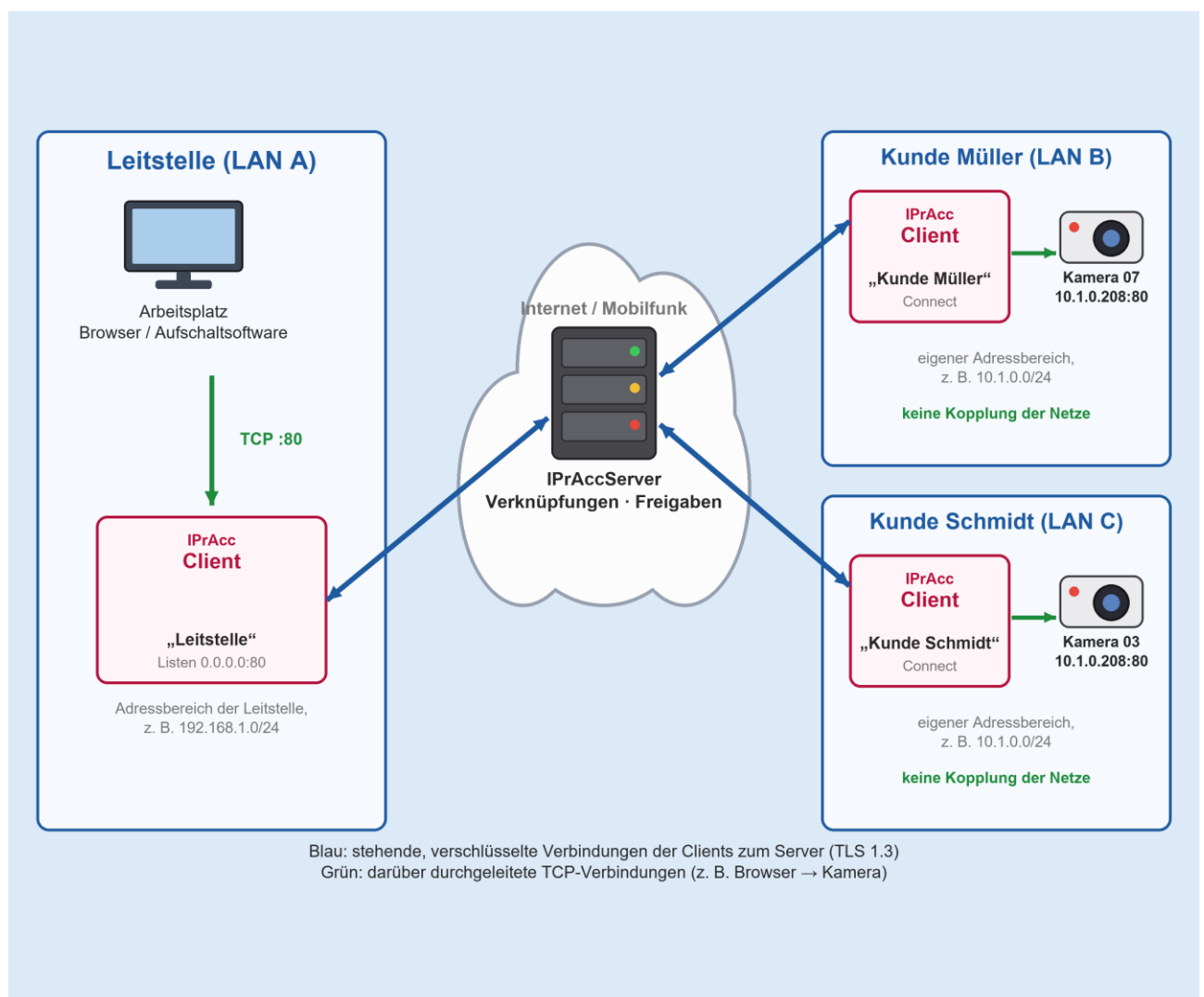


IPrAcc 2.0

Sichere IP-Verbindungen zwischen getrennten Netzen
IP remote Access auf Basis der MicroServices von Accellence



Status: Entwurf

Dieses Dokument ist geistiges Eigentum der Accellence Technologies GmbH und darf nur mit unserer ausdrücklichen Zustimmung verwendet, vervielfältigt oder weitergegeben werden

Inhalt

1	Einleitung	3
1.1	Ausgangslage und Aufgabe	3
1.2	Lösung: Server in der Leitstelle, Clients bei den Kunden	4
1.3	IPrAcc und VPN	5
1.4	Begriffe	6
1.5	Syntaxhinweise	6
2	Schnellstart	7
3	Der IPrAccServer in der Leitstelle	8
3.1	Server starten	9
3.2	Clients anlegen und Schlüssel verwalten	10
3.3	Alternative: Client mit Einmalcode anmelden	12
3.4	Zustand der Clients überwachen	14
3.5	Verbindungen der Clients	16
3.6	Verknüpfungen anlegen und überwachen	17
3.7	Empfangene Statusmeldungen anzeigen	20
3.8	Konfiguration des Servers	21
3.9	Quittung	24
3.10	Erreichbarkeit: Port und Firewall	24
3.11	Als Windows-Dienst installieren	25
4	Der IPrAccClient im Client-Netz	26
4.1	Client-Paket importieren	27
4.2	Alternative: Anmelden mit Einmalcode	28
4.3	Status und stehende Verbindung	29
4.4	Verbindungen und Freigaben	30
4.5	Inhalt der Statusmeldung	32
4.6	Konfiguration des Clients	33
4.7	Als Windows-Dienst installieren	34
5	Verbindungen nutzen: Durchleitung im Detail	35
5.1	Verbindungsaufbau	35
5.2	Mehrere Verbindungen	36
5.3	UDP und Umsetzung zwischen TCP und UDP	36
5.4	Datenübertragung und Grenzen	37
5.5	Anwendungsbeispiele	38
6	Anwendungsszenarien	39
6.1	Leitstelle mit vielen Kunden	39
6.2	Gleiche Adressbereiche bei mehreren Kunden	39
6.3	Gezielte Kopplung zweier Netze	39
6.4	Zwei Netze, die sich nicht erreichen können	39
6.5	Mehrere Server und Instanzen	40
7	Verwaltung mit dem MicroServiceKonfigurator	41
8	Format der Statusdateien	42
9	Erkennen und Melden von Systemausfällen	43
9.1	Lebenszeichen für AccRoutineMonitor	43
9.2	Störungsmeldung bei Ausfall eines Clients	44
10	Datensicherheit	45
11	Digitale Souveränität	48
12	Download und Updates	48
13	Lizenz und Freischaltung	48
13.1	Testmodus	48
13.2	Die Karte „Lizenz“	49
13.3	Lizenz beantragen	50
13.4	Lizenzschlüssel installieren	51
13.5	Was der Schlüssel festlegt	52
13.6	Gültigkeit, Ablauf und Rückfall	52
13.7	Hinweise für den Betrieb	52
14	Was tun wenn	53
15	Support / Hotline	55

1 Einleitung

Mit **IPrAcc** (**IP** remote **Access** von Accellence) lassen sich getrennte Netzwerke (LANs) sicher miteinander verbinden – gezielt, und nur dort, wo es gewollt ist. Eine Leitstelle kann damit zum Beispiel die **Kameras ihrer Kunden aufschalten**, ohne dass in den Kunden-netzen Ports geöffnet, Router umkonfiguriert oder Netze miteinander gekoppelt werden müssen.

IPrAcc besteht aus zwei Diensten: Der **IPrAcc Server** läuft an einer Stelle, die alle Beteiligten erreichen können (typischerweise in der Leitstelle). Ein **IPrAcc Client** läuft in jedem Netz, aus dem oder in das Verbindungen gewünscht sind – also bei jedem Kunden und in der Leitstelle selbst. Jeder Client baut eine **stehende, verschlüsselte Verbindung** zum Server auf und hält sie. Darüber werden die gewünschten TCP/IP-Verbindungen durchgeleitet.

Beide Dienste gehören zu den AccMicroServices und werden wie diese mit dem AccMicroServiceKonfigurator verwaltet. Sie sind für den weitgehend unbeaufsichtigten Dauerbetrieb als Windows-Dienst gedacht und haben eine Weboberfläche für Konfiguration und Überwachung.

Dieses Dokument beschreibt Einrichtung, Konfiguration und Bedienung von **IPrAcc Server** und **IPrAcc Client**. Wer es eilig hat, findet in Kapitel 2 einen Schnellstart; die Gliederung der weiteren Kapitel folgt dem empfohlenen Vorgehen: zuerst den Server in der Leitstelle einrichten (Kapitel 3), dann die Clients (Kapitel 4) und schließlich die Verbindungen nutzen (Kapitel 5).

1.1 Ausgangslage und Aufgabe

Leitstellen und andere Dienstleister müssen Systeme in fremden Netzen erreichen – zum Beispiel IP-Kameras und Videorecorder der Kunden, deren Liegenschaften bewacht werden sollen. Diese Systeme stehen aber selten dort, wo man sie direkt erreichen kann:

- Der Internetzugang des Kunden hat **keine öffentliche oder feste IPv4-Adresse** (z.B. DS-Lite, wechselnde Adressen).
- Das Kundennetz liegt **hinter einer Firewall oder einem NAT-Router**.
- Die Anbindung erfolgt **nur über Mobilfunk** (LTE, 5G).

