Secure Internet Live Conferencing

Christian Horchert <fukami@c3d2.de>

Frank Becker <fb@alien8.de>

Chaostreff Dresden http://www.c3d2.de

2004-03-07, 6. Chemnitzer Linux Tag

Was ist SILC?

Netzwerkprotokoll für
authentifizierte und **verschlüsselte**
Live Kommunikation

Unterstützt Chat- und Instant Messenger Systeme.

/join linuxtag

Agenda

- Geschichte und Zukunft von SILC
- Kommunikationsprotokoll
- SILC-Nutzung
- Software
- Silc Netzwerk

Geschichte/Entwicklung

- entwickelt von Pekka Riikonen, seit '96
- Entwicklung mehrmals unterbrochen
- 1. Veröffentlichung 2000,
- SILC-Client 1.0 Okt. 2003
- z. Zt. “Reifung” des Protokolls
- Protokoll specs bei IETF eingereicht, momentan in “draft phase” (wird bald RFC)
- Version 1 von silcd und toolkit wird bald ...

Das Protokoll

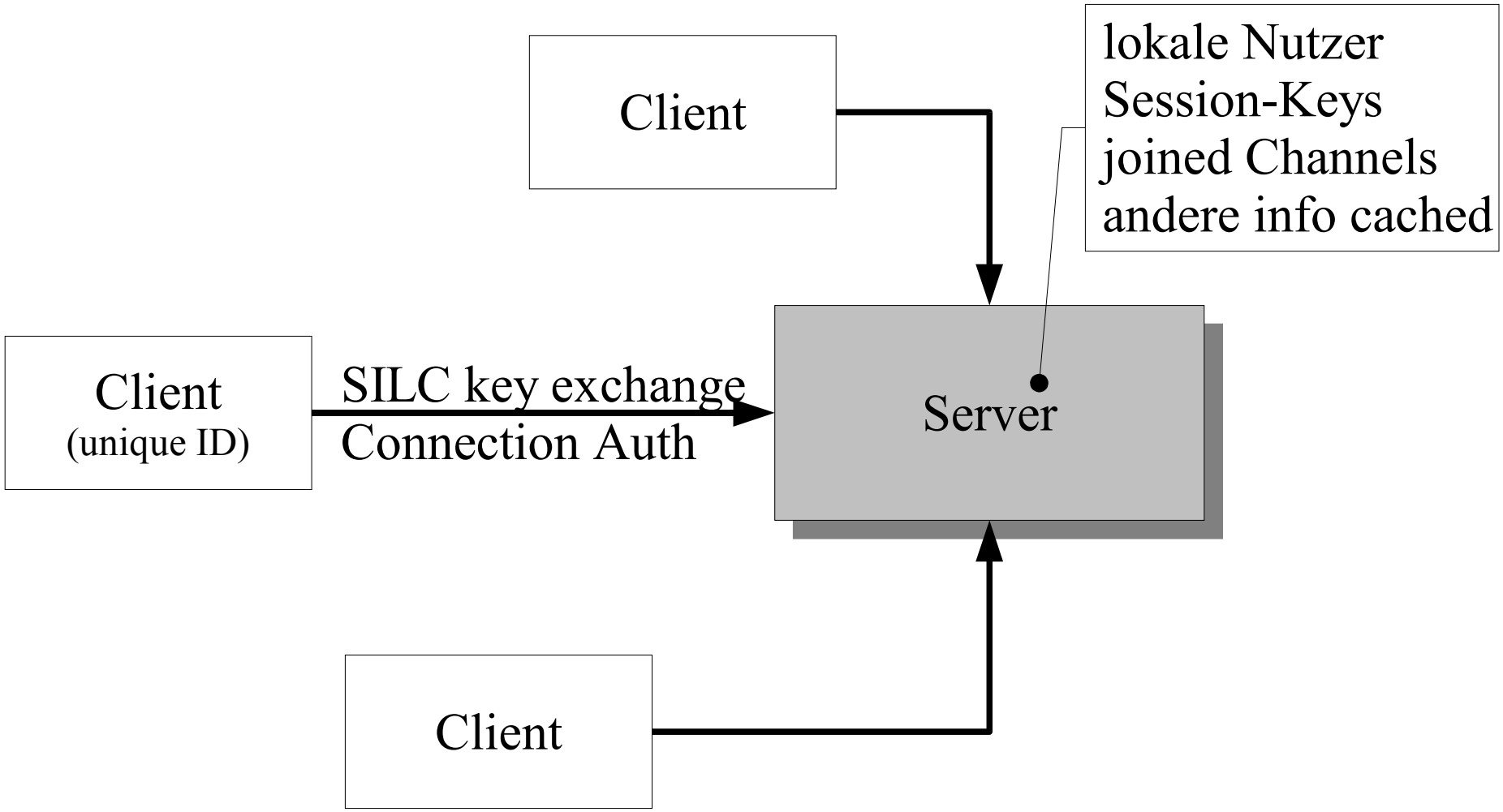
Protokoll

wichtigste Merkmale:

- **alle** Nachrichten sind **verschlüsselt** und **authentifiziert**
- Schlüssel von Server oder Nutzer verwaltet
- Nachrichten über Server-Netz geschickt
- Dateien mittels sftp (P2P)

