

WebRTC Rückruffunktionalität

Johannes Hamfler

Hochschule für Telekommunikation Leipzig

johannes.hamfler@hftl.de

16. Februar 2015

Übersicht

Anforderungen

Problembetrachtung

Begriffsklärung

Anruf

Rückruf

Fehlerbehebung

Teillösung 1

Teillösung 2

Lösung

Zusatzaufgabe

Zusammenfassung

Anforderungen

Aufgabe

- ▶ Rückruffunktionalität ermöglichen
- ▶ Robustheit testen
- ▶ Zuweisung des korrekten `messagingStubs` beim Rückruf

Zusatzaufgabe

- ▶ Binden des `messagingStubs` an eine Conversation

Begriffsklärung

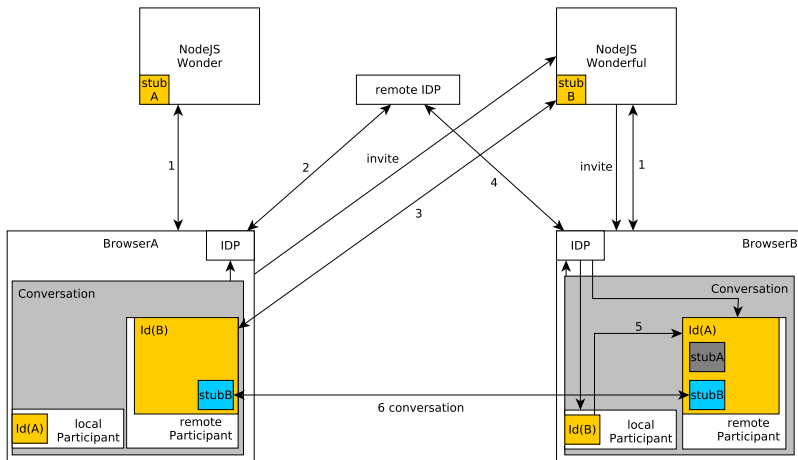
- ▶ Rückrufproblem tritt nur bei unterschiedlichen Domains / Messaging-Servern auf
- ▶ Zwei participants auf gleichem Server sind nicht betroffen
- ▶ `messagingStub` ist ein Objekt zum wrappen von `messagingStub`-Implementierungen

messagingStub Beispiel

```
▼ messagingStub: MessagingStub
  ▶ buffer: Array[0]
  ▶ impl: MessagingStub_NodeJS_1338
  ▶ listeners: Array[3]
  message: "stub downloaded from: http://127.0.0.1:80/stubs/tlabs/MessagingStub_NodeJS_1338.js"
  ▶ __proto__: MessagingStub
```

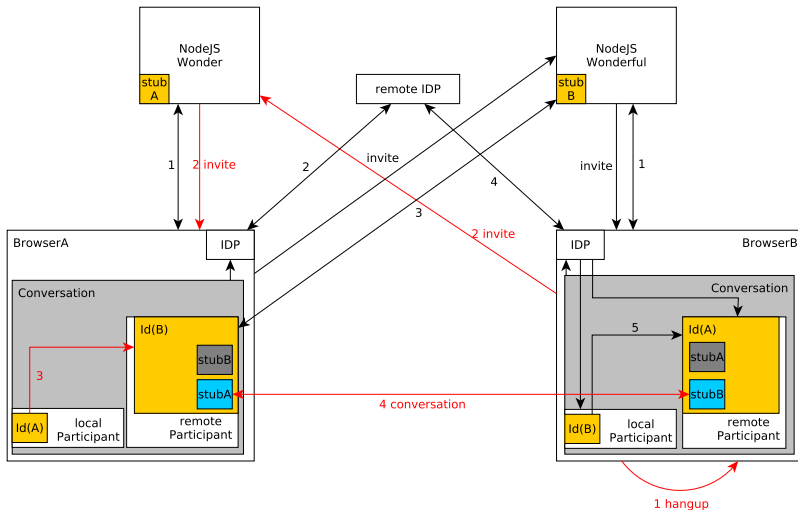
no-messagingStub Beispiel

```
▼ messagingStub: MessagingStub
  ▶ buffer: Array[0]
  impl: null
  ▶ listeners: Array[3]
  message: "No implementation downloaded and assigned to this stub yet!"
  ▶ __proto__: MessagingStub
```



