

EFG Bad Lausick - Dokumentationen

Jitsi-Meet installieren

Bernd Reuther

Bernd Reuther

Inhaltsverzeichnis

1. Jitsi Meet installieren	3
1.1 Ausgangspunkt	3
1.2 Ziel	3
1.3 Installation	3
1.4 Konfiguration anpassen	7
1.5 Zusätzliche Funktionen	17
1.6 Quellen	24
1.7 Kommentare, Fragen, Anregungen	24

1. Jitsi Meet installieren

1.1 Ausgangspunkt

- Ubuntu 18.04 Server
- Apache Webserver
- Domain, unter der Jitsi Meet erreichbar sein soll: `konferenz.domain.tld`
- LetsEncrypt Zertifikat für die verwendete Domain
- Etherpad Installation (optional)

Achtung

Wenn unter `/etc/apache2/sites-available/` schon eine Datei `konferenz.domain.tld.conf` existiert, vorher sichern bzw. umbenennen, sonst scheitert die automatische Konfiguration der Installationsroutine.

1.2 Ziel

- Jitsi Meet Videokonferenzen auf dem eigenen Server
- angepasstes Layout mit eigenem Logo, Begrüßungstext und Hintergrundbild
- Verweise auf Datenschutzerklärung und Impressum
- gemeinsames Arbeiten an Etherpad Dokumenten
- telefonische Einwahl
- datenschutzgerechte, möglichst Ressourcen schonende Installation

Info

Die Änderungen in den verschiedenen Konfigurationsdateien sind in dieser Dokumentation bewusst nicht pro Datei zusammengefasst, sondern nach Sinn und Ziel getrennt. Dadurch soll erreicht werden, dass man deren Bedeutung und Auswirkung besser nachvollziehen kann.

1.3 Installation

Wie auf der Github Seite von Jitsi Meet beschrieben.

```
echo 'deb https://download.jitsi.org stable/' >> /etc/apt/sources.list.d/jitsi-stable.list
```

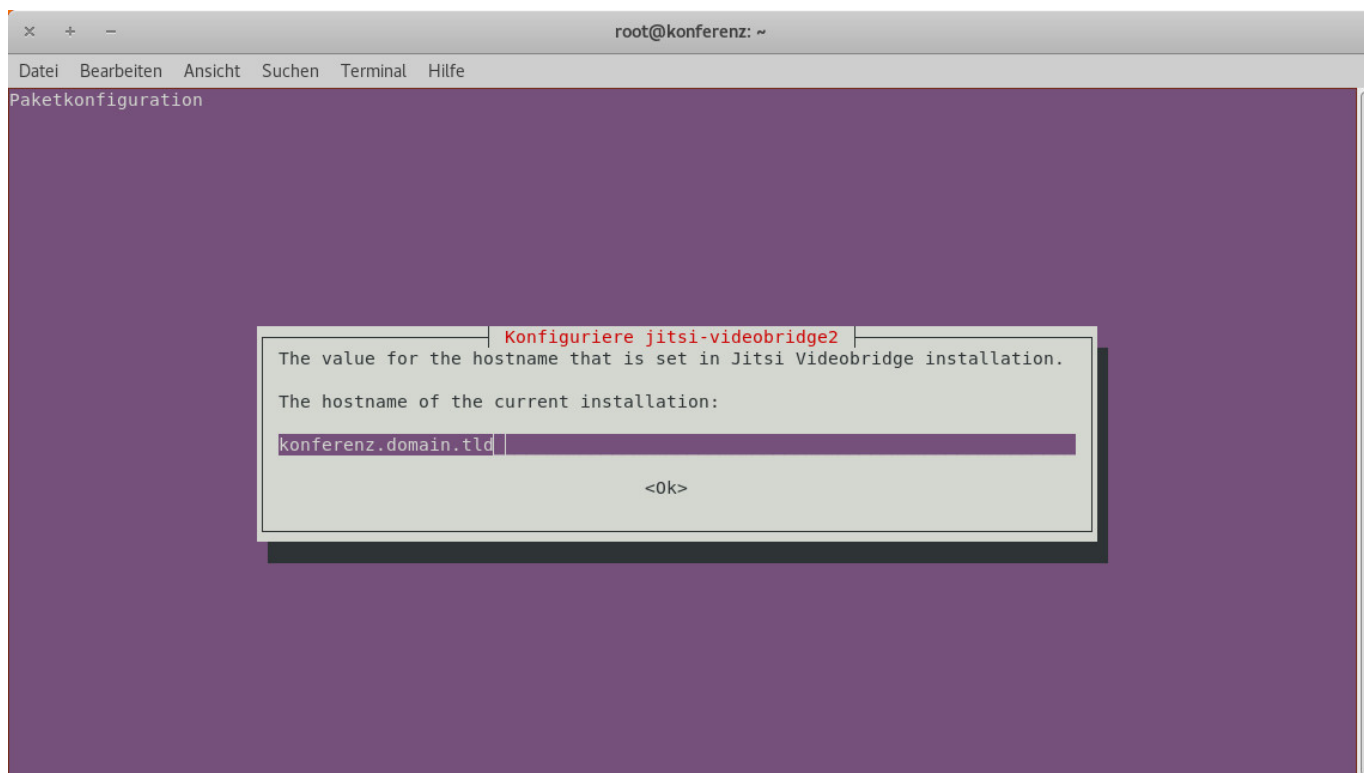
```
wget -q0 - https://download.jitsi.org/jitsi-key.gpg.key | sudo apt-key add -
```

```
apt-get install apt-transport-https
```

```
apt-get update
```

```
apt-get -y install jitsi-meet
```

Während der Installation erscheinen zwei Eingabefenster. Im Ersten wird der Domainname angegeben, unter der Jitsi Meet erreichbar sein soll.



Im zweiten Fenster kann man auswählen, ob während der Installation ein selbst signiertes Zertifikat angelegt werden soll, oder ob man später selbst ein Zertifikat eintragen möchte. Obwohl wir später ein LetsEncrypt Zertifikat verwenden wollen, wird hier die erste Option gewählt. Dadurch kann man vor der weiteren Konfiguration schon einmal testen, ob Jitsi Meet richtig funktioniert.

```
root@konferenz: ~
Datei Bearbeiten Ansicht Suchen Terminal Hilfe
Paketkonfiguration

Konfiguriere jitsi-meet-web-config

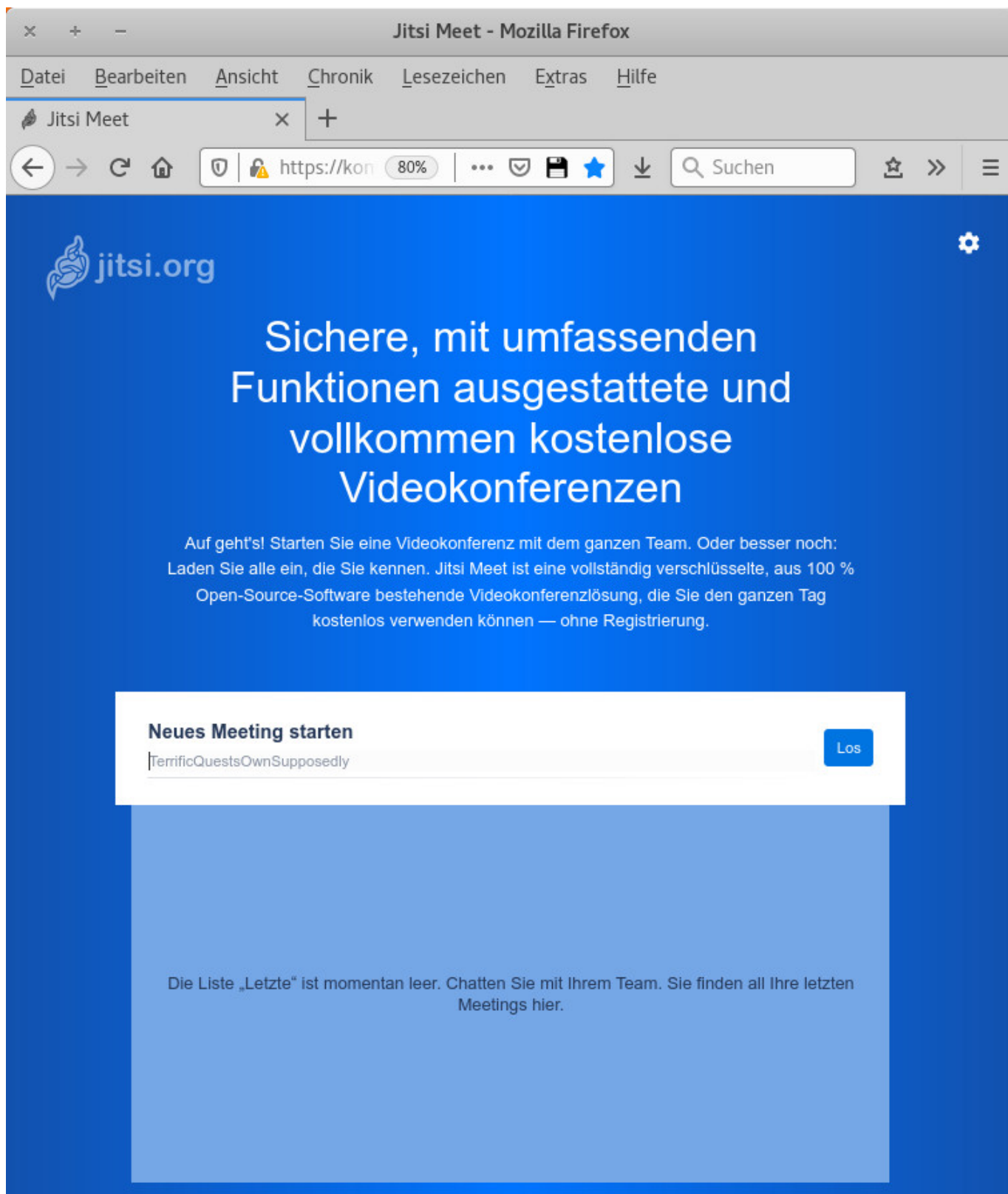
Jitsi Meet is best to be set up with an SSL certificate. Having no certificate, a self-signed one will be generated.
By choosing self-signed you will later have a chance to install Let's Encrypt certificates. Having a certificate
signed by a recognised CA, it can be uploaded on the server and point its location. The default filenames will be
/etc/ssl/--domain.name--.key for the key and /etc/ssl/--domain.name--.crt for the certificate.

SSL certificate for the Jitsi Meet instance

Generate a new self-signed certificate (You will later get a chance to obtain a Let's encrypt certificate)
I want to use my own certificate

<Ok>
```