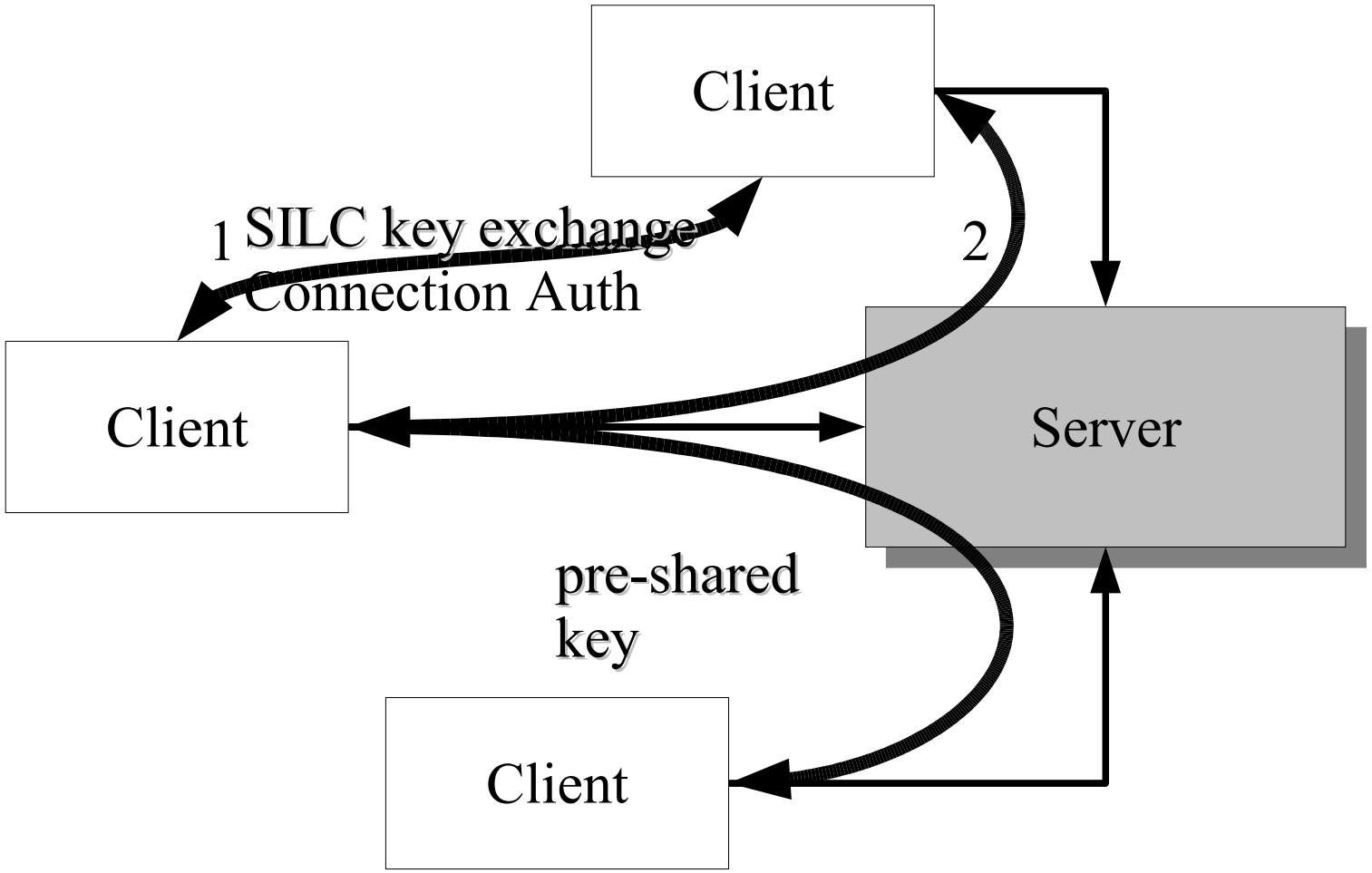
Netzwerk Topologie

normaler Client Traffic

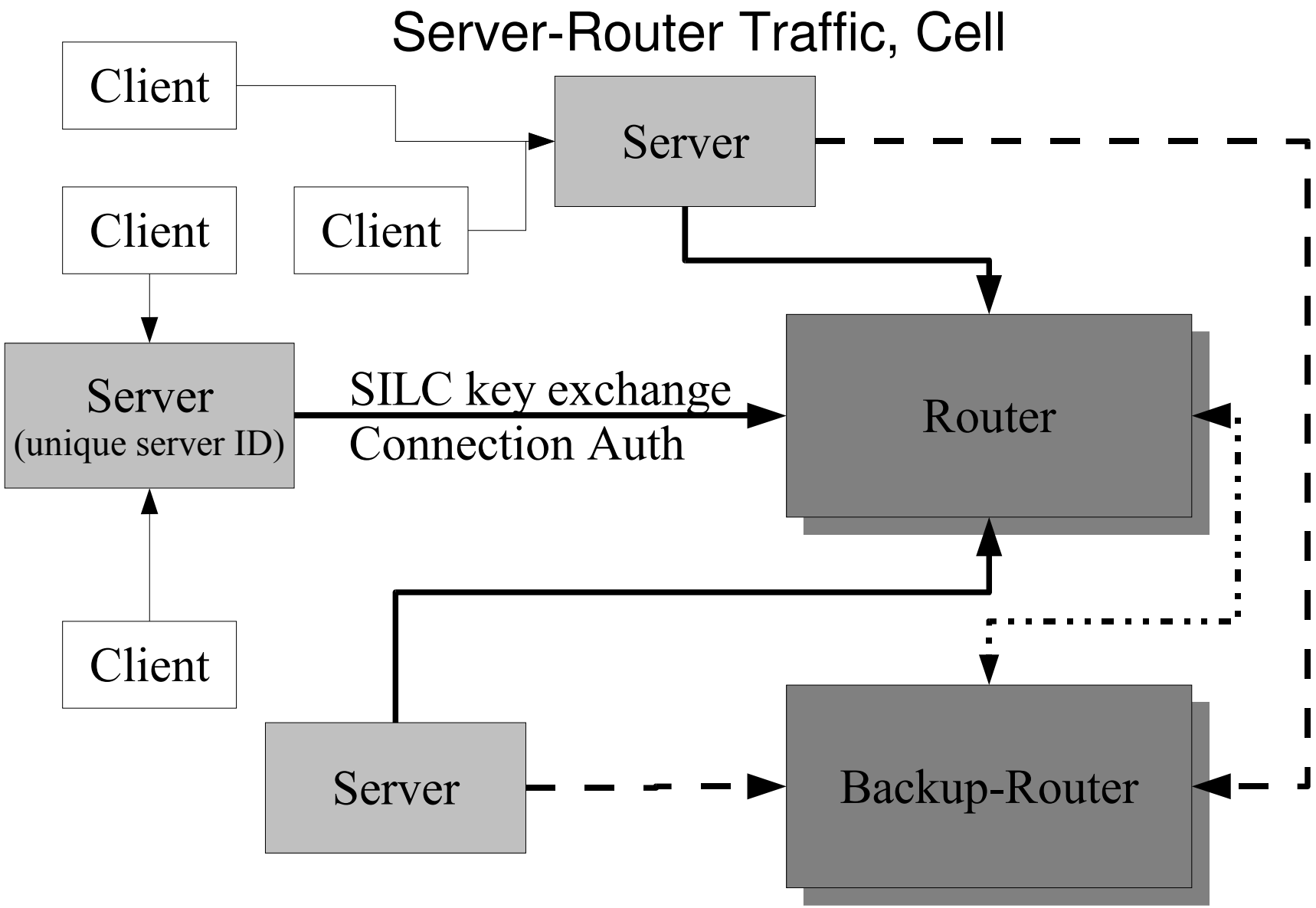


Netzwerk Topologie