- ▶ Remote identity bekommt korrekten messagingStub
- ▶ Eigener messagingStub wird beim Angerufenen kopiert

Rückruf



Teillösung 1

Conversation.prototype.acceptInvitation Zeile 308

```
1 toIdentity.originalStub = toIdentity.↵  
  messagingStub;  
2 toIdentity.messagingStub = that.myParticipant.↵  
  identity.messagingStub;
```

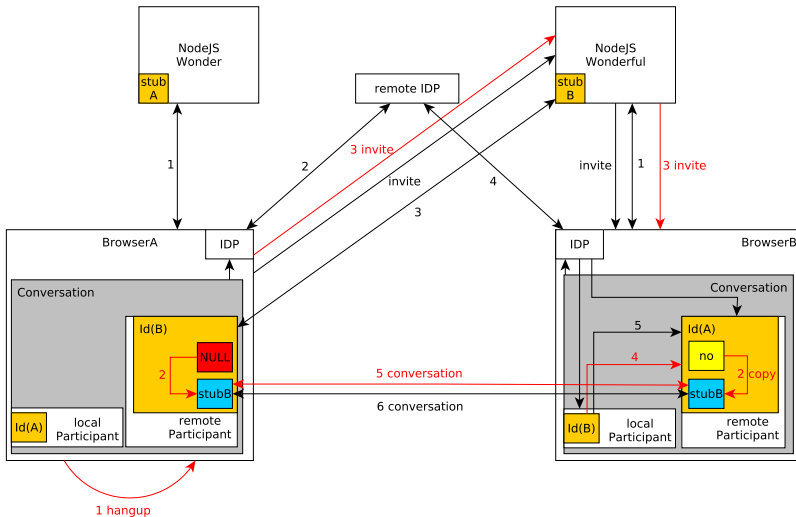
Conversation.prototype.close Zeile 498

```
1 element.identity.messagingStub = element.↵  
  identity.originalStub;
```

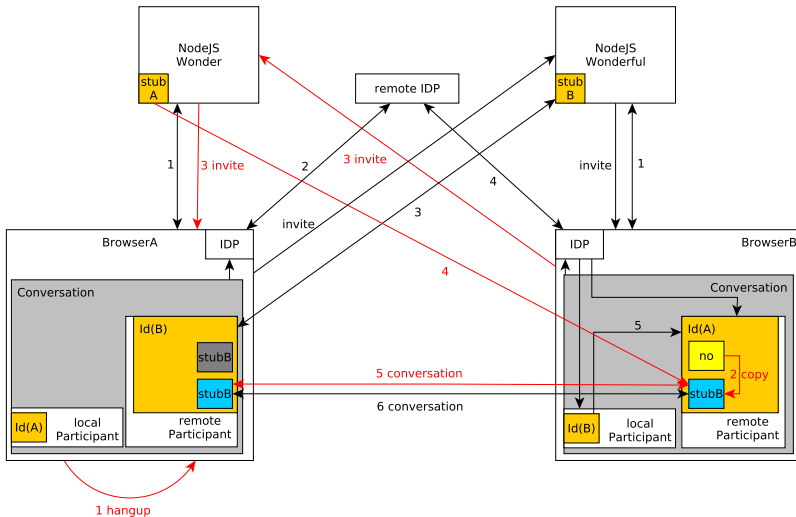
Conversation.prototype.bye Zeile 522

```
1 element.identity.messagingStub = element.↵  
  identity.originalStub;
```