Wenn die Installationsroutine durchgelaufen ist, sollte es schon möglich sein, Jitsi Meet auf dem eigenen Server unter `https://konferenz.domain.tld` zu testen (natürlich mit der üblichen Sicherheitswarnung bei eigenen Zertifikaten).



The screenshot shows a Mozilla Firefox browser window with the title "Jitsi Meet - Mozilla Firefox". The address bar shows "https://kon" with a loading indicator at 80%. The page content is on a blue background with the "jitsi.org" logo in the top left and a gear icon in the top right. The main heading reads "Sichere, mit umfassenden Funktionen ausgestattete und vollkommen kostenlose Videokonferenzen". Below this, a paragraph states: "Auf geht's! Starten Sie eine Videokonferenz mit dem ganzen Team. Oder besser noch: Laden Sie alle ein, die Sie kennen. Jitsi Meet ist eine vollständig verschlüsselte, aus 100 % Open-Source-Software bestehende Videokonferenzlösung, die Sie den ganzen Tag kostenlos verwenden können — ohne Registrierung." A white box titled "Neues Meeting starten" contains a text input field with the placeholder "TerrificQuestsOwnSupposedly" and a blue "Los" button. At the bottom of the page, a light blue box contains the text: "Die Liste „Letzte“ ist momentan leer. Chatten Sie mit Ihrem Team. Sie finden all Ihre letzten Meetings hier."

1.4 Konfiguration anpassen

1.4.1 LetsEncrypt Zertifikate verwenden

Für eine sichere Verbindung wird ein Zertifikat verwendet, die sich sehr einfach von Lets Encrypt ausstellen lässt. Hier wird davon ausgegangen, dass dieses Zertifikat für die Domain `konferenz.domain.tld` bereits vorhanden ist.

Bei der Installation von Jitsi Meet wurde ein selbst signiertes Zertifikat erstellt und in der Apache Konfigurationsdatei eingetragen. Diesen Eintrag ersetzen wir jetzt und verweisen auf das Zertifikat von Lets Encrypt.

Dazu wird die Datei `/etc/apache2/sites-available/konferenz.domain.tld.conf` geändert:

```
SSLCertificateFile /etc/jitsi/meet/konferenz.domain.tld.crt
SSLCertificateKeyFile /etc/jitsi/meet/konferenz.domain.tld.key
```

ersetzen durch:

```
SSLCertificateFile /etc/letsencrypt/live/konferenz.domain.tld/fullchain.pem
SSLCertificateKeyFile /etc/letsencrypt/live/konferenz.domain.tld/privkey.pem
```

Konfiguration neu laden

```
service apache2 reload
```

1.4.2 Jitsi Meet vor unbefugter Benutzung schützen

Nach der Standard-Installation von Jitsi Meet kann jeder, der die Web-Adresse kennt, ein Meeting starten und Gäste einladen. Das bringt den eigenen Server (und die erlaubte Bandbreite) möglicherweise schnell an die Grenze und ist daher nicht ratsam. Eine sinnvolle Einstellung ist, das Erstellen von Meetings nur für registrierte Benutzer zu erlauben, während Gäste ohne Anmeldung zu den Meetings eingeladen werden können.

In der Datei `/etc/prosody/conf.avail/konferenz.domain.tld.cfg.lua` folgende Änderungen vornehmen:

```
VirtualHost "konferenz.domain.tld"
  -- enabled = false -- Remove this line to enable this host
  authentication = "anonymous"
```

ändern in:

```
VirtualHost "konferenz.domain.tld"
  -- enabled = false -- Remove this line to enable this host
  authentication = "internal_plain"
```

Damit können schon einmal keine neuen Konferenzen von nicht registrierten Benutzern angelegt werden. Allerdings müssten sich nun auch Gäste anmelden, um an einer Konferenz teilzunehmen. Daher wird am Ende der Datei noch ein `VirtualHost "guest..."` ohne Authentication eingefügt:

```
VirtualHost "guest.konferenz.domain.tld"
  authentication = "anonymous"
  c2s_require_encryption = false
  modules_enabled = {
    "ping"; -- Enable mod_ping
    "speakerstats";
    "turncredentials";
    "conference_duration";
  }
}
```

Diese neue interne "Gast"-Domain muss jetzt noch in die Datei `/etc/jitsi/meet/konferenz.domain.tld-config.js` eingetragen, bzw. auskommentiert und angepasst werden:

```
var config = {
  // Connection
  //

  hosts: {
    // XMPP domain.
    domain: 'konferenz.domain.tld',

    // When using authentication, domain for guest users.
    // anonymousdomain: 'guest.exemple.com',
  }
}
```

ändern in:

```
var config = {
  // Connection
  //

  hosts: {
    // XMPP domain.
    domain: 'konferenz.domain.tld',

    // When using authentication, domain for guest users.
    anonymousdomain: 'guest.konferenz.domain.tld',
  }
}
```

In der Datei `/etc/jitsi/jicofo/sip-communicator.properties` wird folgender Eintrag eingefügt:

```
org.jitsi.jicofo.auth.URL=XMPP:konferenz.domain.tld
```

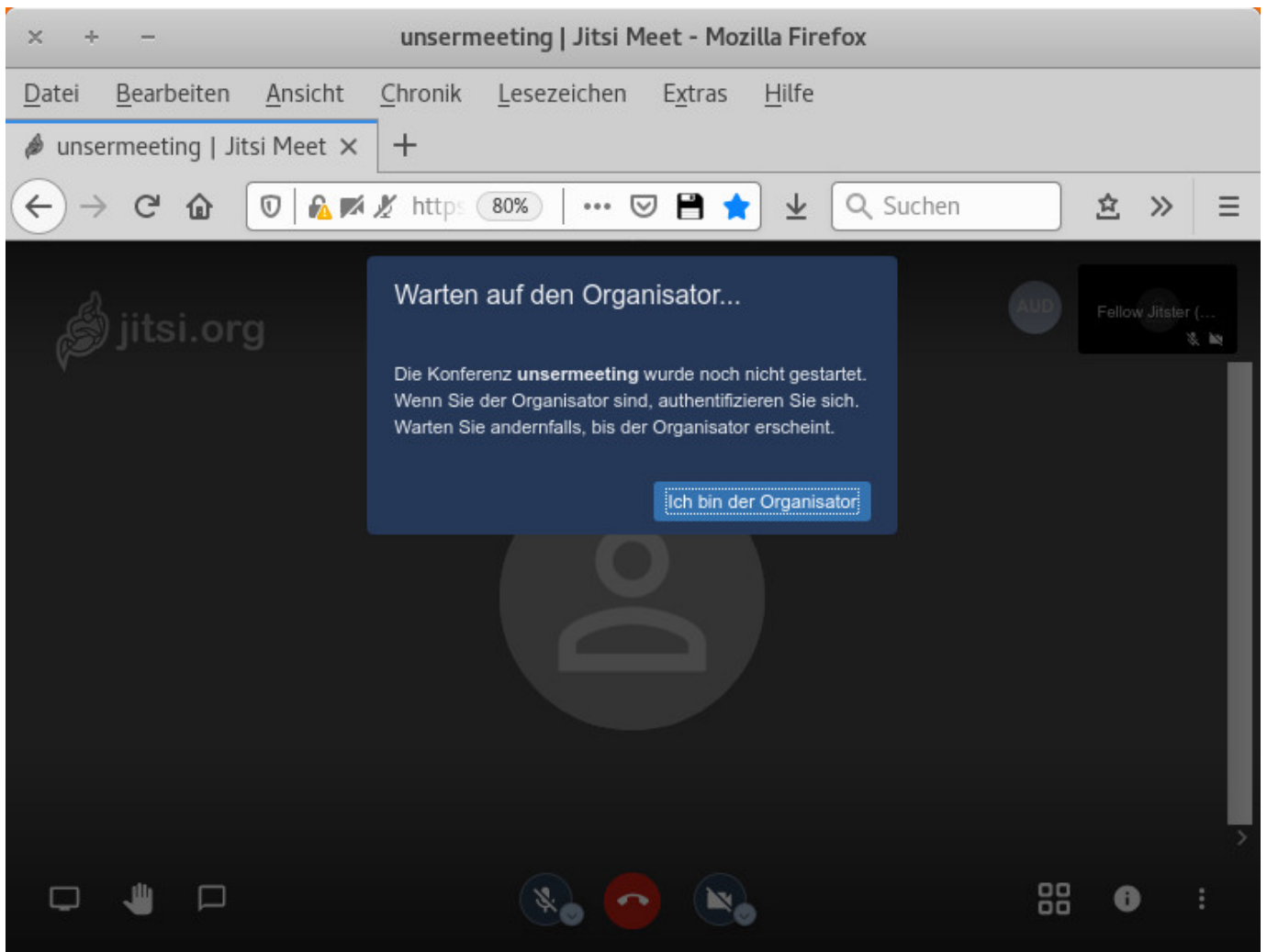
Jetzt wird ein Benutzer mit dem folgenden Befehl angelegt:

```
prosodyctl register <benutzername> konferenz.domain.tld <passwort>
```