Ein direkter Verbindungsaufbau von außen ist dann nicht möglich. Die üblichen Auswege haben jeweils Nachteile:

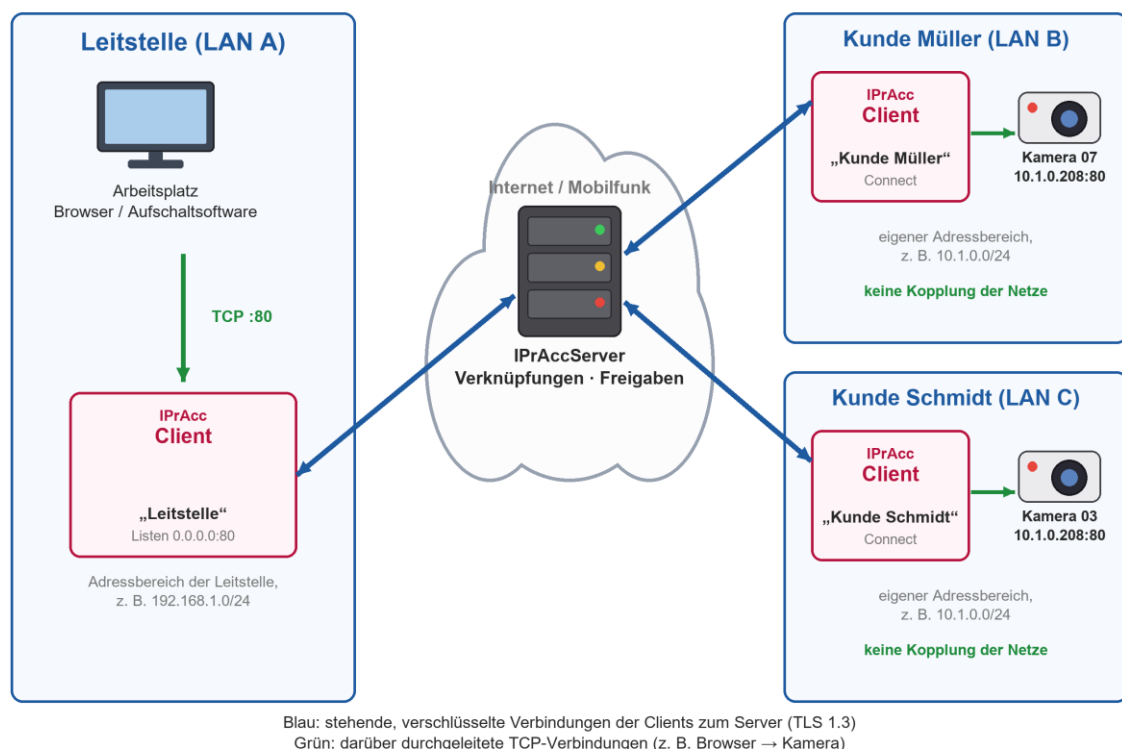
Möglichkeit	Vor- und Nachteile
Port-Forwarding	Einfach, aber der Port ist für jedermann im Internet offen – ein Sicherheitsrisiko. Funktioniert nicht ohne öffentliche IPv4-Adresse.
VPN	Sehr sicher, aber aufwendig einzurichten, besonders wenn viele Gegenstellen anzubinden sind. Koppelt ganze Netze: Dabei drohen Adresskonflikte und ungewollte Verbindungen zwischen den Netzen.
Hardware-Lösungen	Einfach und sicher, aber teuer; brauchen Platz und Strom, müssen vor Ort eingerichtet und gewartet werden.
Peer-to-Peer-Lösungen der Hersteller	Bequem, aber auf Produkte dieses Herstellers beschränkt; Sicherheit und Verfügbarkeit hängen vom Anbieter ab.

Gesucht war deshalb eine Lösung, die **einfach einzurichten, sicher und unabhängig von Herstellern und Netzbetreibern** ist – und die den Kunden die Kontrolle darüber lässt, welche Verbindungen in ihr Netz zugelassen werden.

1.2 Lösung: Server in der Leitstelle, Clients bei den Kunden

Der **IPrAcc Server** wird an einer Stelle betrieben, die alle Beteiligten erreichen können – z.B. in der Leitstelle. Dafür muss nur **ein einziger TCP/IP-Port** von außen erreichbar sein. In jedem Netz, das angebunden werden soll, läuft ein **IPrAcc Client**. Der entscheidende Trick: Der Client baut die Verbindung zum Server **selbst von innen nach außen auf** – das ist praktisch in jedem Netz erlaubt, auch hinter NAT, Firewall oder Mobilfunk. Diese Verbindung bleibt dauerhaft bestehen und wird nach Störungen automatisch wieder aufgebaut.

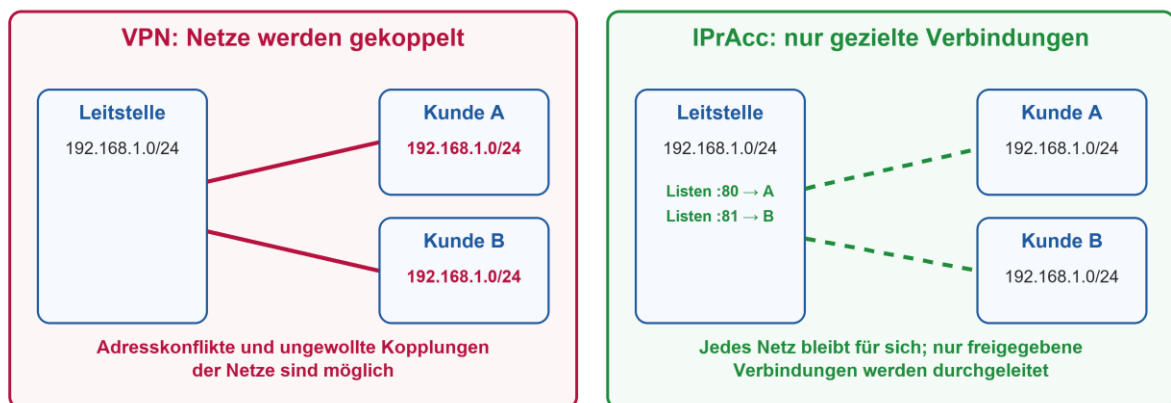
Über diese stehenden Verbindungen können dann „rückwärts“ beliebige TCP/IP-Verbindungen durchgeleitet werden: vom Arbeitsplatz in der Leitstelle zur Kamera im Kundennetz.



Auf der Kundenseite ist dafür **kein eingehender Port** nötig. In der Leitstelle sieht man nur die Adresse des Listen-Ports, den der Leitstellen-Client anbietet – die interne IP-Adresse der Kamera im Kundennetz bleibt unsichtbar.

1.3 IPrAcc und VPN

IPrAcc leitet Verbindungen **auf Anwendungsebene (OSI-Schicht 7)** weiter, nicht auf der Ebene der Netzwerke (Schicht 3) wie ein VPN. Es koppelt keine Netze, sondern reicht gezielt einzelne TCP/IP-Verbindungen durch, die jeweils durch **IP-Adresse und Port** der Gegenstelle beschrieben sind. Wie ein VPN ermöglicht **IPrAcc** damit sichere Verbindungen zwischen verschiedenen LANs – es ist aber wesentlich einfacher zu konfigurieren, und es vermeidet typische VPN-Probleme:



- **Keine Adresskonflikte:** Jeder Client arbeitet im Adressbereich seines eigenen LANs. Mehrere Kunden dürfen dieselben Adressen verwenden (z.B. alle 192.168.1.0/24), ohne dass sich etwas überschneidet.
- **Keine ungewollten Netzwerkkopplungen:** Es werden nur die freigegebenen Verbindungen durchgeleitet, nicht ganze Netze. Ein Kunde kann nie das Netz eines anderen Kunden erreichen.
- **Einfach:** Eine Verbindung besteht aus Namen, IP-Adresse und Port – drei Felder und ein Haken für die Freigabe. Es sind keine Zertifikate, Tunnelnetze oder Routen zu verwalten.
- **Kontrolle beim Kunden:** Der Administrator jedes LANs entscheidet am eigenen Client, welche Verbindungen er freigibt, und kann sie jederzeit wieder sperren.
- **Nur ausgehende Verbindungen:** Die Clients bauen die Verbindung zum Server selbst auf; in den Kundennetzen muss nichts von außen erreichbar sein.

Hinweis:

- **Durchgeleitet** werden **TCP- und UDP-Verbindungen** (Abschnitt 5.3).
- **Nicht durchgeleitet** werden **Broadcast** und **Multicast** sowie Protokolle ohne IP-Ports (z.B. **ICMP/Ping**).