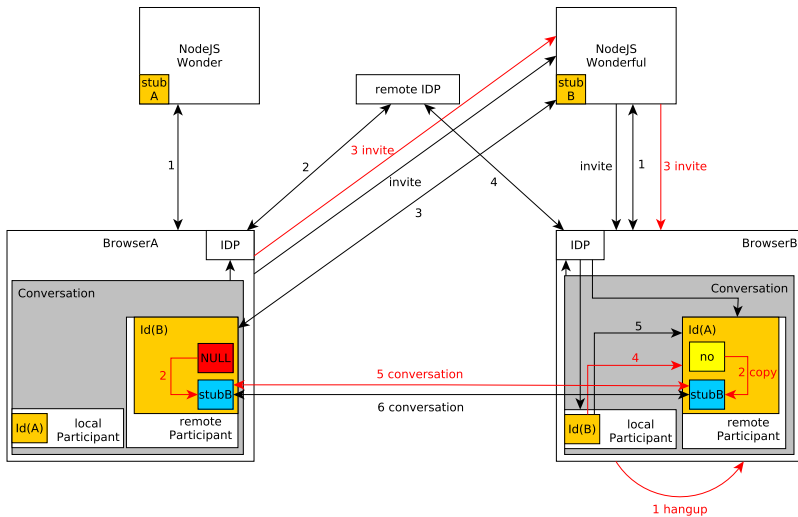
Teillösung 1



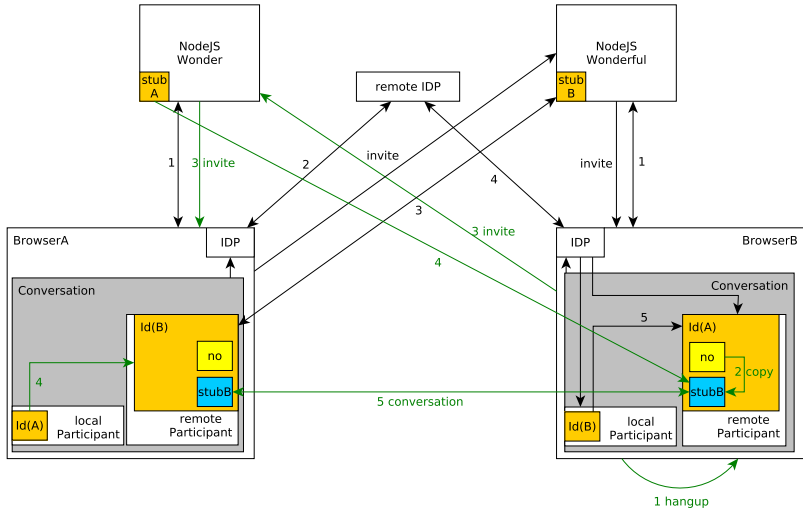
Teillösung 1



Teillösung 1



Teillösung 1



Teillösung 1

- ▶ Rückruf jetzt möglich, jedoch nicht
 - ▶ Wenn der Anrufer aufhängt
 - ▶ Wenn der Angerufene auflegt und der Anrufer erneut anruft
- ▶ Kopieren des `messagingStubs` löst das Problem nur teilweise beim Auflegen des Angerufenen
- ▶ 1/4 Probleme gelöst, jedoch auch 1/4 Probleme erzeugt
- ▶ Nicht funktionstüchtig
 - ▶ A hangup & A call B
 - ▶ A hangup & B call A
 - ▶ B hangup & A call B
- ▶ Funktionstüchtig
 - ▶ B hangup & B call A

Nullstub umgehen

```
1 Identity.prototype.resolve = ↵  
    function( callback ) {  
2      var that = this;  
3      console.log( "resolving ↵  
        identity:↵"  
        + this.↵  
        rtcIdentity);  
4      // if ( ! this.messagingStub.↵  
        impl ) {  
5      if ( ! this.messagingStub || ! ↵  
        this.messagingStub.impl ) {  
6        var knownStub = Idp.↵  
            getInstance().↵  
            getResolvedStub(this.↵  
            messagingStubLibUrl);
```

Teillösung 2

- ▶ Im angerufenen remote participant wird nur noch bei `Conversation.prototype.bye` der Stub verändert