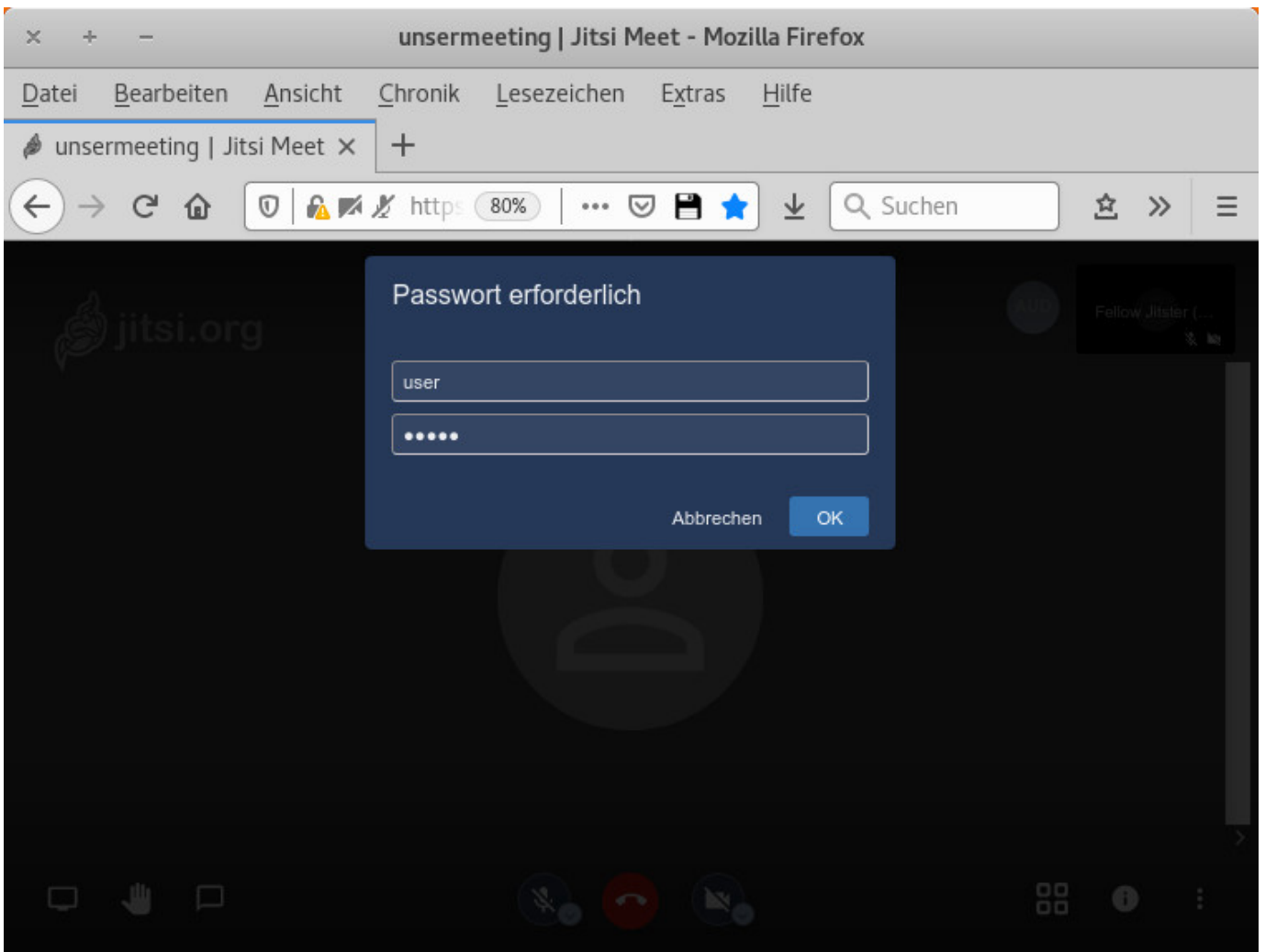
Und noch die entsprechenden Dienste neu starten:

```
systemctl restart jitsi-videobridge2 jicofo prosody
```

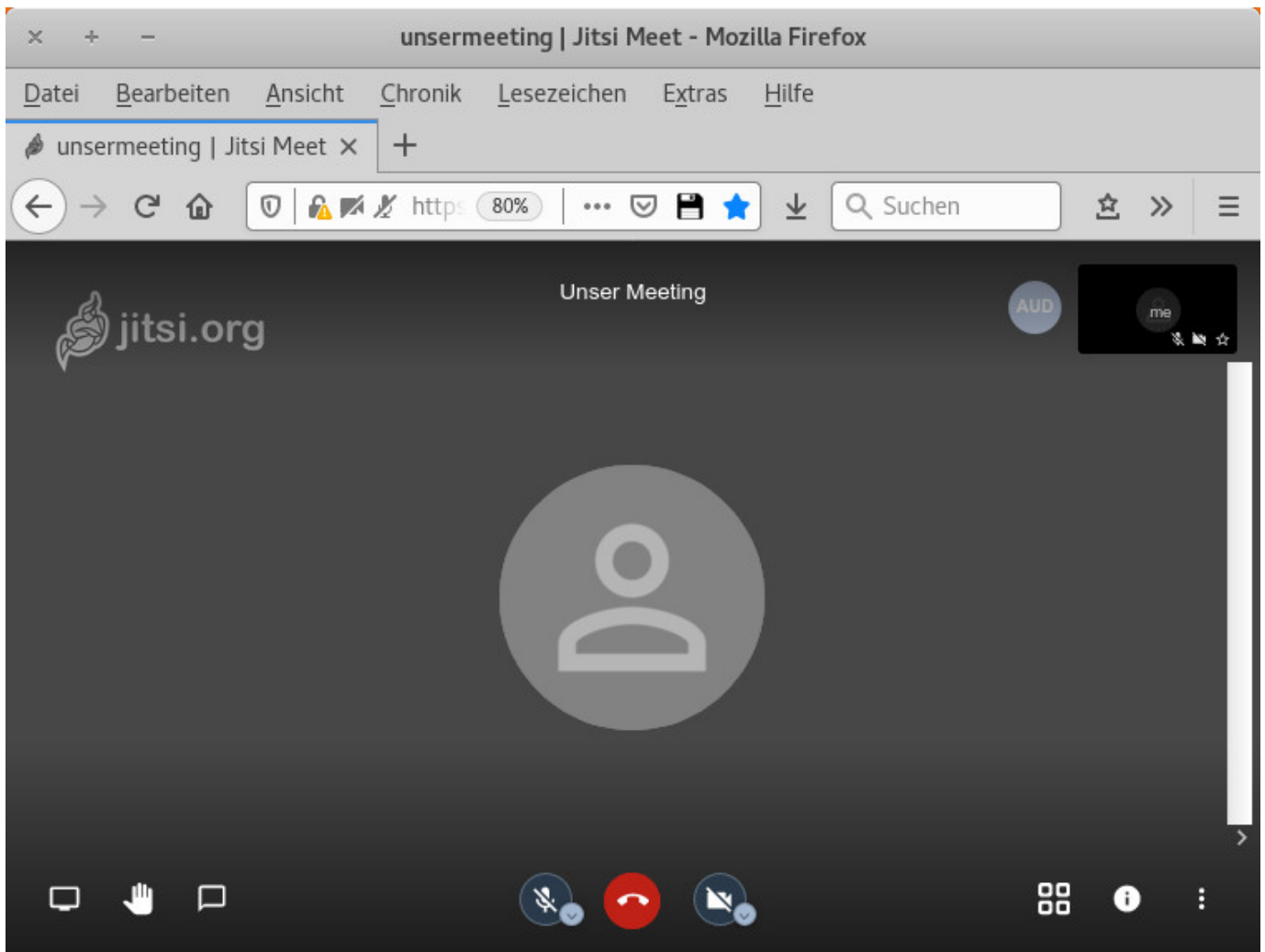
Wenn man jetzt ein neues Meeting startet, kommt ein Hinweis, dass man sich als "Organisator" anmelden muss:



Mit den im vorherigen Schritt erstellten Benutzerdaten kann man sich jetzt anmelden:



...und das Meeting starten:



Solange das Meeting aktiv ist, können Gäste, die den Link kennen, ohne Anmeldung teilnehmen.