1.4 Begriffe

Begriff	Bedeutung
IPrAcc Server	Der Dienst in der Leitstelle. Nimmt die stehenden Verbindungen der Clients an, verwaltet Clients, Freigaben und Verknüpfungen und leitet die freigegebenen Verbindungen durch.
IPrAcc Client	Der Dienst in jedem angebundenen Netz. Hält die stehende Verbindung zum Server, meldet seine Verbindungen und stellt sie – nach Freigabe – zur Verfügung.
Verbindung	Ein Eintrag in der Verbindungstabelle eines Clients: Name (z.B. „Kamera 07: Videostream“), IP-Adresse, Port, Protokoll (TCP oder UDP) und Art. Jede Verbindung hat vom Client eine UUID, die auch bei geändertem Namen oder geänderter Adresse dieselbe bleibt.
Art: Listen	Der Client öffnet einen Port in seinem LAN und nimmt dort nacheinander beliebig viele Verbindungen an (z.B. von Arbeitsplätzen der Leitstelle). IP-Adresse 0.0.0.0 bedeutet: auf allen Netzwerkschnittstellen des PCs.
Art: Connect	Der Client baut auf Anforderung eine Verbindung zu IP-Adresse und Port in seinem LAN auf (z.B. zur Kamera).
Protokoll	TCP (Standard) oder UDP : das Protokoll, das dieser Client an seinem Ende der Verbindung verwendet. Die beiden Enden einer Verknüpfung sind unabhängig: Ein UDP-Listen-Port kann mit einem TCP-Ziel verknüpft werden und umgekehrt – so lässt sich UDP in TCP umsetzen und TCP in UDP.
Freigabe	Der Haken, mit dem der Administrator des Client-Netzes eine Verbindung bewusst zur Nutzung freigibt. Ohne Freigabe wird die Verbindung nur gemeldet, aber nie benutzt.
Verknüpfung	Vom Server-Administrator angelegte Zuordnung: Der Listen-Port eines Clients wird mit einer Connect-Verbindung eines anderen Clients verbunden. Nur so entstehen durchgeleitete Verbindungen.
Listen-Richtlinie	Die Festlegung des Server-Administrators, welche Clients Listen-Ports anbieten dürfen und auf welchen Ports (Abschnitt 3.4). Der Server setzt sie durch und teilt sie den Clients mit.
Client-Paket / Einmalcode	Zwei Wege, einen neuen Client mit dem Server bekannt zu machen und ihm seinen geheimen Schlüssel zu geben (Abschnitte 3.2 und 3.3).
Statusmeldung	Kurze Meldung, die jeder Client regelmäßig über die stehende Verbindung sendet (Lebenszeichen mit IP, Zeitpunkt und Zustand) und die der Server quittiert und in Statusdateien ablegt.

1.5 Syntaxhinweise

<x> Platzhalter, für den konkrete Werte eingesetzt werden müssen.

→ Verweis auf weitere Dokumente oder Textstellen.

Schaltfläche Bezeichnung einer Schaltfläche oder eines Bereichs in einer Weboberfläche.

Gezeigt sind die Weboberflächen im Standardaufbau; je nach Systemkonfiguration können sie auch anders aussehen.

Hinweis: Geprüfte Plattform ist Windows. Der Quellcode ist plattformneutral gehalten, damit Linux und weitere Plattformen später ohne Umbau folgen können.

2 Schnellstart

Dieses Kapitel zeigt an einem Beispiel, wie Sie in wenigen Schritten die erste Verbindung herstellen. Die Leitstelle möchte die **Kamera 07** des Kunden **Müller** (IP-Adresse 10.1.0.208, Port 80 im Netz des Kunden) in ihrem Browser aufschalten. Jeder Schritt ist in den späteren Kapiteln ausführlich beschrieben.

1. **Server starten** (Abschnitt 3.1): IPrAccServer.exe in der Leitstelle ausführen und dafür sorgen, dass der Port 5050 von den Clients aus erreichbar ist (Abschnitt 3.10).
2. **Clients anlegen** (Abschnitt 3.2): In der Server-Weboberfläche im Bereich **Clients** zwei Clients anlegen – „Leitstelle“ und „Kunde Müller“ – und jeweils das **Client-Paket** speichern.
3. **Clients installieren und verbinden** (Abschnitt 4.1): IPrAccClient.exe auf einem PC im Netz der Leitstelle und auf einem PC im Netz des Kunden starten und dort das jeweilige Client-Paket importieren. Der Client zeigt **verbunden**, der Server zeigt den Client als **Verbunden**.
4. **Kamera beim Kunden freigeben** (Abschnitt 4.4): In der Weboberfläche des Clients „Kunde Müller“ im Bereich **Verbindungen** eine Zeile mit Name Kamera 07: Videostream, IP-Adresse 10.1.0.208, Port 80, Protokoll **TCP** und Art **Connect** eintragen, **Freigabe** ankreuzen und auf **Speichern und melden** klicken.
5. **Listen-Port in der Leitstelle freigeben** (Abschnitt 4.4): In der Weboberfläche des Clients „Leitstelle“ eine Zeile mit Name Leitstelle:80, IP-Adresse 0.0.0.0, Port 80, Protokoll **TCP** und Art **Listen** eintragen, **Freigabe** ankreuzen und auf **Speichern und melden** klicken.
6. **Verknüpfung anlegen** (Abschnitt 3.6): In der Server-Weboberfläche im Bereich **Verknüpfungen** einen Namen eingeben, als Listen-Port „Leitstelle: Leitstelle:80“ und als Gegenstelle „Kunde Müller: Kamera 07: Videostream“ wählen und auf **Verknüpfung anlegen** klicken. Der Zustand der Verknüpfung wechselt auf **bereit**.
7. **Testen**: Im Browser auf einem Arbeitsplatz der Leitstelle `http://<IP-Adresse des PCs mit dem Leitstellen-Client>:80/` aufrufen. Sie sehen die Weboberfläche der Kamera 07 – obwohl diese im Kundennetz nicht direkt erreichbar ist.

Die Zeilen der Verbindungstabellen im Überblick:

Client	Verbindungsname	IP-Adresse	Port	Protokoll	Art	Freigabe
Kunde Müller	Kamera 07: Videostream	10.1.0.208	80	TCP	Connect	ja
Leitstelle	Leitstelle:80	0.0.0.0	80	TCP	Listen	ja

Hinweis: Bei **Listen** legt die IP-Adresse fest, auf welcher Netzwerkschnittstelle des PCs der Port geöffnet wird: 0.0.0.0 auf allen (erreichbar für alle Arbeitsplätze des Leitstellennetzes), 127.0.0.1 nur für den PC selbst, oder die Adresse einer bestimmten Schnittstelle. Wählen Sie den kleinstmöglichen Kreis, der benötigt wird.

Ohne **Lizenz** läuft der Server im **Testmodus** mit max. 2 gleichzeitig verbundenen Clients. Wenn Sie mehr möchten → Kapitel 13.

3 Der IPrAccServer in der Leitstelle

Nach der Installation von EBÜS finden Sie den **IPrAcc** Server im Verzeichnis

```
C:\EBUES\AccMicroServices\IPrAccServer
```

Der Ordner enthält neben der Anwendung (IPrAccServer.exe) zwei Textdateien:

- IPrAccServer.cfg – die Konfiguration (Port, Schutzgrenzen, Statusverzeichnis, ...)
- message.inf – die Vorlage der Quittung, die jeder Client nach seiner Statusmeldung zurückbekommt

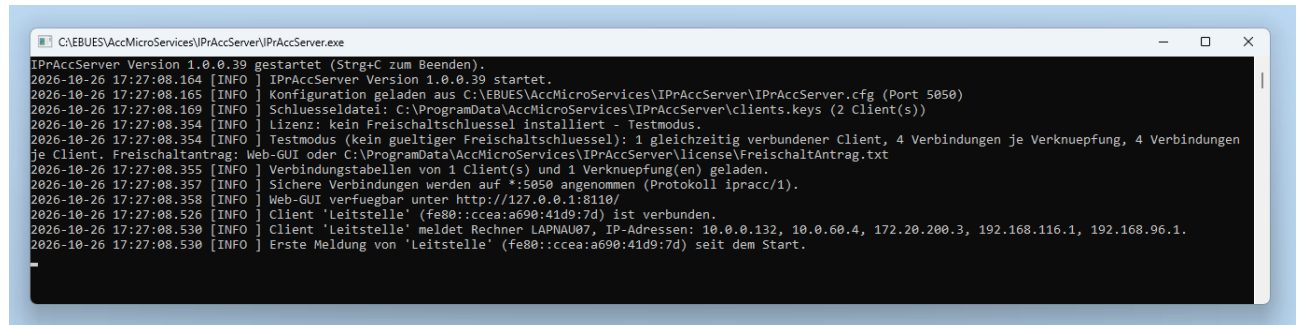
Beide Dateien lassen sich bequem über die Weboberfläche bearbeiten. Zusätzlich legt der Server außerhalb des Programmordners, im Ordner

```
C:\ProgramData\AccMicroServices\IPrAccServer
```

drei geschützte Dateien an: die **Schlüssel** der Clients (clients.keys), die zuletzt von den Clients gemeldeten **Verbindungstabellen** (connections.tables) und die **Verknüpfungen** (links.tbl).