Client-Client traffic

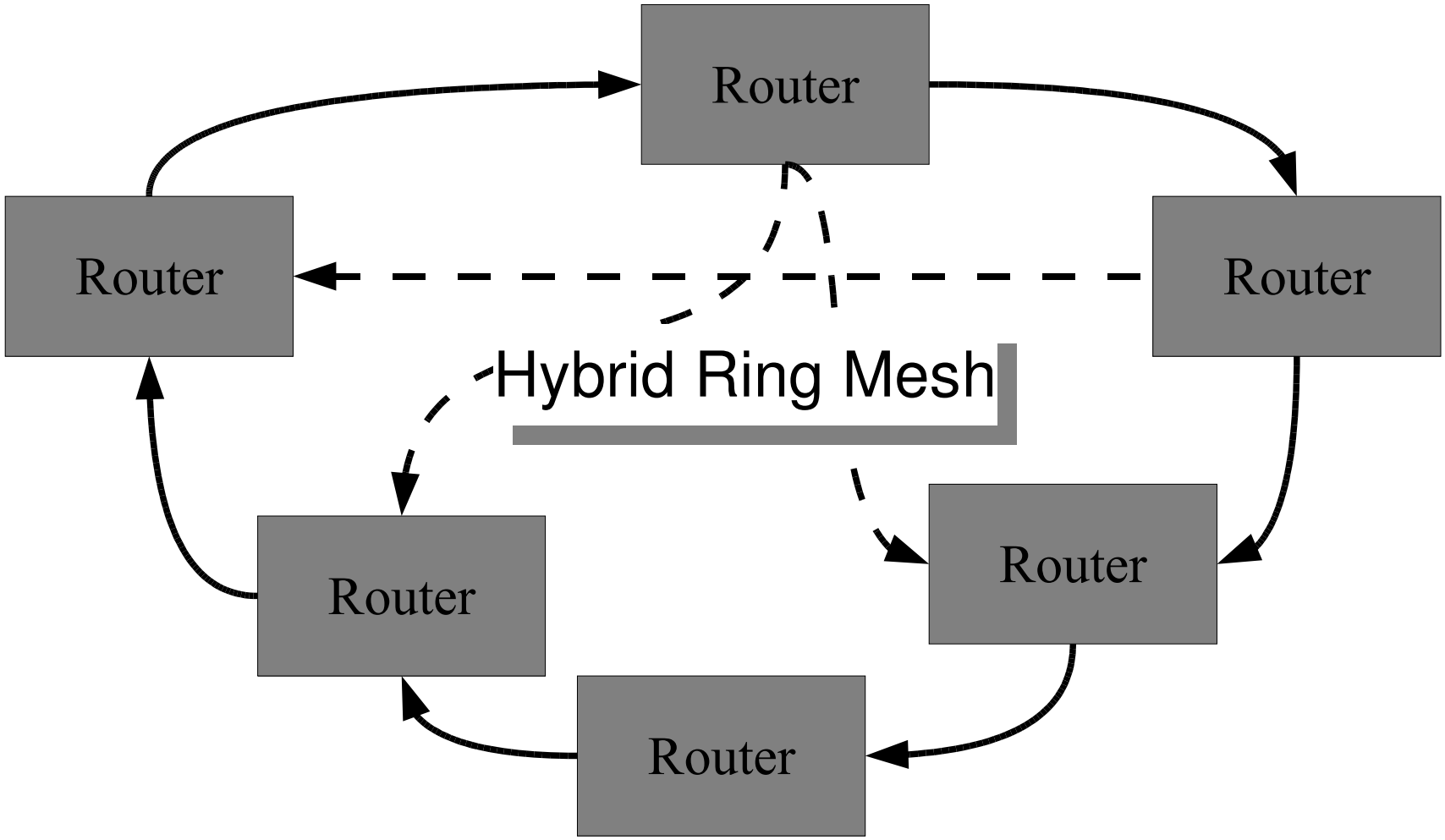


Netzwerk Topologie

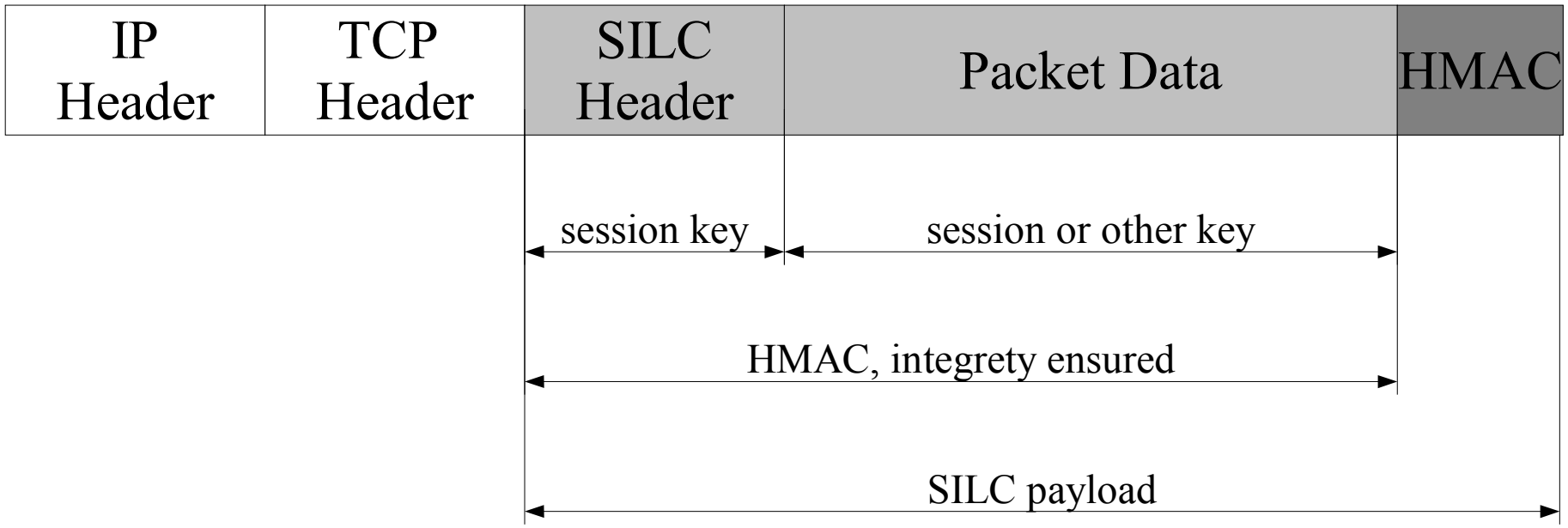


Netzwerk Topologie

Router-Router Traffic



Paket Format Überblick



SILC Key Exchange (SKE)

(1) Initiator sendet seine Methoden (cipher, hash Funktion, HMAC Funktion, Public Key Algorithmus)