1.4.3 Design anpassen

Wir möchten jetzt noch das Design des selbst gehosteten Jitsi Meet Servers anpassen, das eigene Logo einfügen und Impressum sowie Datenschutzerklärung in der Fußzeile anzeigen.

Damit die dafür zu ändernden Dateien nicht mit dem nächsten Update überschrieben werden, kopieren wir sie zunächst in ein anderes Verzeichnis. Als Beispiel verwenden wir hier ein Unterverzeichnis von `/var/www/html`, wo üblicherweise die Daten für den Webserver liegen.

```
mkdir /var/www/html/jitsi-custom
```

```
cp /usr/share/jitsi-meet/interface_config.js /var/www/html/jitsi-custom/interface_config.js
```

```
cp /usr/share/jitsi-meet/plugin.head.html /var/www/html/jitsi-custom/plugin.head.html
```

```
cp /usr/share/jitsi-meet/static/welcomePageAdditionalContent.html /var/www/html/jitsi-custom/welcomePageAdditionalContent.html
```

```
cp /usr/share/jitsi-meet/css/all.css /var/www/html/jitsi-custom/all.css
```

```
cp /usr/share/jitsi-meet/lang/main-de.json /var/www/html/jitsi-custom/main-de.json
```

Dort platzieren wir auch noch die Dateien für Impressum, Datenschutzerklärung, Logo und Hintergrundbild

```
/var/www/html/impressum.html
/var/www/html/datenschutz.html
/var/www/html/logo.jpg
/var/www/html/hg.jpg
```

In der Datei `interface_config.js` ändern wir den Seitentitel und den Link für das Logo:

```
var interfaceConfig = {
  ...
  JITSI_WATERMARK_LINK: 'http://eigene.webseite',
  ...
  APP_NAME: 'Unser Meeting',
  ...
}
```

Bei Bedarf können in dieser Datei auch noch andere Änderungen vorgenommen werden. Mike Kuketz beschreibt in seinem Blog zum Beispiel, wie man den Link zum Download der Android-App vom Google PlayStore zum F-Droid-Store umleitet.

Die Datei `plugin.head.html` ist standardmäßig leer und bekommt folgenden Inhalt:

```
<style>
  .welcome-page-content {
    display: flex;
    flex-direction: column;
    flex-grow: 1;
    justify-content: space-between;
    position: relative;
    z-index: 1;
    margin-top: 35px;
    width: 100%
  }
  .welcome-page-content .welcome-footer {
    color: #FFF;
    display: flex;
    padding-bottom: 20px;
    padding-top: 20px;
    width: 100%;
    z-index: 1
  }
  .welcome-page-content .welcome-footer-content a {
    color: #fff !important;
    text-decoration: underline
  }
  .welcome-page-content .welcome-footer-content {
    display: flex;
    justify-content: center;
    width: 100%;
    z-index: 2
  }
  .welcome-page-content .welcome-footer-about {
    display: flex;
    flex-direction: column;
    flex: 1;
    font-size: 14px;
    line-height: 20px;
    text-align: center;
    justify-content: center
```

```

}

.welcome-page-content .welcome-footer-about:last-child {
  margin-left: 4px
}
</style>

```

Der Inhalt der Datei `welcomePageAdditionalContent.html` wird mit den folgenden Zeilen ersetzt:

```

<template id = "welcome-page-additional-content-template">
  <div class="welcome-page-content">
    <div class="welcome-footer">
      <div class="welcome-footer-content">
        <div class="welcome-footer-about">
          <div>
            <a href="https://jitsi.org/jitsi-meet/" rel="noopener" target="_blank">Powered by Jitsi Meet</a> | <a href="/
            impressum.html" rel="noopener" target="_blank">Impressum</a> | <a href="/datenschutz.html" rel="noopener"
            target="_blank">Datenschutzerklärung</a>
          </div>
        </div>
      </div>
    </div>
  </div>
</template>

```

Damit ein eigenes Hintergrundbild angezeigt wird, muss die Datei `all.css` geändert werden. Leider liegt diese Datei nur "minifiziert" vor und ist damit schwer lesbar. Für geringfügige Änderungen reicht es aus, wenn man im Texteditor nach der entsprechenden Stelle sucht. Es gibt aber auch Tools und Webdienste, die minifizierte css-Dateien wieder in ein lesbares Format umwandeln.

Der zu suchende Eintrag lautet:

```
.welcome{background-image:linear-gradient(-90deg,#1251AE 0,#0074FF 50%,#1251AE 100%);
```

und wird ersetzt durch:

```
.welcome{background-image:url(../hg.jpg);background-size:cover;
```

Für einen eigenen Text auf der Begrüßungsseite, muss die Datei `main-de.json` angepasst werden. Dort ändert man im Schlüssel "welcomepage" die Einträge "appDescription" und "title".

```

"welcomepage": {
  ...,
  "appDescription": "Hier kann auch ein eigener Text stehen. Die benutzerdefinierten Texte aus der geänderten Datei 'main-
  de.json' werden natürlich nur angezeigt, wenn die Sprache des Browsers auf deutsch eingestellt ist. Für andere Sprachen müssten die
  dafür vorgesehenen Dateien geändert werden.",
  ...,
  "title": "Unser eigener Jitsi Meet Server"
},

```

Der Apache Webserver muss jetzt so konfiguriert werden, dass er diese Dateien aus einem anderen Verzeichnis ausliefert.

Dazu wird wieder die Datei `/etc/apache2/sites-available/konferenz.domain.tld.conf` geändert bzw. um folgende Einträge ergänzt:

```

Alias "/interface_config.js" "/var/www/html/jitsi-custom/interface_config.js"
<Location /interface_config.js>
  Require all granted
</Location>

Alias "/plugin.head.html" "/var/www/html/jitsi-custom/plugin.head.html"
<Location /plugin.head.html>
  Require all granted
</Location>

Alias "/static/welcomePageAdditionalContent.html" "/var/www/html/jitsi-custom/welcomePageAdditionalContent.html"
<Location /static/welcomePageAdditionalContent.html>
  Require all granted
</Location>

Alias "/lang/main-de.json" "/var/www/html/jitsi-custom/main-de.json"
<Location /lang/main-de.json>
  Require all granted
</Location>

Alias "/css/all.css" "/var/www/html/jitsi-custom/all.css"
<Location /css/all.css>
  Require all granted
</Location>

Alias "/images/watermark.png" "/var/www/html/jitsi-custom/logo.png"
<Location /images/watermark.png>
  Require all granted
</Location>

Alias "/hg.jpg" "/var/www/html/jitsi-custom/hg.jpg"
<Location /hg.jpg>
  Require all granted
</Location>

Alias "/impressum.html" "/var/www/html/jitsi-custom/impressum.html"
<Location /impressum.html>
  Require all granted
</Location>

Alias "/datenschutz.html" "/var/www/html/jitsi-custom/datenschutz.html"
<Location /datenschutz.html>
  Require all granted
</Location>

```

Nachdem die Konfiguration des Webservers neu geladen wurde:

```
service apache2 reload
```



wird Jitsi Meet mit dem geänderten Layout angezeigt:

Unser Meeting - Mozilla Firefox

Datei Bearbeiten Ansicht Chronik Lesezeichen Extras Hilfe

Unser Meeting

https://kon 80% Suchen

Unser eigener Jitsi Meet Server

Hier kann auch ein eigener Text stehen. Die benutzerdefinierten Texte aus der geänderten Datei 'main-de.json' werden natürlich nur angezeigt, wenn die Sprache des Browsers auf deutsch eingestellt ist. Für andere Sprachen müssten die dafür vorgesehenen Dateien geändert werden.