3.1 Server starten

Führen Sie IPrAccServer.exe durch Doppelklick aus. Es öffnet sich ein Konsolenfenster, in dem alle Status- und Fehlermeldungen erscheinen:

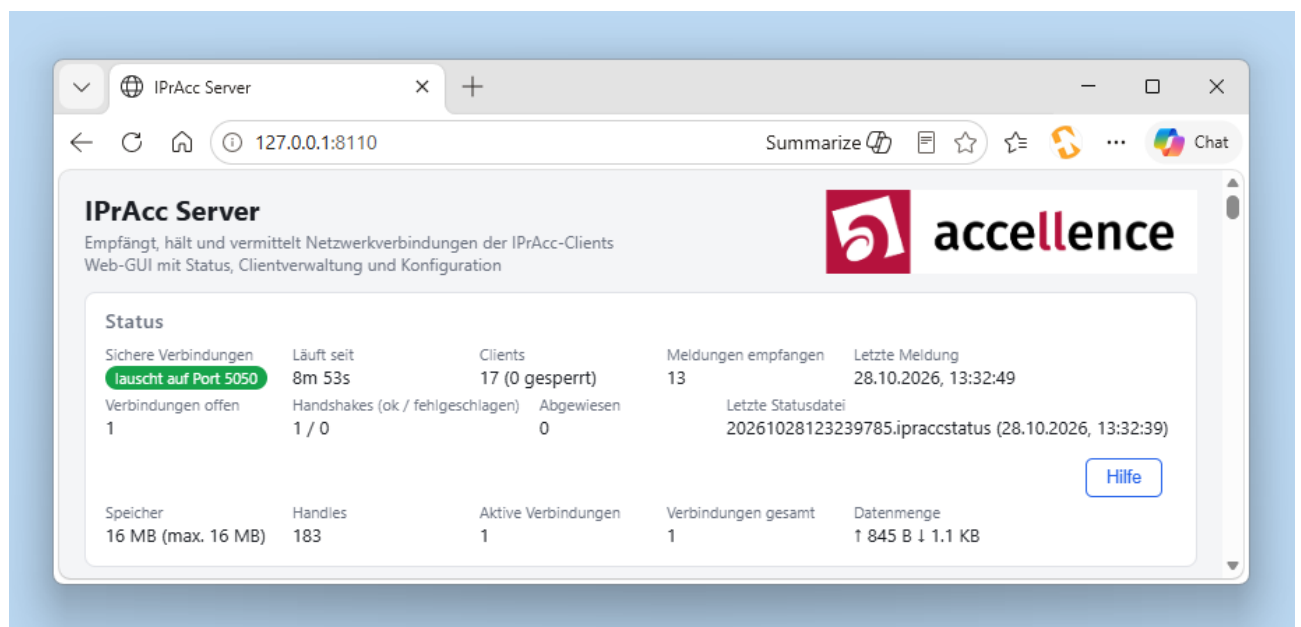


```

C:\EBUES\AccMicroServices\IPrAccServer\IPrAccServer.exe
IPrAccServer Version 1.0.0.39 gestartet (Strg+C zum Beenden).
2026-10-26 17:27:08.164 [INFO] IPrAccServer Version 1.0.0.39 startet.
2026-10-26 17:27:08.165 [INFO] Konfiguration geladen aus C:\EBUES\AccMicroServices\IPrAccServer\IPrAccServer.cfg (Port 5050)
2026-10-26 17:27:08.169 [INFO] Schlüsseldatei: C:\ProgramData\AccMicroServices\IPrAccServer\clients.keys (2 Client(s))
2026-10-26 17:27:08.354 [INFO] Lizenz: kein Freischaltsschlüssel installiert - Testmodus.
2026-10-26 17:27:08.354 [INFO] Testmodus (kein gültiger Freischaltsschlüssel): 1 gleichzeitig verbundener Client, 4 Verbindungen je Verknuepfung, 4 Verbindungen je Client. Freischaltantrag: Web-GUI oder C:\ProgramData\AccMicroServices\IPrAccServer\license\FreischaltAntrag.txt
2026-10-26 17:27:08.355 [INFO] Verbindungstabellen von 1 Client(s) und 1 Verknuepfung(en) geladen.
2026-10-26 17:27:08.357 [INFO] Sichere Verbindungen werden auf *:5050 angenommen (Protokoll ipracc/1).
2026-10-26 17:27:08.358 [INFO] Web-GUI verfuegbar unter http://127.0.0.1:8110/
2026-10-26 17:27:08.526 [INFO] Client 'Leitstelle' (fe80::ccea:a690:41d9:7d) ist verbunden.
2026-10-26 17:27:08.530 [INFO] Client 'Leitstelle' meldet Rechner LAPNAU07, IP-Adressen: 10.0.0.132, 10.0.60.4, 172.20.200.3, 192.168.116.1, 192.168.96.1.
2026-10-26 17:27:08.530 [INFO] Erste Meldung von 'Leitstelle' (fe80::ccea:a690:41d9:7d) seit dem Start.
  
```

Der Server wird mit **Strg+C** im Konsolenfenster wieder beendet. Für den Dauerbetrieb installieren Sie ihn als Windows-Dienst (Abschnitt 3.11).

Ist in der Konfiguration `gui autostart=1` gesetzt (werkseitig), öffnet sich zusätzlich die Weboberfläche des Servers im Browser (Standard: `http://127.0.0.1:8110/`):

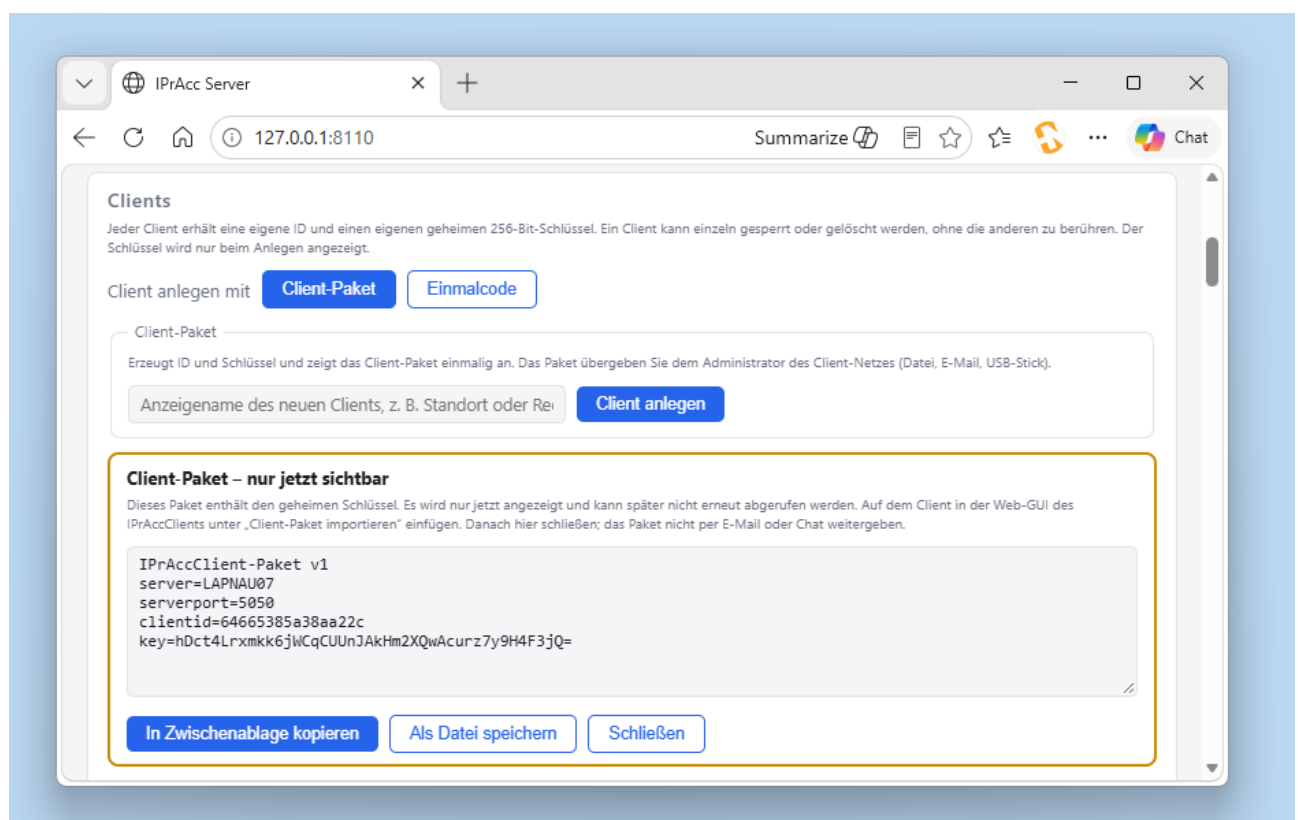


Der Bereich **Status** zeigt auf einen Blick, ob der Server für sichere Verbindungen bereitsteht (Port), seit wann er läuft, wie viele Clients angelegt sind, wie viele Statusmeldungen empfangen wurden, wie viele Verbindungen offen sind und wie viele Verbindungsversuche abgewiesen wurden (z.B. wegen einer Sperrliste, eines Verbindungslimits oder falscher Schlüssel). Kann der Port nicht geöffnet werden, erscheint hier die Ursache; der Server versucht es automatisch erneut.

3.2 Clients anlegen und Schlüssel verwalten

Jedes Netz, das angebunden werden soll – auch die Leitstelle selbst, wenn von dort Verbindungen aufgebaut werden –, ist im Server als Client angelegt und erhält dabei eine eigene ID und einen eigenen geheimen 256-Bit-Schlüssel. Zum Anlegen stehen zwei gleichwertige Wege zur Verfügung: Hinter „Client anlegen mit“ wählen Sie den Button Client-Paket (dieser Abschnitt) oder Einmalcode (Abschnitt 3.3). Ein Klick auf einen der Buttons öffnet die zugehörigen Eingabefelder; ein weiterer Klick oder ein Klick auf den anderen Button schließt sie wieder, sodass stets nur die Felder der gewählten Methode sichtbar sind.