(2)Antwortender wählt Methoden

(3)Diffie-Hellman Schlüsselaustausch
zusätzlich Austausch öffentlicher Schlüssel

(4)Mutual authentication mode

->: Session key

SILC Verbindungsauthentifizierung

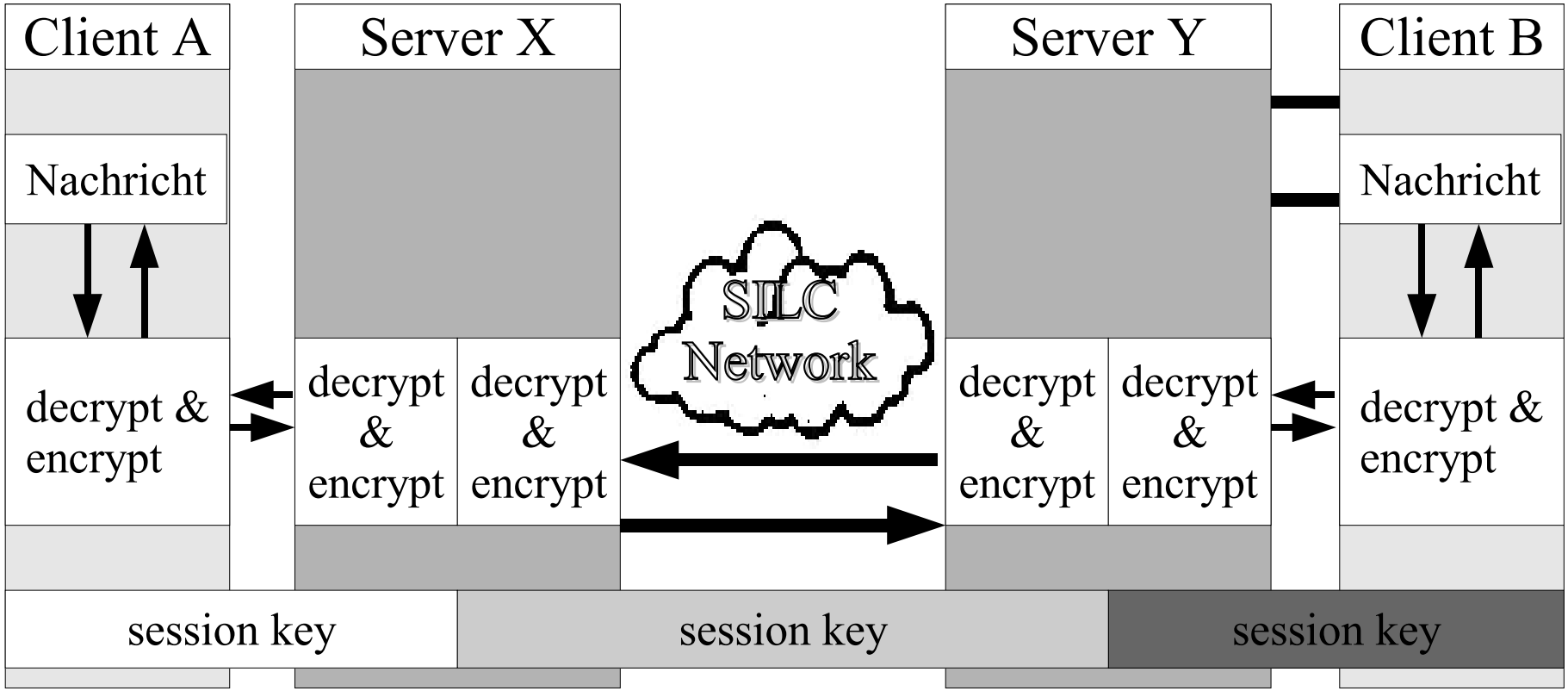
- erfolgt gleich nach SKE
- Autentifizierung der Verbindungsparteien (z.B. Client zu Server)
- basierend auf:

  Passphrase (packet encrypted) oder

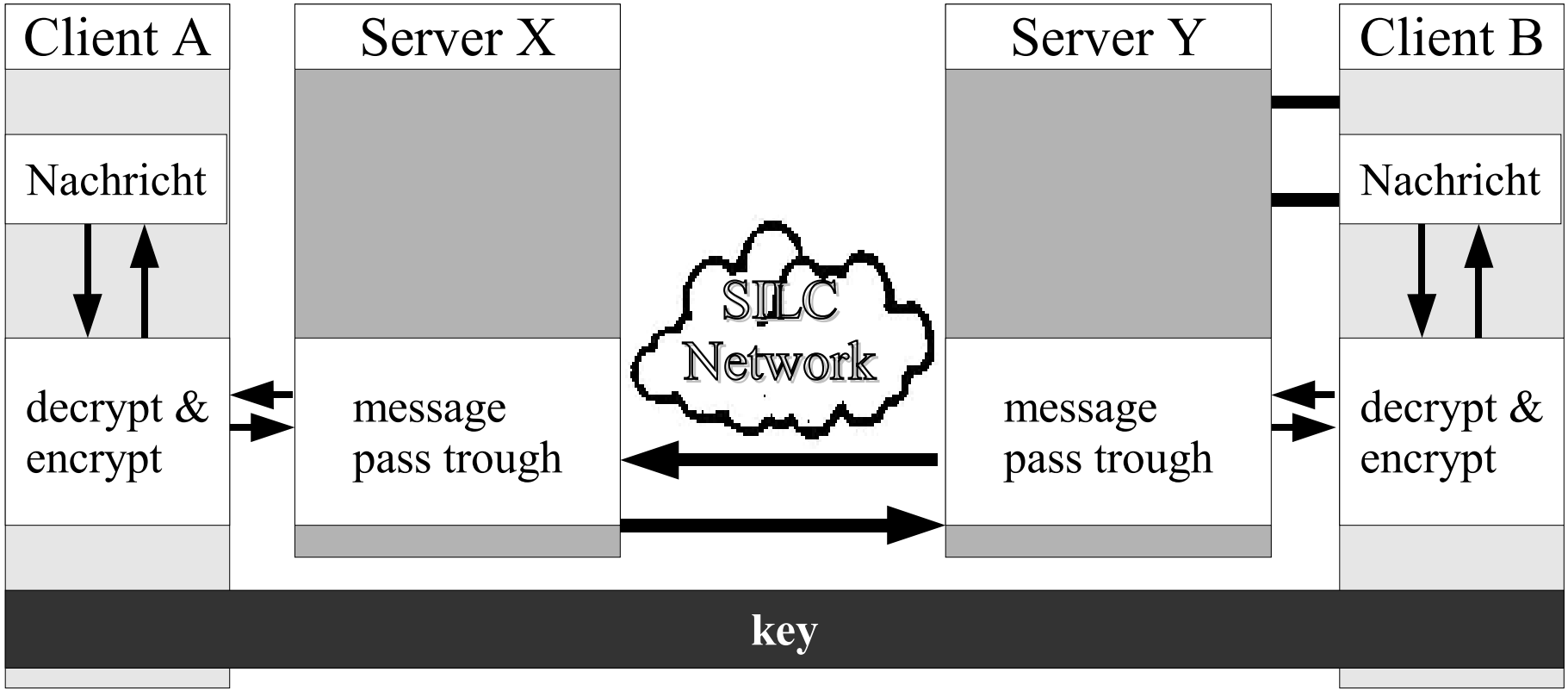
  Public-Key (Challenge an Client gesendet)