Neues Meeting starten Los

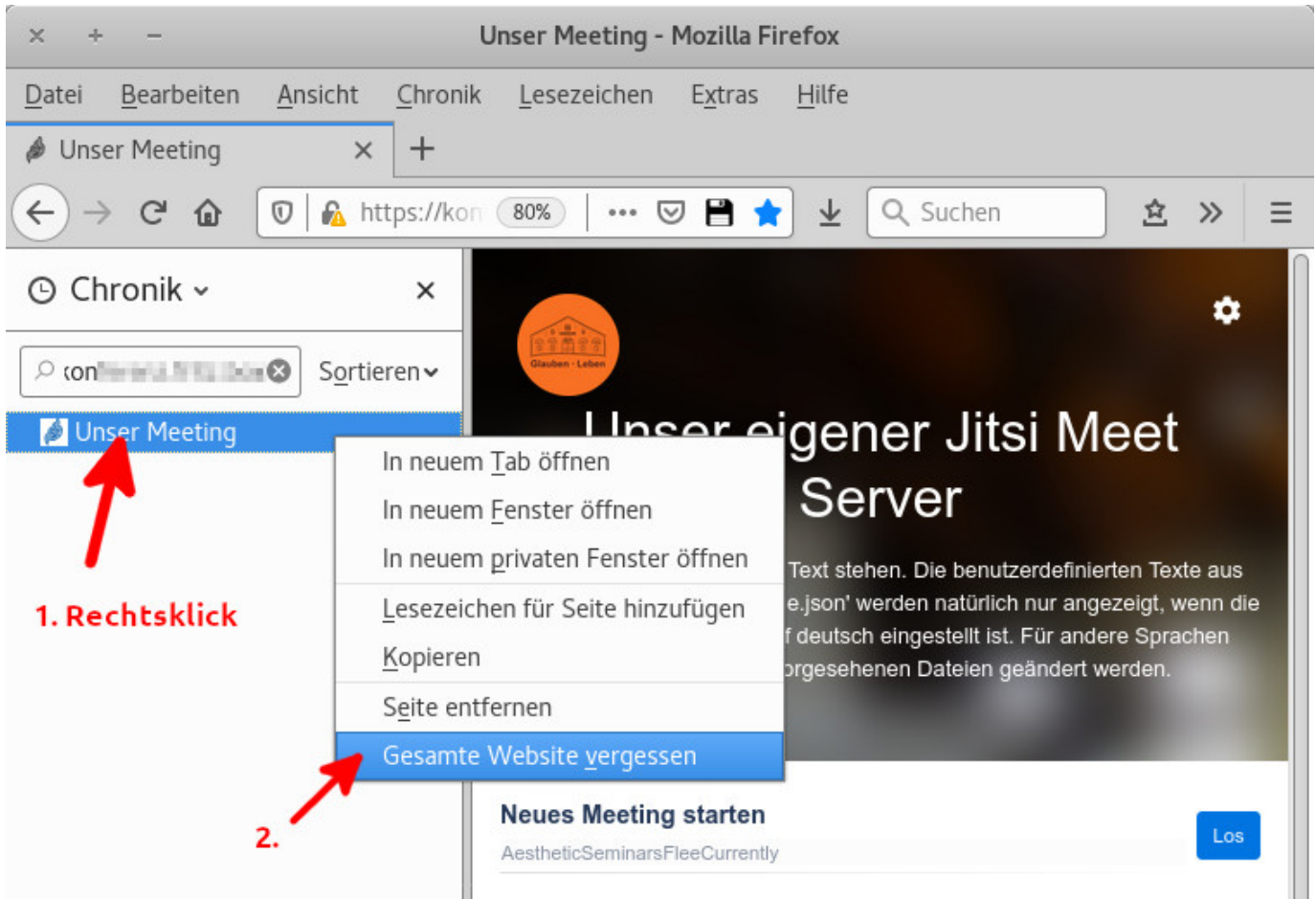
BalancedCollectionsBlendRoughly

Die Liste „Letzte“ ist momentan leer. Chatten Sie mit Ihrem Team. Sie finden all Ihre letzten Meetings hier.

Powered by Jitsi Meet | [Impressum](#) | [Datenschutzerklärung](#)

Hinweis

Wenn die Seite schon vor den Änderungen im Browser geöffnet wurde, kann es sein, dass nicht alle Änderungen angezeigt werden, da sich die Inhalte noch im Browser-Cache befinden. Selbst das Drücken von "F5" funktioniert dann nicht. Erst wenn der Cache für diese Seite gelöscht wird, sieht man alle Änderungen.



1.4.4 Sonstige sinnvolle Einstellungen

In der Datei `/etc/jitsi/meet/konferenz.domain.tld-config.js` werden noch folgende Änderungen vorgenommen, indem die entsprechenden Einträge auskommentiert bzw. geändert werden:

```
// Sprache einstellen
defaultLanguage: 'de',

// verringert die CPU-Last auf den Client-Geräten
disableAudioLevels: true,

// nur für die letzten 6 aktiven Sprecher wird das Video angezeigt
channelLastN: 6,

// Verbindung zu anderen Anbietern verhindern (z.B.: Gravatar)
disableThirdPartyRequests: true,

// Performance Verbesserung, wenn Chrome Client(ab V69) verwendet wird
enableLayerSuspension: true,

// erfordert einen Benutzernamen
```



```
requireDisplayName: true,  
  
// zeigt vor dem Betreten der Konferenz eine Seite an, in der ein Name vergeben und Kamera bzw. Mikrofon konfiguriert werden können  
prejoinPageEnabled: true,
```

Dienste neu starten:

```
systemctl restart jitsi-videobridge2 jicofo prosody
```

1.5 Zusätzliche Funktionen

1.5.1 Etherpad einbinden

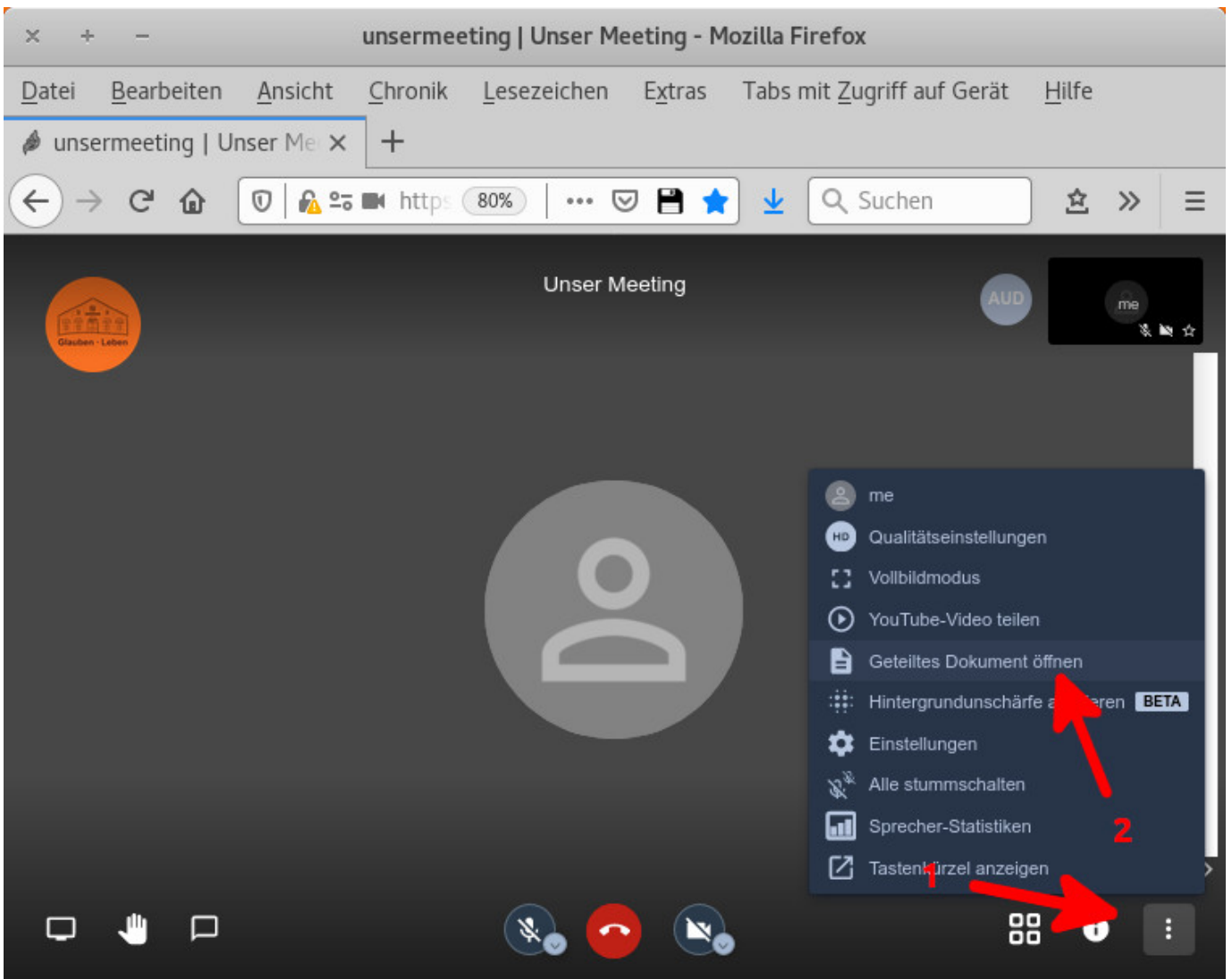
Jitsi Meet bietet die Möglichkeit, während einer Konferenz gemeinsam an einem Textdokument zu arbeiten. Dazu ist eine Etherpad-Installation nötig, die über die Konfigurationsdatei `/etc/jitsi/meet/konferenz.domain.tld-config.js` eingebunden wird:

```
etherpad_base: 'https://pfad.zu.etherpad/p/',
```