1. Im Bereich **Clients** der Server-Weboberfläche hinter dem Text „Client anlegen mit“ auf den Button **Client-Paket** klicken und im geöffneten Feld einen **Anzeigenamen** eingeben (z.B. den Namen des Kunden oder Standorts)
2. Auf **Client anlegen** klicken.
3. Daraufhin wird das **Client-Paket** angezeigt: ein kurzer Text mit Serveradresse, Port, Client-ID und Schlüssel. Mit **In Zwischenablage kopieren** oder **Als Datei speichern** sichern Sie es und geben es dem Administrator des Client-Netzes. Dort wird es im Client importiert (Abschnitt 4.1).



Hinweis: Das Client-Paket enthält den **geheimen Schlüssel**. Es wird nur beim Anlegen angezeigt und kann später **nicht erneut abgerufen** werden – geht es verloren, legen Sie einen neuen Client an und löschen den alten. Geben Sie das Paket nicht unverschlüsselt per E-Mail oder Chat weiter.

Aus Sicherheitsgründen lassen sich Client-Schlüssel nur erzeugen, solange die Web-oberfläche des Servers auf 127.0.0.1 beschränkt ist (Standard); andernfalls würde das Paket unverschlüsselt über das Netz gehen. Der Schlüssel steht **nicht** in der Konfigurations-

datei, sondern in einer Schlüsseldatei, auf die nur SYSTEM, die Administratoren und das Dienstkonto zugreifen können:

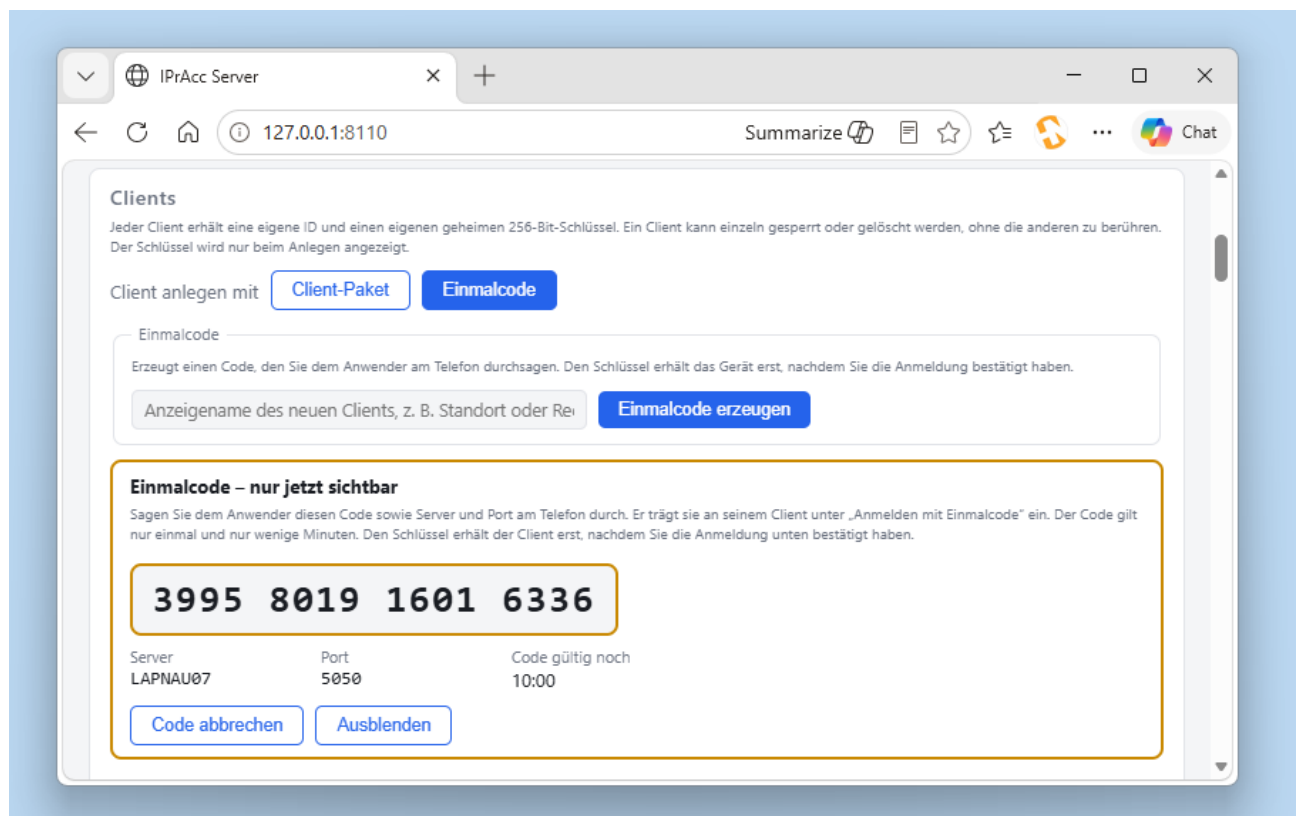
```
C:\ProgramData\AccMicroServices\IPrAccServer\clients.keys
```

(Bei einer benannten Instanz heißt der Ordner IPrAccServer.<Instanzname>.)
Behandeln Sie diese Datei – und eine Sicherung davon – wie ein Passwort.

3.3 Alternative: Client mit Einmalcode anmelden

Lässt sich das Client-Paket nicht übertragen – zum Beispiel weil der Client in einer Kamera, einem Videorecorder oder einem Alarmsystem steckt, das keine E-Mails empfangen kann und keine USB-Sticks zulässt –, melden Sie den Client stattdessen mit einem Einmalcode an. Dafür genügt ein Telefonat: Es muss nichts übertragen oder kopiert werden. Das Client-Paket (Abschnitt 3.2) bleibt daneben unverändert möglich.

1. Im Bereich **Clients** der Server-Weboberfläche hinter dem Text „Client anlegen mit“ auf den Button **Einmalcode** klicken und im geöffneten Feld einen **Anzeigenamen** eingeben (leer lassen, dann übernimmt der Server den Rechnernamen des Geräts) und auf **Einmalcode erzeugen** klicken.
2. Es erscheint ein 16-stelliger Code (z.B. 4821 0937 5516 2084) mit Server, Port und der Restzeit (10 Minuten). Sagen Sie dem Anwender Code, Server und Port am Telefon durch. Der Code gilt nur einmal.
3. Der Anwender trägt sie an seinem Client unter **Anmelden mit Einmalcode** ein (Abschnitt 4.2) und nennt Ihnen die dort angezeigte **Bestätigungszahl** (4 Ziffern).
4. Im Bereich **Anmeldungen mit Einmalcode** erscheint die wartende Anmeldung mit Quelle (IP), Rechnernamen und Bestätigungszahl; im Titel der Browser-Registerkarte steht die Zahl der wartenden Anmeldungen, z.B. „(1)“. Vergleichen Sie die Zahl, und klicken Sie nur dann auf **Bestätigen**, wenn sie übereinstimmt und Sie gerade mit dem Anwender sprechen. Im Zweifel: **Ablehnen**.
5. Erst jetzt erzeugt der Server den dauerhaften Schlüssel und überträgt ihn verschlüsselt an den Client. Der Client erscheint in der Liste der Clients und verbindet sich.



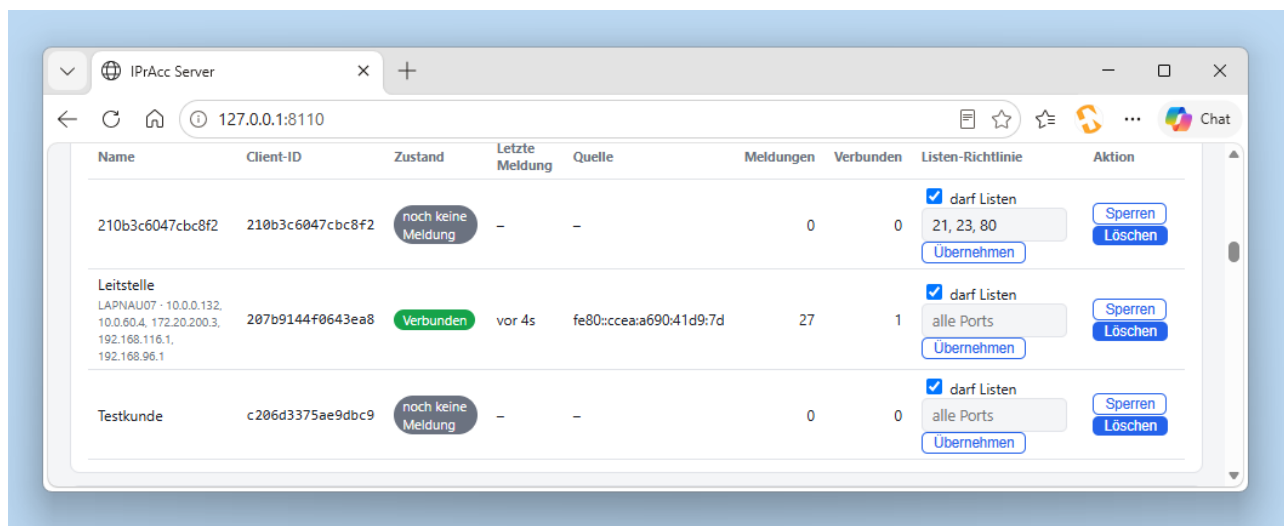
Der Bereich **Anmeldungen mit Einmalcode** zeigt zu jedem Code den Zustand: wartet auf das Gerät (Code ausgegeben, noch nicht benutzt – mit **Abbrechen** lässt er sich zurückziehen), wartet auf Ihre Bestätigung, angemeldet, abgelehnt, abgelaufen (der Code wurde nicht innerhalb der Frist benutzt bzw. nicht rechtzeitig bestätigt), abgebrochen oder fehlgeschlagen (z.B. wenn die Verbindung beim Übergeben abbricht; der Server legt dann keinen Client an). Beendete Einträge bleiben zwei Minuten sichtbar.