->: authentifizierter Client

Private Nachricht mit Session Keys



Private Nachricht mit Private Key



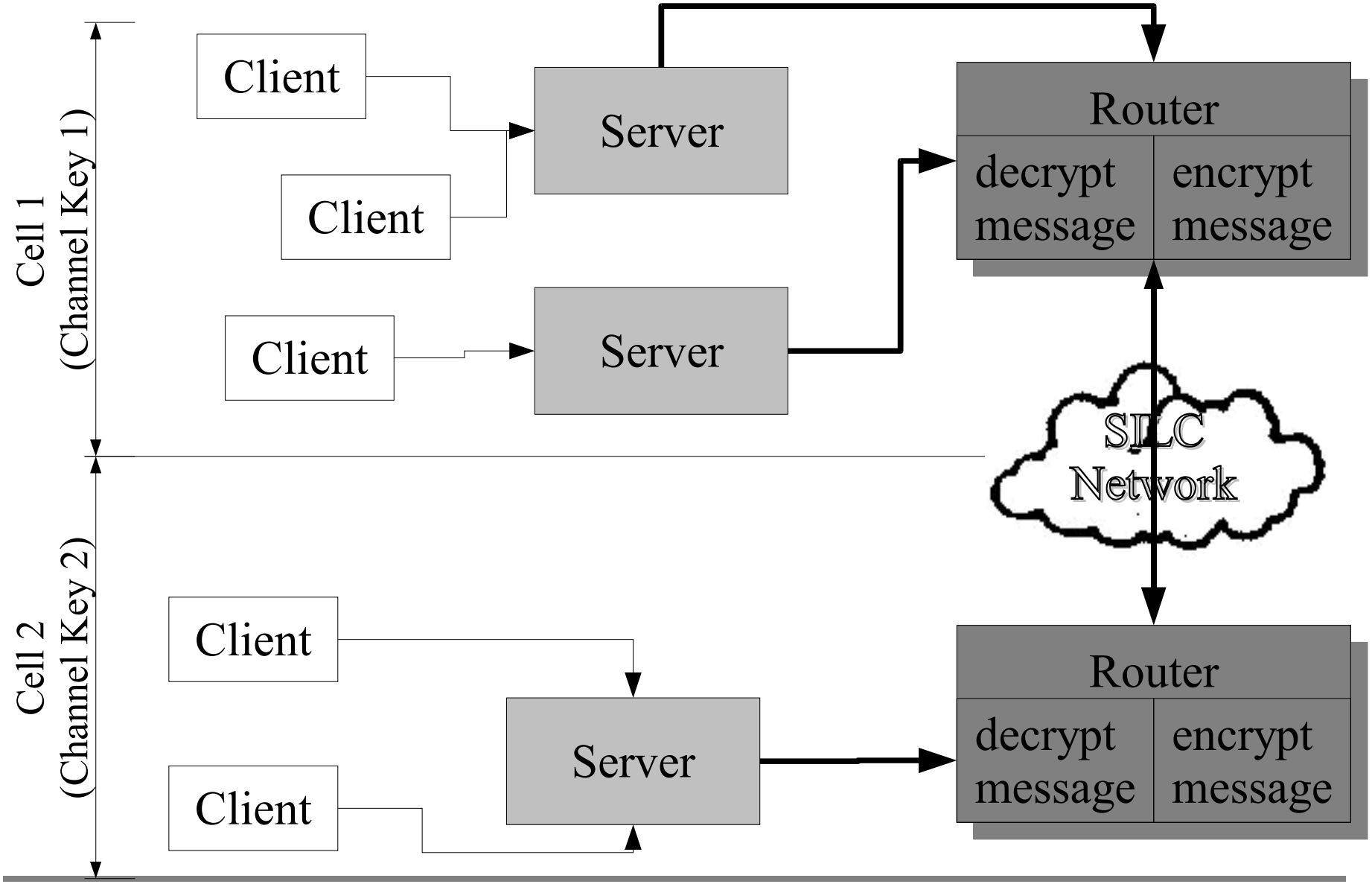
Channels

- benannte Gruppe, alle teilnehmenden Clients empfangen die selben Nachrichten
- Channel Namen sind eindeutig, max. 256 Zeichen