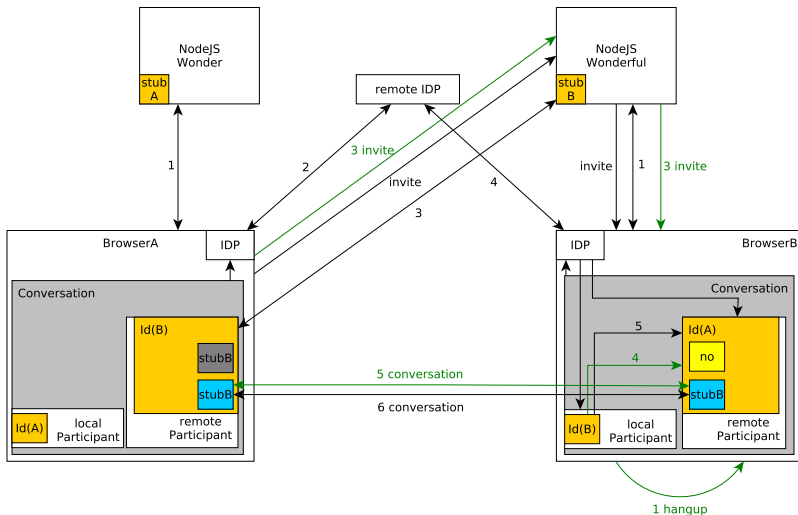
Conversation.prototype.acceptInvitation

```
1 toIdentity.originalStub = toIdentity.↵  
  messagingStub;  
2 toIdentity.messagingStub = that.myParticipant.↵  
  identity.messagingStub;
```

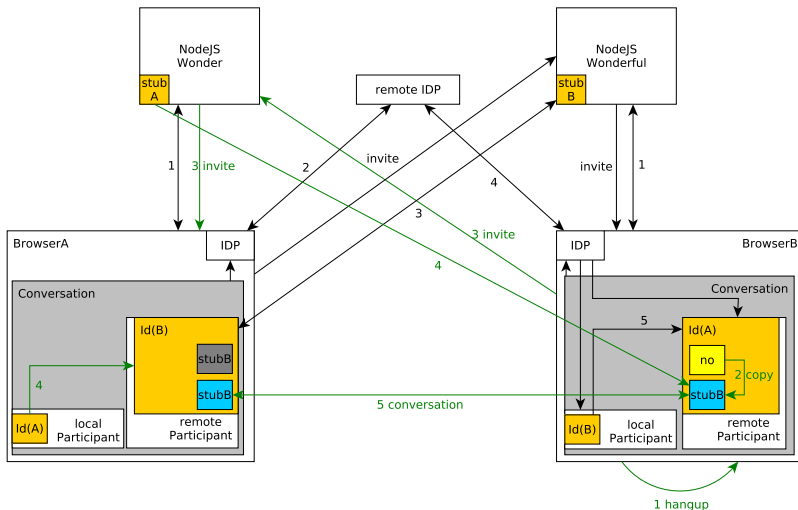
Conversation.prototype.bye

```
1 element.identity.messagingStub = element.↵  
  identity.originalStub;
```