Hinweis: Der Code allein genügt nicht: Ohne Ihre Bestätigung erhält niemand einen Schlüssel – auch nicht, wer den Code mithört. Die Bestätigungszahl wird aus der verschlüsselten Verbindung abgeleitet; hat jemand den Code zuerst benutzt oder sich dazwischengeschaltet, stimmen die Zahlen nicht überein. Der Code besteht aus 16 Ziffern, davon eine Prü fziffer, die Tippfehler am Client sofort erkennt. Er gilt einmal und nur für `enrollvalidseconds` (Standard 600 Sekunden); wird er mehrfach falsch eingegeben (`enrollmaxfailures`, Standard 20), zieht der Server alle unbenutzten Codes zurück. Der Server erzeugt Codes nur, solange seine Weboberfläche auf 127.0.0.1 beschränkt ist; das Bestätigen ist auch über den AccWebUiGateway möglich.

3.4 Zustand der Clients überwachen

Die Tabelle im Bereich **Clients** zeigt für jeden Client:

Spalte	Bedeutung
Name / Client-ID	Anzeigenname und die zufällige, nicht aussagekräftige Kennung des Clients.
Zustand	Verbunden – die stehende Verbindung besteht. Getrennt – der Client war schon verbunden, im Moment besteht keine Verbindung (er baut sie selbstständig neu auf). überfällig – der Client ist verbunden, hat aber seit mehr als staleafterseconds (Standard 120 s) keine Statusmeldung gesendet. gesperrt – der Client wird abgewiesen. noch keine Meldung – angelegt, aber seit dem Start des Servers nie verbunden.
Letzte Meldung	Alter der letzten Statusmeldung. Führt man mit der Maus darüber, erscheint ihr Inhalt; übersichtlicher zeigt ihn der Bereich Meldungen (Abschnitt 3.7).
Quelle	IP-Adresse, von der die letzte Statusmeldung kam.
Netzinfo (unter dem Namen)	Rechnername und IP-Adressen des Client-PCs, wie der Client sie selbst meldet (ohne Loopback und Link-Local-Adressen), mit der Client-Version als Tooltip. Der Client meldet sie nach dem Verbinden und bei jeder Änderung; der Server schreibt jede Änderung ins Log und – falls eingerichtet – ins Verbindungsjournal (CLIENT_NET). So erkennen Sie, wenn ein Client umgezogen ist oder ein anderer Rechner seinen Schlüssel benutzt.
Meldungen	Anzahl der seit dem Start empfangenen Statusmeldungen.
Listen-Richtlinie	Haken darf Listen und erlaubte Ports dieses Clients (siehe unten, „Listen-Richtlinie“).
Aktion	Sperren / Entsperren und Löschen (siehe unten).



Mit **Sperren** wird ein Client sofort abgewiesen und seine bestehende Verbindung **getrennt** (ohne Neustart); sein Schlüssel bleibt gespeichert, und **Entsperren** hebt die Sperre wieder auf – der Client verbindet sich dann selbstständig neu. **Löschen** entfernt den Client samt Schlüssel, Verbindungstabelle und Verknüpfungen endgültig. Für den Client sieht ein gesperrter oder gelöschter Zugang genauso aus wie ein falscher Schlüssel (Kapitel 10).

Die Zustände „Getrennt“ und „überfällig“ sind im Server nur eine Anzeige. Eine automatische Alarmierung bei Ausfällen übernehmen weitere Anwendungen anhand der Statusdateien (Kapitel 9).

Listen-Richtlinie: wer darf Listen-Ports anbieten?

Listen-Ports öffnen die **Clients**, nicht der Server. Damit nicht jeder Client beliebige Ports in seinem Netz anbietet – etwa privilegierte Ports unter 1024 –, legen Sie in der Spalte **Listen-Richtlinie** der Clientliste für jeden Client fest:

- **darf Listen:** Ist der Haken nicht gesetzt, darf dieser Client keine Listen-Ports anbieten. Verknüpfungen mit seinen Listen-Einträgen lassen sich nicht anlegen oder sind nicht bereit, und der Client öffnet keine Listen-Ports.
- **Ports:** Die Portbereiche, die dieser Client für Listen-Ports verwenden darf, z.B. 8000–8999, 9100 (Komma oder Leerzeichen als Trenner). **Leer** = der Standard aus der Konfiguration (`defaultlistenports`), `*` = alle Ports.
- **Übernehmen** speichert die Einstellung. Sie gilt sofort: Verknüpfungen, die sie verletzen, werden „nicht bereit“ (**Client darf laut Server-Richtlinie nicht Listen bzw. Listen-Port laut Server-Richtlinie nicht erlaubt**), laufende Verbindungen enden, und der Client schließt den Port.

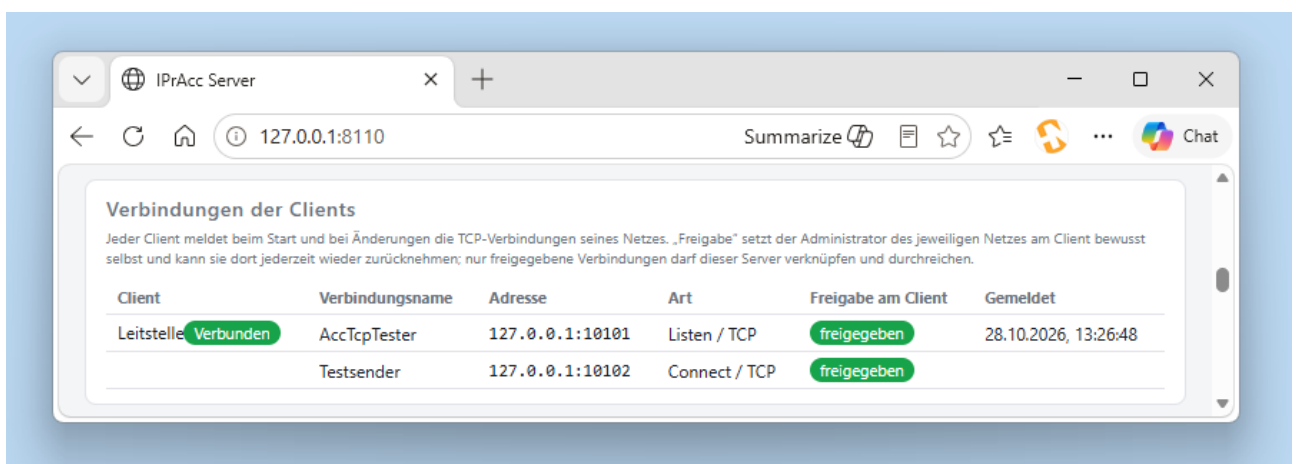
Die **Voreinstellung für alle Clients** steht in der Konfiguration (Abschnitt 3.8): `defaultmaylisten` (Standard: ja) und `defaultlistenports` (Standard: leer = alle Ports). Ohne Einstellung ändert sich also nichts. Ein Beispiel: Mit `defaultlistenports=8000–8999` dürfen alle Clients nur Listen-Ports im Bereich 8000 bis 8999 anbieten; für einen einzelnen Client genügt dann ein eigener Eintrag, um eine Ausnahme zu machen.

Der Server teilt jedem Client seine Richtlinie beim Verbinden und bei jeder Änderung mit (angezeigt unter der Verbindungstabelle des Clients, Abschnitt 4.4). Durchgesetzt wird sie aber in jedem Fall vom Server, auch wenn ein Client sie ignoriert: Der Listen-Eintrag eines Clients, der sie verletzt, wird nie durchgeleitet. Die Richtlinie steht in der Datei `policy.tbl` im Ordner der Schlüsseldatei und bleibt über Neustarts erhalten; wird ein Client gelöscht, entfällt auch seine Richtlinie.

3.5 Verbindungen der Clients

Jeder Client meldet beim Start, nach jedem Verbindungsaufbau und bei jeder Änderung seine **Verbindungstabelle** (Abschnitt 4.4) an den Server. Der Bereich **Verbindungen der Clients** fasst die Meldungen aller Clients in einer Übersicht zusammen. Der Server merkt sich den zuletzt gemeldeten Stand auch über einen Neustart hinweg.