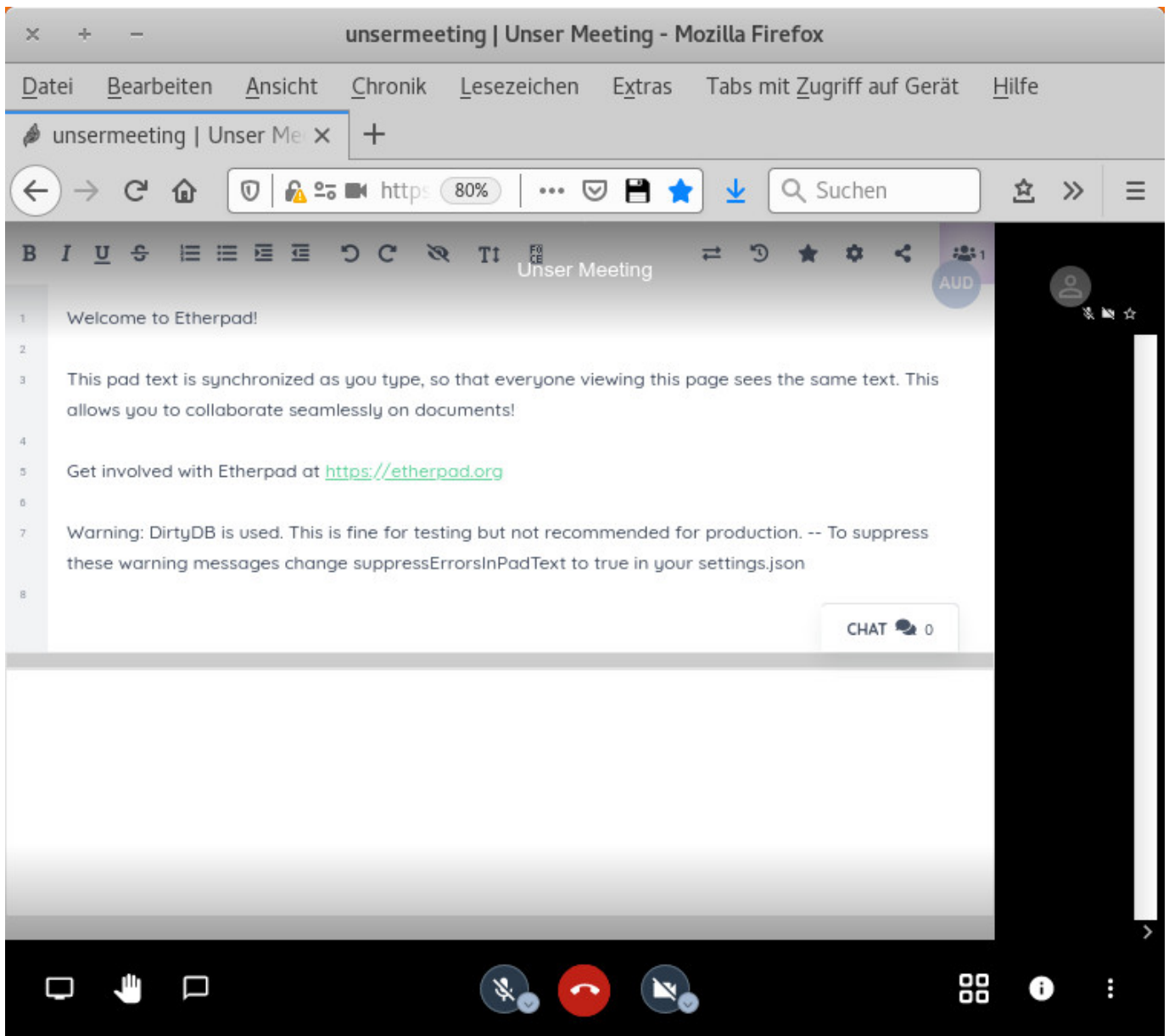
Dienste neu starten:

```
systemctl restart jitsi-videobridge2 jicofo prosody
```

Jetzt sollte es möglich sein, in einer Konferenz ein Etherpad-Dokument zu öffnen:



und gemeinsam zu bearbeiten:



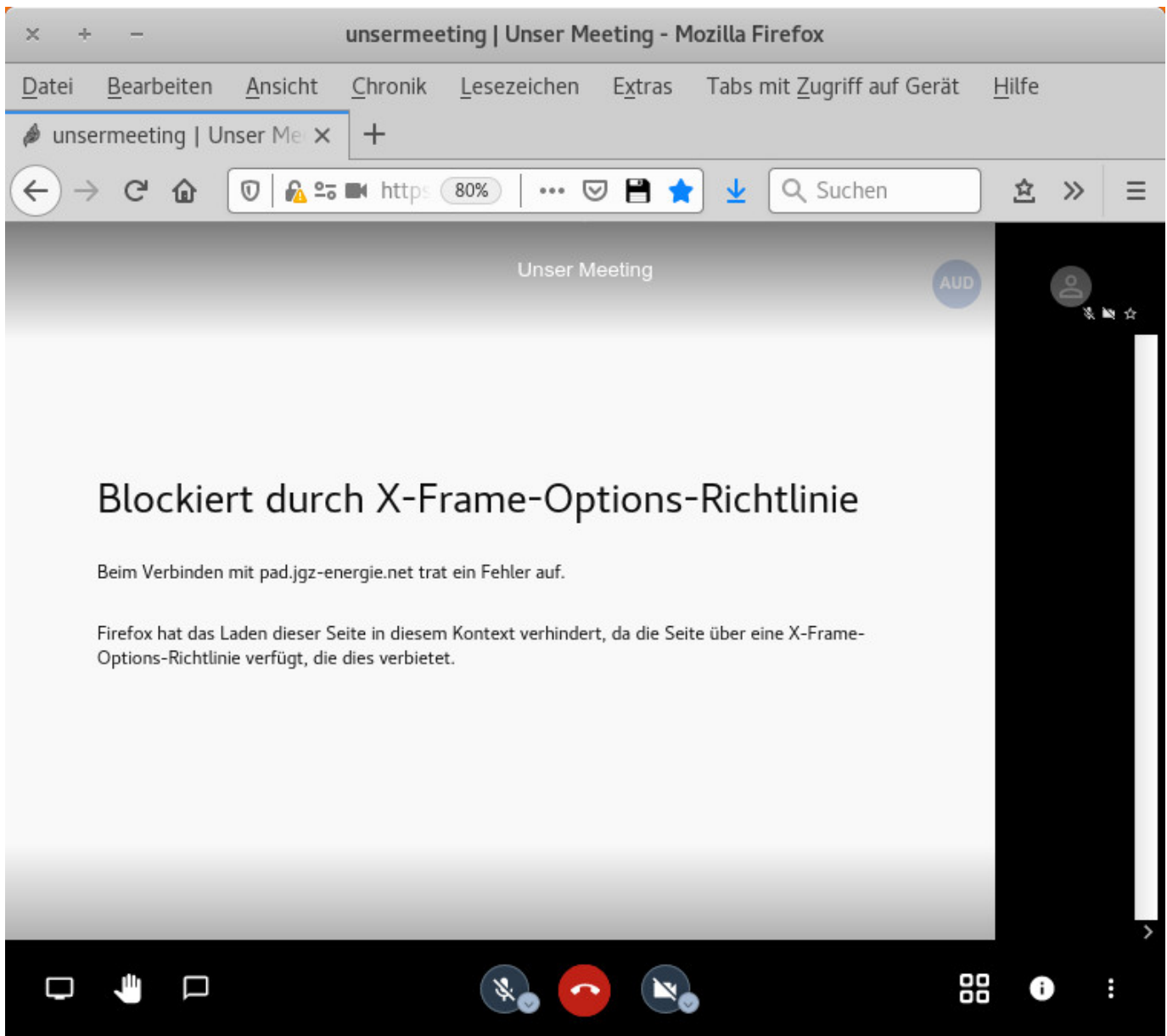
The screenshot shows a Mozilla Firefox browser window with the title "unsermeeting | Unser Meeting - Mozilla Firefox". The address bar shows "https://etherpad.org" with a zoom level of 80%. The page content is an Etherpad document with the following text:

```
1 Welcome to Etherpad!  
2  
3 This pad text is synchronized as you type, so that everyone viewing this page sees the same text. This  
4 allows you to collaborate seamlessly on documents!  
5 Get involved with Etherpad at https://etherpad.org  
6  
7 Warning: DirtyDB is used. This is fine for testing but not recommended for production. -- To suppress  
8 these warning messages change suppressErrorsInPadText to true in your settings.json
```

The interface includes a toolbar with various editing tools (bold, italic, underline, list, link, unlink, undo, redo, eraser, text color, background color) and a chat window at the bottom right with the text "CHAT 0".