Channel Nachrichtenzustellung

- Alle Nachrichten in einer **Cell** sind verschlüsselt und authentifiziert mit einem Channel key
- Nachrichten zwischen Cells sind mit dem Session Key verschlüsselt
- regelmäßiges Re-keying
- zusätzl.: Key wird neu generiert, wenn Channel gegründet wird oder Client dazu kommt/verläßt
- Private Keys sind möglich(passphrase, publickey)

Channel Nachrichtenzustellung



SILC-Nutzung

SILC-Nutzung: Nicks

- keine eindeutigen Nickname
- keine Nick Services (um Nick wars zu verhindern)
- Authentifiziert und Identifiziert durch Public Key
- Attribute und Present Modi
- andere nützliche Modi
  - blocking non op msgs
  - blocking private msgs
  - marking / blocking bot msgs
  - reject watching

SILC-Nutzung: Channels

- joining und founding eines channels (Backup vom key!!!)

```
/cmode +f channel
```

- keine channel services nötig (takeovers sind sehr schwierig)
- kann privat, geheim, moderiert, nur invite, limitiert sein
- channel keys setzten

SILC-Nutzung: Channels mit Geheimnis

- key besorgen per mail, Telefon, ...

```
/JOIN channel

/CMODE +k

/KEY CHANNEL channelname set secret
```

- Muss von jedem Client gemacht werden
- => NutzerD (und Server Admin) kann keine Nachrichten mitlesen

SILC-Nutzung: Channel mit Public Channel Keys

- Key per Mail besorgen, `/getkey`, ...

```
/JOIN channel

/CMODE channel +C +pubkeyUserA \
    +pubkeyUserB +pubkeyUserC
```

- NutzerB und NutzerC join w/ `/JOIN channel -auth`
- => NutzerD (und Server Admin) können nicht mitlesen

SILC-Nutzung: Messaging

- Nachrichten können signiert werden (mit `/SMSG`)

```
[?] fukami: signed msg, you don't have the key
[S] fukami: signed msg, you've got the key
```

- MIME Nachrichten

```
/SCRIPT LOAD silc-mime.pl
/MMSG -CHANNEL channelname path/to/file
```

- private Nachrichten mit Key schützen

SILC-Nutzung: Absicherung privater Nachrichten

- Shared Secret per Telefon, Mail, Fax...

```
/KEY MSG UserA set secret
/KEY MSG UserB set secret
```

- => gesicherte Kommunikation

SILC-Nutzung: Datei-Transfer

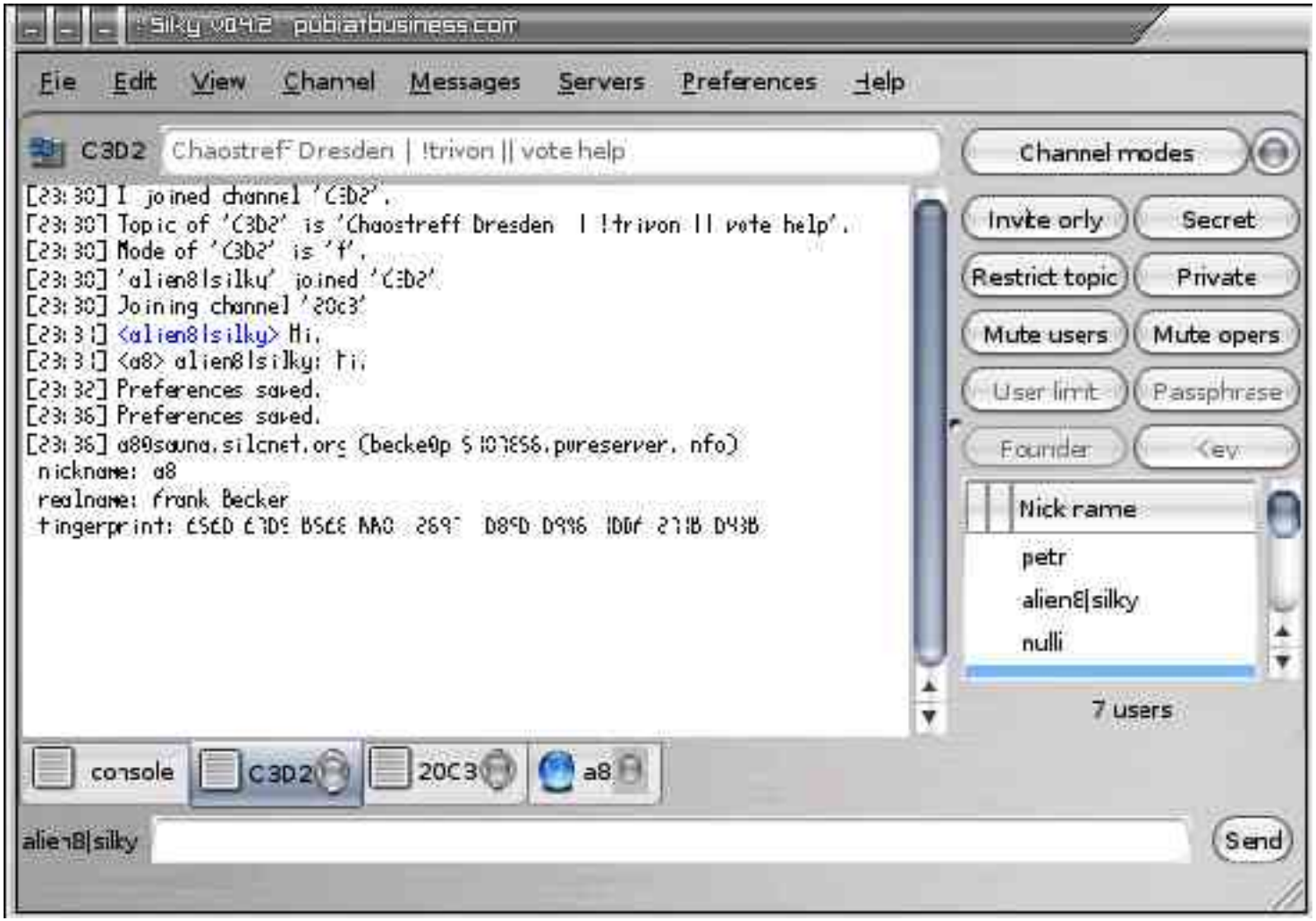
- Dateiaustausch (p2p) mit sftp

```
/FILE SEND path/to/file UserB
/FILE ACCEPT UserB
/FILE CLOSE
```
(um Session sofort zu beenden)