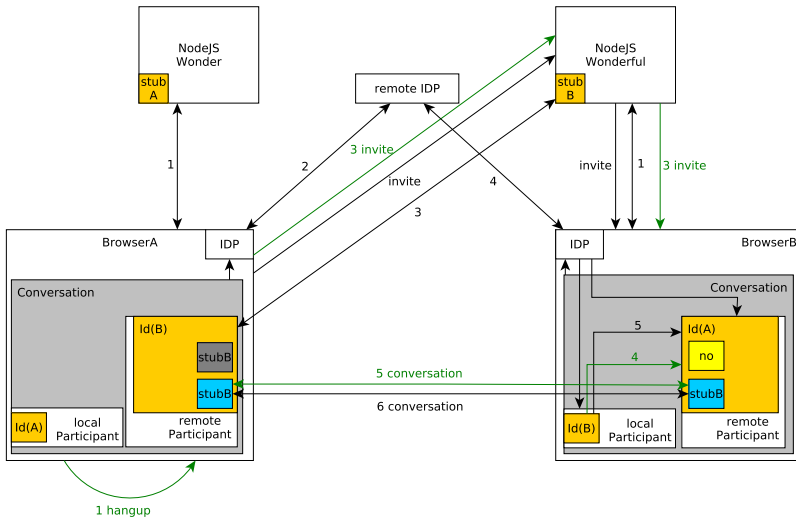
Teillösung 2



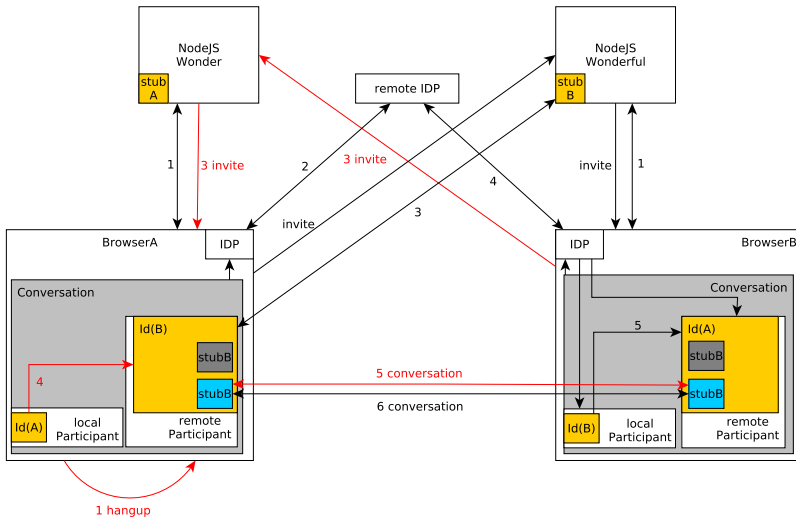
Teillösung 2



Teillösung 2



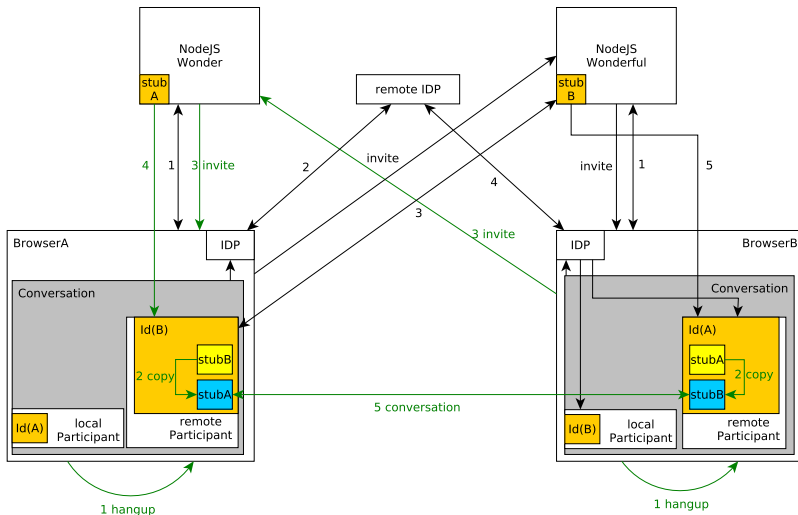
Teillösung 2



Teillösung 2

- ▶ Rückruf jetzt möglich, jedoch nicht
 - ▶ Wenn der Anrufer aufhängt
- ▶ 3/4 Problemen behoben

Lösung



Lösung

- ▶ *.bye und *.close nicht geeignet
- ▶ Beide nutzen jedoch die leave-Methode

Participant.prototype.leave

```
1 this.identity.messagingStub=this.identity.↵  
   originalStub;
```

- ▶ Recall jetzt in jeder Kombination möglich
- ▶ Da originaler messagingStub gespeichert wird, muss kein erneuter Download erfolgen

Zusatzaufgabe

Zusammenfassung

Referenzen



RFC 3828 (Juli 2004),

The Lightweight User Datagram Protocol (UDP-Lite)

<http://tools.ietf.org/html/rfc3828>

Vielen Dank
für eure Aufmerksamkeit.

Sind Fragen offen?