⚠ Achtung!

Bei aktuellen LetsEncrypt Installationen wird unter Umständen eine Apache-Konfigurationsdatei erstellt und aktiviert, die den Browser anweist, keine externen Quellen zu laden. Das ist prinzipiell gut und nützlich, verhindert aber in diesem Fall, dass Etherpad in Jitsi Meet geladen wird. Außerdem verhindert diese Einstellung die Benutzung dieses Servers mit der Jitsi Meet Desktop App.



In diesem Fall muss in der Datei `/etc/apache2/conf-enabled/ssl-params.conf` der Eintrag `Header always set X-Frame-Options DENY` auskommentiert werden.

1.5.2 Telefoneinwahl

Mit dem zusätzlichen Software-Paket `jigasi` ist es möglich, per Telefoneinwahl an einer Jitsi Konferenz teilzunehmen. Das kann nützlich sein, wenn gerade kein PC oder Smartphone zur Verfügung steht, oder wenn man an einem PC ohne Mikrofon arbeitet.

Normalerweise kann `jigasi` mit einem beliebigen SIP-Konto verknüpft werden und wäre damit über das Telefonnetz erreichbar. Wir möchten jedoch zusätzlich erreichen, dass

- nach der Telefoneinwahl eine Pin-Nummer abgefragt wird
- mehrere Räume genutzt werden können

Dazu wird unser Jitsi Meet Server mit einem bestehenden Asterisk-Server verbunden, auf dem zuerst ein neues internes SIP-Konto angelegt werden muss.

Anmerkung

Die Beschreibung der kompletten Asterisk-Installation ist ein Thema für sich und würde hier zu weit führen. Ausgangspunkt für die hier beschriebene Anpassung ist ein funktionierender Asterisk-Server, der mit einem Sipgate-Basic Account verbunden und damit über das öffentliche Telefonnetz erreichbar ist.

Vorbereitungen am Asterisk-Server

In der Datei `/etc/asterisk/sip.conf` werden folgende Einträge hinzugefügt:

```
[2000]
type = friend
context = meine-telefone
secret = sip-passwort
host = dynamic
```

Daraus ergeben sich folgende Zugangsdaten, die später bei der Installation von `jigasi` eingetragen werden müssen:

- SIP Benutzername: 2000@sip.server (wobei "sip.server" natürlich auch die IP-Adresse des Asterisk Servers sein kann)
- SIP Passwort: sip-passwort (selbstverständlich hier nur als Beispiel)

Nun müssen noch die Einwahl-Regeln in die Datei `/etc/asterisk/extensions.conf` eingetragen werden:

```
[von-voip-provider]
; xxxxxx = username vom Sipgate Zugang
; yyy1..yyy8 = Beispiele für Asterisk-Konferenzräume
; zzzz = Beispiel für vordefinierte Jitsi Meet Konferenz
exten => xxxxxx,1,Answer()
same => n,Wait(1)
same => n,Playback(conf-getpin)
same => n,Read(NUMMER,,0,3)
same => n,GotoIf($[ ${NUMMER} = yyy1 ]?telko:)
same => n,GotoIf($[ ${NUMMER} = yyy2 ]?telko:)
same => n,GotoIf($[ ${NUMMER} = yyy3 ]?telko:)
same => n,GotoIf($[ ${NUMMER} = yyy4 ]?telko:)
same => n,GotoIf($[ ${NUMMER} = yyy5 ]?telko:)
same => n,GotoIf($[ ${NUMMER} = yyy6 ]?telko:)
same => n,GotoIf($[ ${NUMMER} = yyy7 ]?telko:)
same => n,GotoIf($[ ${NUMMER} = yyy8 ]?telko:)
same => n,GotoIf($[ ${NUMMER} = zzzz ]?videochat1:)
same => n,GotoIf($[ ${NUMMER} > 99999 ]?videochat2:)
same => n,Hangup
same => 20(telko),ConfBridge(${NUMMER},default_bridge,user_ohne_pw,sample_user_menu)
same => 30(videochat1),SIPAddHeader(Jitsi-Conference-Room:UserMeeting)
same => 31,Dial(SIP/2000,20)
same => 40(videochat2),SIPAddHeader(Jitsi-Conference-Room:${NUMMER})
same => 41,Dial(SIP/2000,20)
```

Asterisk Konfiguration neu laden:

```
asterisk -rx 'core reload'
```

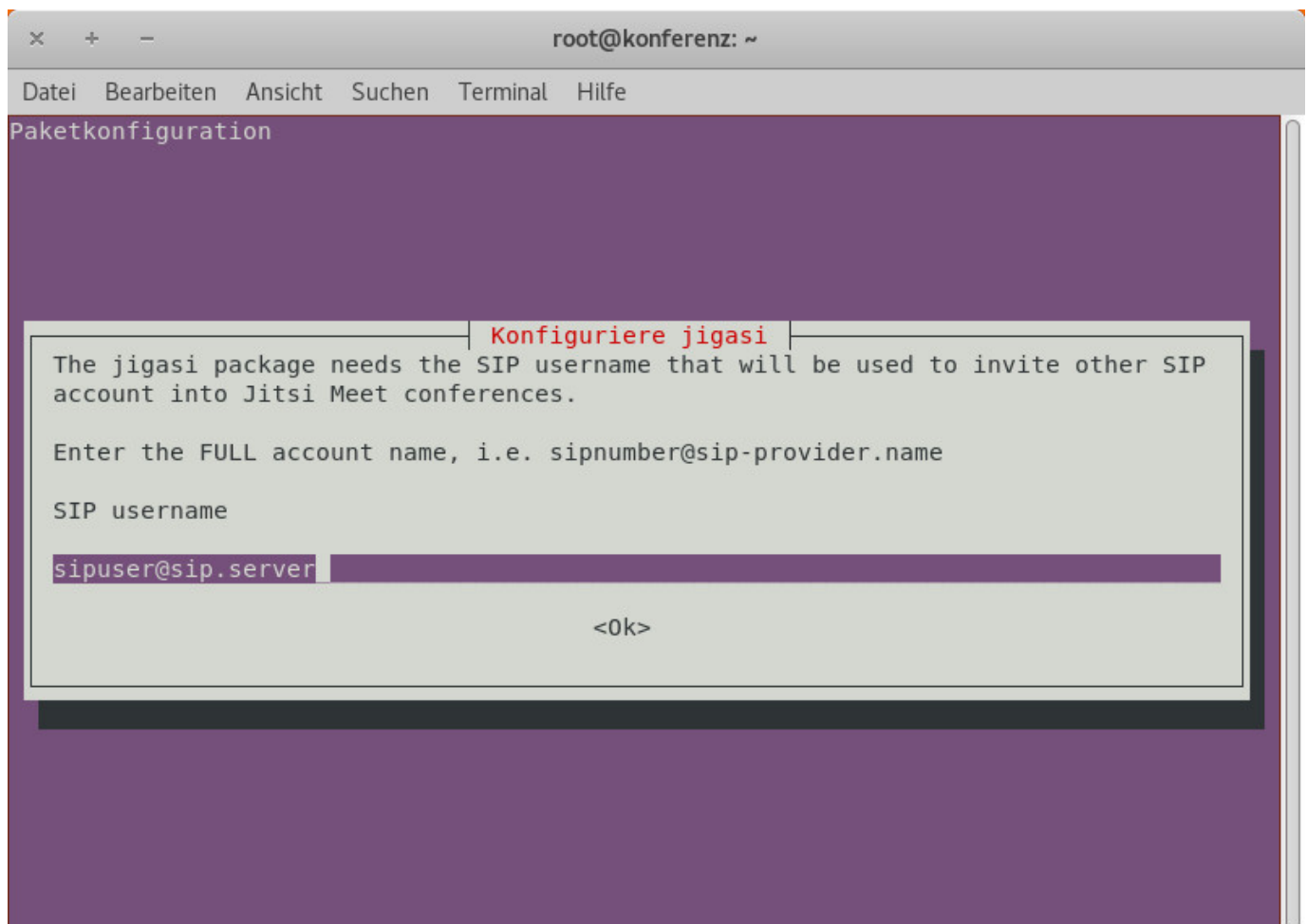
Bei diesem Beispiel wird bei der Telefoneinwahl als erstes eine Konferenz-Pin-Nummer abgefragt. Die ersten 8 Auswertungen dieser Pin prüfen auf voreingestellte, 4-stellige Nummern, die zu "normalen" Telefonkonferenz-Räumen führen. Dann wird als Beispiel eine 4-stellige Nummer ausgewertet, die zu einem voreingestellten Jitsi Meet Konferenzraum führt `UnserMeeting`. Davon können auch noch mehr angelegt werden. Wenn dann das Meeting mit dem entsprechenden Namen läuft, kann man sich mit der Pin dazu einwählen. Die letzte Pin-Auswertung prüft auf eine mindestens 6-stellige Zahl und verwendet diese dann dynamisch als Raumname. Hier muss die Jitsi Meet Konferenz mit einer solchen, gültigen Zahl als Name gestartet werden. Der Einladungs-Link enthält dann automatisch auch die Telefoneinwahl-Pin. Mit dieser Variante stehen sehr viele Konferenzräume mit Telefoneinwahl zur Verfügung, die allerdings keine "sprechenden" Namen haben können.