mit `-no-listener` falls hinter NAT

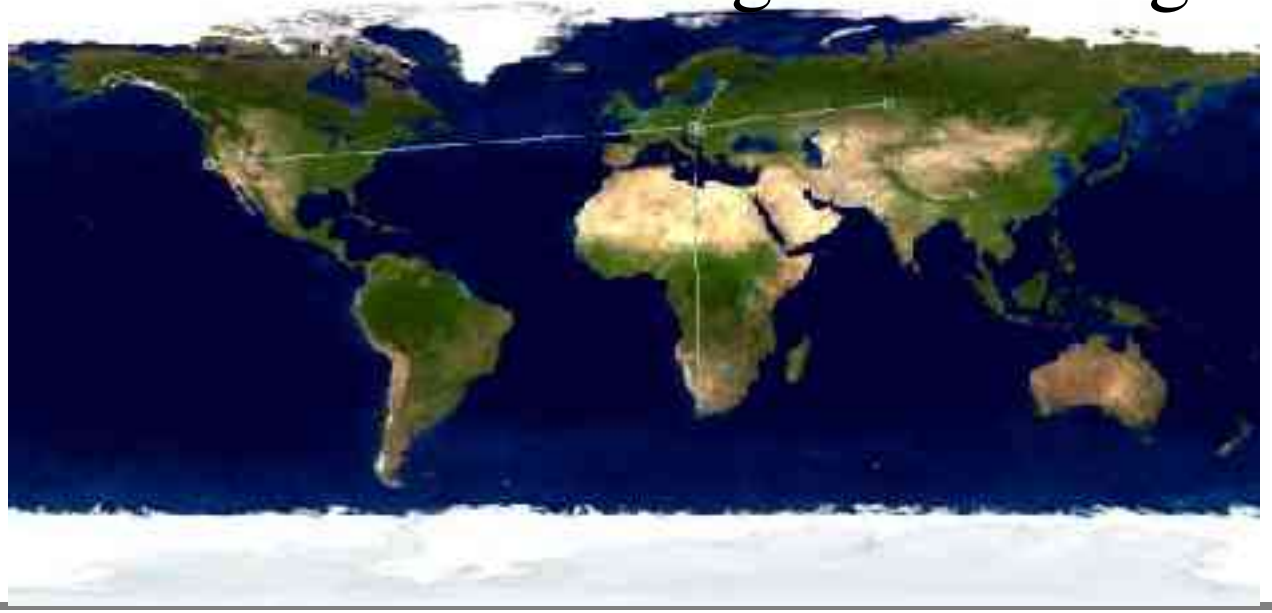
Programme und Frameworks

- silcd
- silc-client
- silc-toolkit
- Silky
- gaim
- jsilc
- samadhi



SILC Netzwerk

- Verbinde Dich zu einem Server in **Deiner Nähe**
- silc.silcnet.net macht Round-Robin DNS mit allen SILCNet Servern
- mehrere Server Verbindungen sind möglich



einige Credits

- Pekka Riikonen (Hauptentwickler, SILCNet Admin, gaim-Plugin)
- Timo "cras" Sirainen (Irssi/SILC client)
- Jochen "c0ffee" Eisinger (SILC plugin/ Silc-Client Maintainer)
- Toni Willberg (Silky)
- Giovanni Giacobbi (silcconfig,silclog, silcd bugfixes, RPM Pakete)
- Lubomir "salo" Sedlacik (NetBSD Pakete, Projekt Server Admin)
- Tamas Szerb (Debian Packete)
- Mika "Bostik" Boström (Man-pages, Fehlersuche)
- Juha Räsänen (ElGamal Implementierung)
- Ville Räsänen (Client-teil des STATS-Kommandos, Einige ROBXOdoc Formatierungen, Fehlersuche)
- Patrik Weiskircher (whois Attribute, Fehlersuche)

Links

- SILC-Seite: Anleitungen, FAQs, White Papers, vorkompilierte Pakete von silcd, silc-client (rpm, deb) und vieles mehr: http://silcnet.org
- Irssi-plugin: http://penguin-breeder.org/silc/
- Silky: http://silky.sf.net
- Ports for NetBSD, FreeBSD, OpenBSD, Darwin