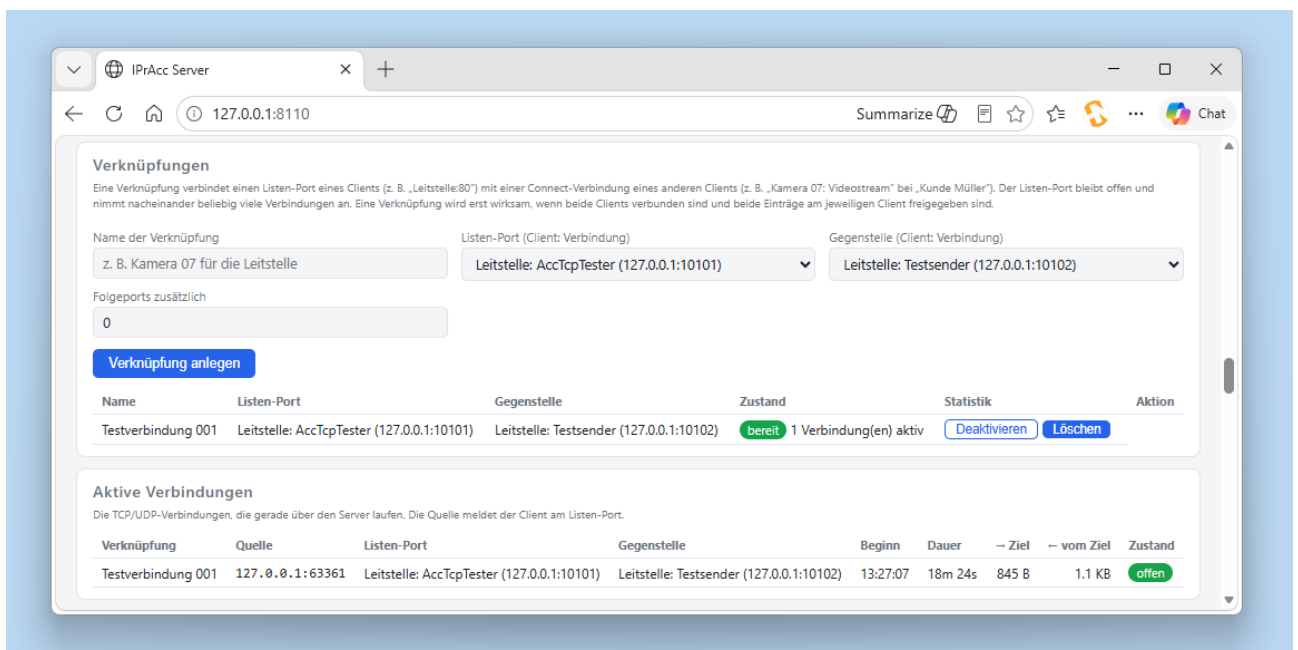
Spalte	Bedeutung
Client	Anzeigenname des Clients und sein Zustand (Verbunden/Getrennt).
Verbindungsname	Der Name, den der Administrator des Client-Netzes der Verbindung gegeben hat (z.B. „Kamera 07: Videostream“). Der Text stammt vom Client und wird nur als Text angezeigt.
Adresse	IP-Adresse und Port der Verbindung im Netz des Clients. Bei Listen-Verbindungen mit Absenderliste steht dahinter „(nur von ...)“.
Art	Listen (der Client öffnet einen Port) oder Connect (der Client verbindet sich mit der Adresse), jeweils mit dem Protokoll TCP oder UDP (z.B. „Listen / UDP“).
Freigabe am Client	freigegeben oder gesperrt : ob der Administrator des Client-Netzes die Verbindung zur Nutzung freigegeben hat. Gesperrte Verbindungen sind sichtbar, dürfen aber nicht verwendet werden. Verletzt ein Listen-Eintrag die Listen-Richtlinie des Servers (Abschnitt 3.4), steht dahinter zusätzlich „Server: darf nicht Listen“ bzw. „Server: Port nicht erlaubt“.
Gemeldet	Zeitpunkt der letzten Meldung der Tabelle dieses Clients.



3.6 Verknüpfungen anlegen und überwachen

Eine **Verknüpfung** verbindet den **Listen-Port** eines Clients (z.B. „Leitstelle:80“ im Client der Leitstelle) mit einer **Connect-Verbindung** eines Clients (z.B. „Kamera 07: Videostream“ im Client „Kunde Müller“). **Nur der Administrator am Server legt Verknüpfungen an** – ein Client kann nur solche Gegenstellen erreichen, die dort für ihn konfiguriert wurden.

1. Im Bereich **Verknüpfungen** einen **Namen der Verknüpfung** eingeben (z.B. „Kamera 07 für die Leitstelle“).
2. Unter **Listen-Port (Client: Verbindung)** den Listen-Port wählen. Angeboten werden alle gemeldeten Verbindungen der Art Listen.
3. Unter **Gegenstelle (Client: Verbindung)** die Connect-Verbindung wählen. Angeboten werden alle gemeldeten Verbindungen der Art Connect.
4. Auf **Verknüpfung anlegen** klicken.



Kameras mit aufeinanderfolgenden Ports: Viele Kameras brauchen mehrere Ports in Folge (z.B. Basisport für die Steuerung und Basisport+1 für den Videostream). Tragen Sie dann bei **Folgeports zusätzlich** die Zahl *n* ein: Der Server legt außer der gewählten Verknüpfung je eine weitere für Port+1 bis Port+n an („Name +1“, „Name +2“ ...). Die Einträge mit diesen Portnummern werden dazu auf beiden Seiten (beim Client des Listen-Ports und beim Client der Gegenstelle, mit gleichem Protokoll) gesucht. Fehlt einer, wird nichts angelegt und der Server nennt die fehlenden Ports. Jede dieser Verknüpfungen ist danach eine ganz normale Verknüpfung, die einzeln gesperrt oder gelöscht werden kann; die Freigabe an den Clients bleibt je Eintrag nötig.

Regeln und Eigenschaften:

- Ein Listen-Port gehört zu **genau einer** Gegenstelle. Dieselbe Connect-Verbindung kann dagegen über mehrere Listen-Ports erreichbar sein (z.B. von verschiedenen Arbeitsplatz-Clients aus).
- Die Protokolle der beiden Enden dürfen verschieden sein (TCP ↔ UDP, UDP ↔ UDP, TCP ↔ TCP); der Server leitet die Daten unabhängig davon durch (Abschnitt 5.3). In den Auswahllisten und in der Tabelle sind UDP-Verbindungen mit „UDP“ gekennzeichnet.
- Eine Verknüpfung lässt sich auch anlegen, solange die Verbindungen am Client noch nicht freigegeben sind. Sie wirkt aber erst, wenn **beide** Seiten freigegeben sind.
- Der Listen-Port bleibt offen: Über eine bereite Verknüpfung sind **beliebig viele Verbindungen nacheinander oder gleichzeitig** möglich, z.B. beim Aufschalten von Kameras im Browser (höchstens `maxstreamsperLink`, Standard 16).
- Mit **Deaktivieren** setzen Sie eine Verknüpfung außer Kraft, ohne sie zu löschen; **Aktivieren** schaltet sie wieder ein. **Löschen** entfernt sie. In beiden Fällen enden laufende Verbindungen sofort.
- Wird ein Client gelöscht, entfallen seine Verbindungstabelle und alle seine Verknüpfungen.

Die Spalte **Zustand** zeigt, ob und warum eine Verknüpfung wirkt. Bei einer bereiten Verknüpfung steht dahinter die Zahl der gerade durchgeleiteten Verbindungen:

Zustand	Bedeutung und Abhilfe
bereit	Die Verknüpfung ist aktiv, beide Einträge sind am jeweiligen Client freigegeben und beide Clients sind verbunden. Verbindungen werden durchgeleitet.
Client getrennt	Mindestens einer der beiden Clients hat im Moment keine Verbindung zum Server. Sobald er sich wieder verbunden hat (automatisch), wird die Verknüpfung wieder bereit. Siehe Zustand der Clients.
Listen-Port am Client nicht freigegeben	Der Administrator des Listen-Clients hat Freigabe nicht angekreuzt oder wieder entfernt. Nur er kann das ändern.
Gegenstelle am Client nicht freigegeben	Dasselbe für die Connect-Verbindung (z.B. die Kamera) im Netz des Kunden.
deaktiviert	Die Verknüpfung wurde am Server mit Deaktivieren außer Kraft gesetzt.
Client darf laut Server-Richtlinie nicht Listen	Dem Client des Listen-Ports ist das Anbieten von Listen-Ports verboten (Abschnitt 3.4). Erlauben Sie es in der Clientliste (Haken darf Listen) oder verknüpfen Sie einen anderen Client.
Listen-Port laut Server-Richtlinie nicht erlaubt	Der Port des Listen-Eintrags liegt außerhalb der erlaubten Portbereiche des Clients. Erweitern Sie die Ports in der Clientliste oder bitten Sie den Administrator des Client-Netzes, einen erlaubten Port zu verwenden.
Eintrag fehlt am Client	Der Client meldet die Verbindung nicht mehr (Zeile gelöscht). Verknüpfung löschen oder die Verbindung am Client wieder eintragen – sie bekäme eine neue UUID, die Verknüpfung müsste dann neu angelegt werden.
Art des Eintrags geändert	Die Art (Listen/Connect) der Verbindung wurde am Client geändert. Verknüpfung löschen und neu anlegen.

