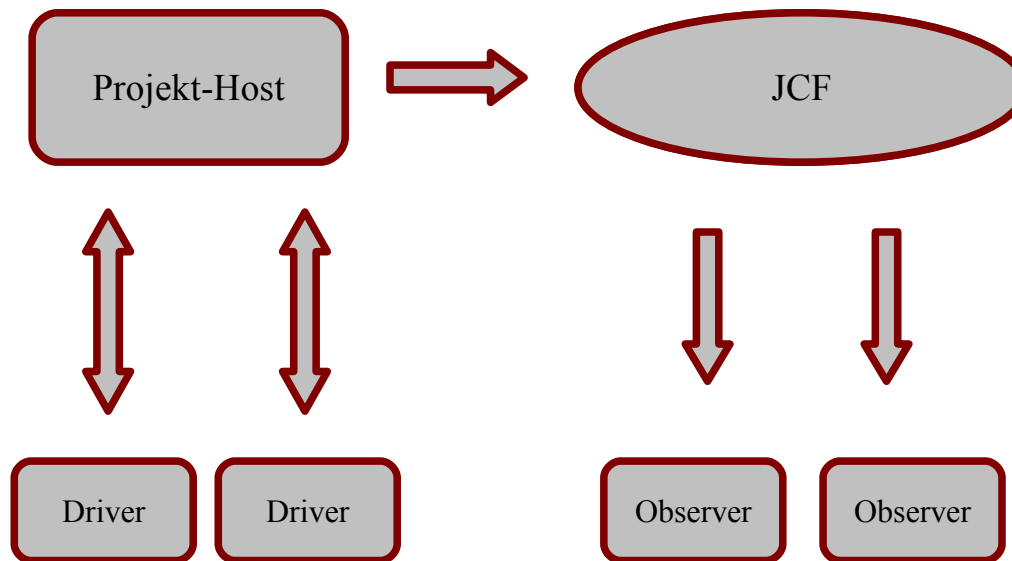
$$c_1' = \text{ins}(0,a)$$

# n-Wege Protokoll

- Erweiterung des 2-Wege zu einem n-Wege Protokoll mit zentraler Server-Instanz
  - Server setzt für jeden Client das 2-Wege Protokoll um
  - Proxy-Komponente Server-Seite des 2-Wege-Protokolls



- Projekt-Host als zentraler Jupiter-Server
  - Driver schicken Jupiter-Requests direkt an Projekt-Host
  - Jupiter-Requests werden im Jupiter-Server synchronisiert
  - Projekt-Host kommuniziert konfliktfreie Abfolge



## Integration in Saros (2)

- Jabber-Conference-Room
  - konfliktfreie Nachrichten
- Jupiter-Server-Instanz für konkurrierend geöffnete Dokumente
  - Hinzufügen / Entfernen von Proxies bei Rollenwechsel
- Driver-Verwaltung
  - Projekt-Host steuert Rollenwechsel
- Erweiterte Konsistenzprüfung
  - Überprüfung der Datei-Checksumme beim Speichern
  - Fehlerfall: Synchronisation der fehlerhaften Datei
- Exklusives Schreibrecht
  - bei Datei- / Ordner-Operationen

- Umstellung der gesamten Kommunikation auf P2P
  - Reduzierung der Latenzzeit für Übertragung durch XMPP
- Verfolger-Modus für beliebige Driver
  - nur Projekt-Host beobachtbar
- Exklusives Schreibrecht
  - exklusives Schreibrecht für beliebige Driver
- Nachverfolgung der Änderungsoperationen
  - Darstellung gelöschter Zeichenketten
- Multi-Media-Anwendungen

# Fragen

**Danke!**