Jigasi installieren

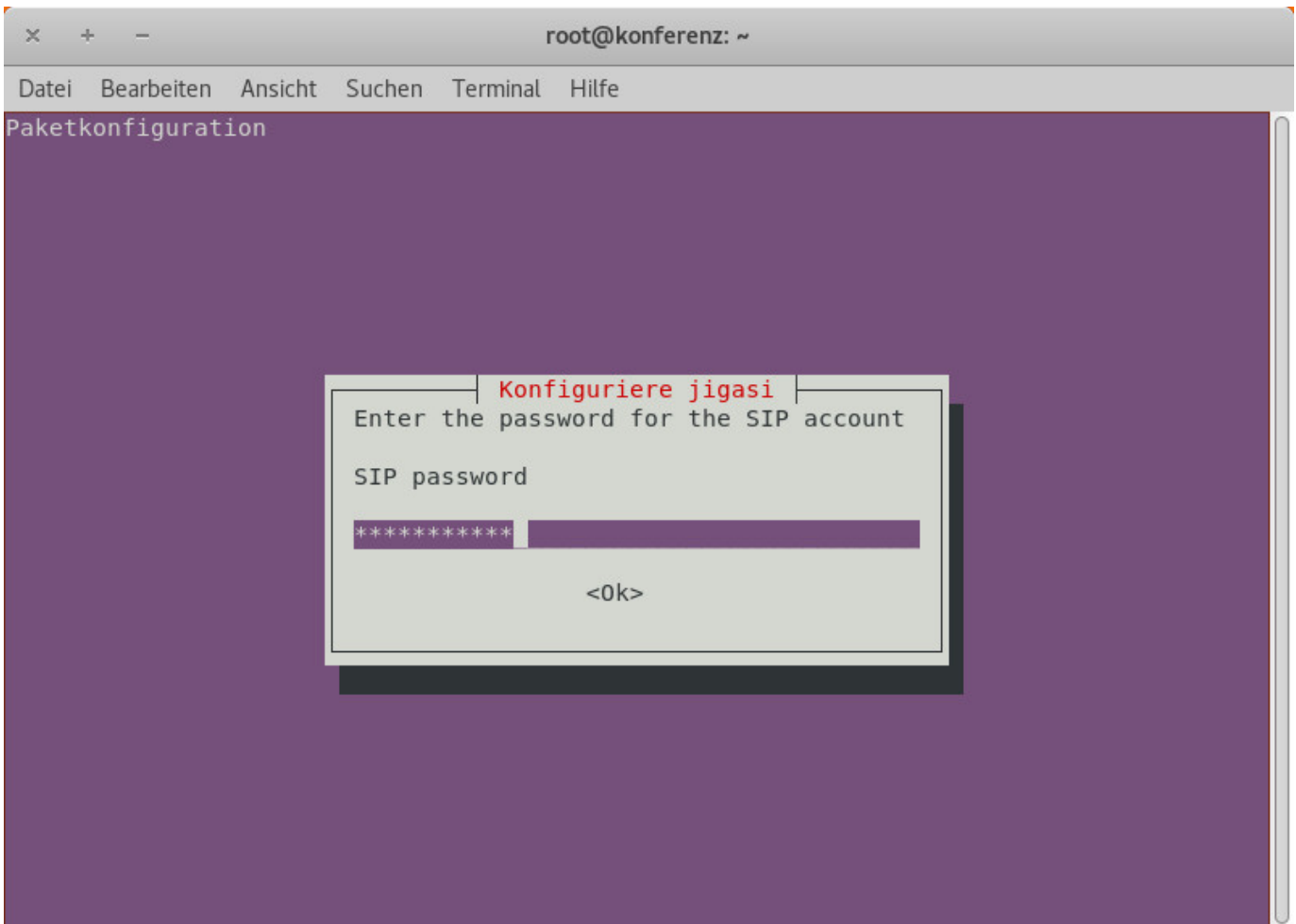
Die Installation von `jigasi` ist mit einem Befehl erledigt:

```
apt-get -y install jigasi
```

Während der Installation wird man aufgefordert, die Zugangsdaten für ein SIP-Konto einzugeben. Dabei werden die im Asterisk-Server eingetragenen Daten verwendet.



und noch das Passwort:



Für Jigasi wird ein neuer Benutzer angelegt (oder es kann auch der bereits vorhandene verwendet werden).

```
prosodyctl register <sipbenutzer> konferenz.domain.tld <passwort>
```

In der Datei `/etc/jitsi/jigasi/sip-communicator.properties` muss noch folgendes eingetragen bzw. auskommentiert und geändert werden:

```
# verhindert Verbindungsabbrüche mit Asterisk
net.java.sip.communicator.impl.protocol.sip.SKIP_REINVITE_ON_FOCUS_CHANGE_PROP=true

# Raumname, wenn ein externes Sip-Komto verwendet werden soll,
# welches keine Sip-Header unterstützt
org.jitsi.jigasi.DEFAULT_JVB_ROOM_NAME=JitsiTelKo

org.jitsi.jigasi.MUC_SERVICE_ADDRESS=conference.konferenz.domain.tld

# wenn Asterisk nicht auf dem Standard Port 5060 hört (was sinnvoll ist)
# Adresse und Port entsprechend anpassen
net.java.sip.communicator.impl.protocol.sip.acc1403273890647.PROXY_ADDRESS=sip.server
net.java.sip.communicator.impl.protocol.sip.acc1403273890647.PROXY_AUTO_CONFIG=false
net.java.sip.communicator.impl.protocol.sip.acc1403273890647.PROXY_PORT=5228
net.java.sip.communicator.impl.protocol.sip.acc1403273890647.PREFERRED_TRANSPORT=UDP

# If you want jigasi to perform authenticated login instead of anonymous login
# to the XMPP server, you can set the following properties.
```

```
org.jitsi.jigasi.xmpp.acc.USER_ID=sipbenutzer@konferenz.domain.tld
org.jitsi.jigasi.xmpp.acc.PASS=passwort
org.jitsi.jigasi.xmpp.acc.ANONYMOUS_AUTH=false

# Activate this property if you are using self-signed certificates or other
# type of non-trusted certificates. In this mode your service trust in the
# remote certificates always.
net.java.sip.communicator.service.gui.ALWAYS_TRUST_MODE_ENABLED=true
```

Den Dienst neu starten

```
systemctl restart jigasi
```

Jetzt sollte es möglich sein, sich über das Telefonnetz in eine laufende Jitsi Meet Konferenz einzuwählen.

i Info

Nach meinem Kenntnisstand, können sich über eine Sippgate Nummer bis zu 15 Personen einwählen. In unseren normalen Telefonkonferenzen hatten wir bis jetzt immer nur bis zu 8 Teilnehmer.

1.6 Quellen

Die Informationen in dieser Dokumentation stammen unter Anderem aus den folgenden Internetseiten:

- <https://github.com/jitsi/jitsi-meet/blob/master/doc/quick-install.md>
- <https://github.com/jitsi/jitsi-videobridge/blob/master/doc/last-n.md>
- <https://www.kuketz-blog.de/jitsi-meet-server-einstellungen-fuer-einen-datenschutzfreundlichen-betrieb/>
- <https://www.kuketz-blog.de/jitsi-meet-optimierung-der-performance/>
- <https://www.digitalocean.com/community/tutorials/how-to-install-jitsi-meet-on-ubuntu-18-04>
- <https://dev.to/noandrea/self-hosted-jitsi-server-with-authentication-ie7>
- <https://community.jitsi.org/t/sip-calls-into-jigasi-from-freeswitch-fail-on-jigasi-1-1-38-succeed-on-jigasi-1-0-244/25734>
- <https://github.com/jitsi/jigasi/issues/132>

Diese Dokumentation darf gemäß der CC BY-NC-SA 3.0 Lizenz verwendet werden.

Erstellt: 04-2020 von Bernd Reuther, Evangelisch-Freikirchliche Gemeinde Bad Lausick.

geändert: 20.Oktober 2020

1.7 Kommentare, Fragen, Anregungen