Hinweis: Wird die Freigabe zurückgenommen, ein Client getrennt oder die Verknüpfung deaktiviert, **enden laufende Verbindungen sofort** (der Server prüft die Bereitschaft jede Sekunde). Das ist beabsichtigt: Der Administrator eines Client-Netzes behält jederzeit die Hoheit darüber, was in sein Netz gelangt.

Aktive Verbindungen, Statistik und Verbindungsjournal

Wer greift gerade worauf zu, wie oft ist es schon passiert, und wer hat es früher getan? Das beantworten vier Hilfsmittel im Server:

Hilfsmittel	Bedeutung
Aktive Verbindungen	Die Karte Aktive Verbindungen listet die TCP/UDP-Verbindungen, die gerade über den Server laufen: Verknüpfung, Quelle (die Adresse, die sich am Listen-Port verbunden hat, vom Listen-Client gemeldet), Listen-Port und Gegenstelle jeweils mit Client- und Eintragsnamen, Beginn, Dauer, übertragene Datenmenge in beide Richtungen und Zustand (wird aufgebaut bzw. offen).
Statistik	In der Verknüpfungsliste zeigt die Spalte Statistik seit dem Start des Servers: Verbindungen / fehlgeschlagen / abgewiesen sowie die Datenmengen zum Ziel (↑) und vom Ziel (↓). In der Clientliste steht in der Spalte Verbunden , wie oft sich der Client seit dem Start neu verbunden hat; mit dem Mauszeiger erscheint der Zeitpunkt und Grund der letzten Trennung. Baut ein Client in 10 Minuten mindestens 5-mal neu auf, wird er als instabil gekennzeichnet (typisch: derselbe Client läuft zweimal, z.B. als Dienst und als Anwendung, und beide verdrängen sich gegenseitig) und einmal je Stunde im Log gemeldet.
Systemmonitor	Im Bereich Status zeigt der Server außerdem: Arbeitsspeicher des Serverprozesses (in Klammern der Höchstwert seit dem Start), Zahl der Windows-Handles, aktive und insgesamt aufgebaute Verbindungen und die seit dem Start durchgeleitete Datenmenge zum Ziel (↑) und vom Ziel (↓). Damit lässt sich die Auslastung beurteilen und ein schleichend wachsender Ressourcenverbrauch früh erkennen.
Verbindungsjournal	Der Nachweis, wer wann über welche Verknüpfung auf welches Ziel zugegriffen hat . Er wird nur geschrieben, wenn die Option <code>journalpath</code> (Konfiguration, Verbindungsjournal) einen Ordner nennt. Der Server legt dort je Tag eine Datei <code>ipracc_<JJJJMMTT>.journal</code> an (Textdatei, UTF-8, Spalten durch Tabulator getrennt, Zeit als UTC17). Die Karte Verbindungsjournal zeigt die neuesten Einträge mit Suchfeld.

Spalten des Journals: `time_utc17`, `event`, `client`, `source`, `link`, `target`, `bytes_to_target`, `bytes_from_target`, `duration_ms`, `detail`. Ereignisse:

Ereignis	Bedeutung
CONNECT	Eine Verbindung wurde durchgeschaltet (Quelle, Verknüpfung, Listen-Port und Ziel).
DISCONNECT	Die Verbindung wurde beendet – mit Datenmengen, Dauer und Grund (z.B. regulär beendet, vom Listen- bzw. Ziel-Ende abgebrochen, Client getrennt, Freigabe zurückgenommen).
FAILED	Die Verbindung zur Gegenstelle konnte nicht aufgebaut werden (z.B. Ziel nicht erreichbar); der Grund steht in <code>detail</code> .
REFUSED	Der Aufbau wurde abgewiesen, etwa weil die Verknüpfung nicht bereit ist oder eine Grenze der Lizenz erreicht wäre; der Grund steht in <code>detail</code> .
CLIENT_UP, CLIENT_DOWN	Ein Client hat sich verbunden bzw. wurde getrennt (mit dem Grund).

Dateien, die älter als `journalretentiondays` Tage sind (Standard 90), werden automatisch gelöscht. Ohne eingetragenen Ordner wird kein Journal geschrieben.

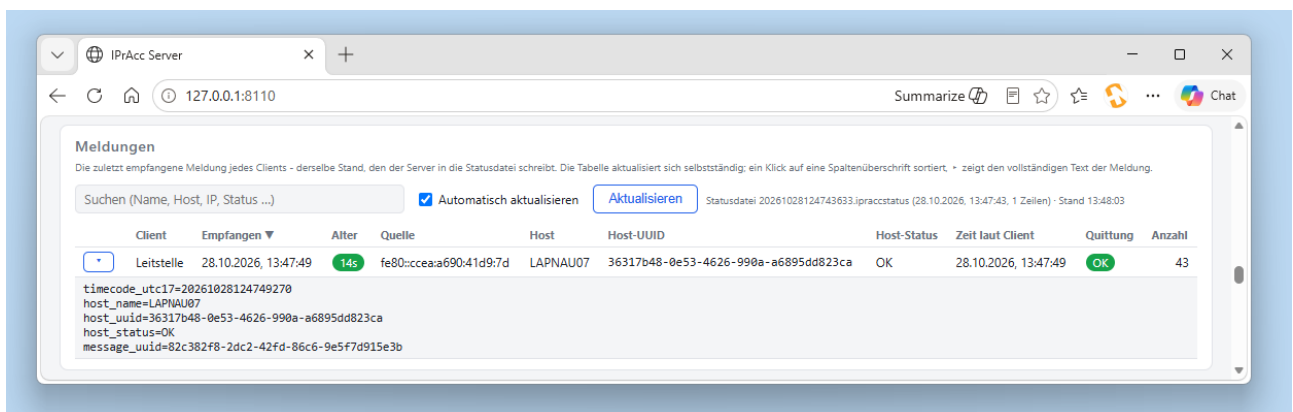
Hinweis: Das Journal enthält Adressen und Namen von Personen und Geräten und kann damit personenbezogene Daten enthalten. Legen Sie den Ordner so ab, dass nur berechnigte Administratoren Zugriff haben, und wählen Sie die Aufbewahrungszeit passend zu Ihren Datenschutzvorgaben.

3.7 Empfangene Statusmeldungen anzeigen

Der Bereich **Meldungen** zeigt die zuletzt empfangene Statusmeldung jedes Clients übersichtlich als Tabelle – denselben Stand, den der Server in die Statusdatei schreibt (Kapitel 8). Die Felder der Meldung sind dabei in eigene Spalten zerlegt:

Spalte	Bedeutung
Client	Anzeigename des Clients (bei einem gesperrten Client mit Kennzeichnung).
Empfangen / Alter	Zeitpunkt des Empfangs (Ortszeit des Browsers) und wie lange das her ist. Das Alter ist grün, solange die Meldung jünger als staleafterseconds ist, danach gelb („überfällig“).
Quelle	IP-Adresse, von der die Meldung kam.
Host, Host-UUID, Host-Status	Die Felder host_name, host_uuid und host_status aus der Meldung (Abschnitt 4.5).
Zeit laut Client	Das Feld timecode_utc17 der Meldung, in Ortszeit umgerechnet. Weicht es deutlich vom Empfangszeitpunkt ab, geht die Uhr des Clients falsch.
Quittung	Was der Server dem Client geantwortet hat: OK, oder bei einer fehlerhaften Meldung z.B. NO_MESSAGE_UUID (rot).
Anzahl	Anzahl der seit dem Start des Servers empfangenen Meldungen dieses Clients.

Die Tabelle aktualisiert sich selbstständig alle wenige Sekunden (abschaltbar mit **Automatisch aktualisieren**; **Aktualisieren** lädt sofort neu). Ein Klick auf eine Spaltenüberschrift sortiert die Tabelle; das Feld **Suchen** blendet alle Zeilen aus, in denen der Suchbegriff nicht vorkommt. Mit ► wird der vollständige, unveränderte Text der Meldung unter der Zeile eingeblendet. Sortierung, Suche und aufgeklappte Meldungen bleiben bei der automatischen Aktualisierung erhalten.



Hinweis: Der Text der Meldungen stammt von den Clients. Die Weboberfläche zeigt ihn ausschließlich als Text an; enthaltene Steuerzeichen oder HTML-Befehle werden nicht ausgeführt. Zu lange Felder werden gekürzt dargestellt (die Statusdatei enthält die vollständige Meldung).

3.8 Konfiguration des Servers

Der Bereich **Konfiguration** der Weboberfläche enthält die Einstellungen der Datei `IPrAccServer.cfg`. Änderungen werden mit **Speichern** sofort übernommen; ändern sich Port, Schutzgrenzen oder Netzlisten, startet der Server seinen Listener für sichere Verbindungen selbstständig kurz neu. **Dabei werden alle stehenden Verbindungen unterbrochen**; die Clients bauen sie automatisch innerhalb weniger Sekunden wieder auf. Nur die Schlüsseldatei und die Einstellungen der Weboberfläche selbst (Port, Adresse) wirken erst nach einem Neustart.

Sichere Verbindungen

Einstellung	Standard	Bedeutung
listenaddress	leer	Adresse, auf der gelauscht wird. Leer = alle Netzwerkadapter (IPv4 und IPv6). Damit lässt sich gezielt das Netzwerksegment wählen, auf dem die Clients erwartet werden.
listenport	5050	TCP-Port, auf dem die IPrAcc Clients ihre stehenden Verbindungen aufbauen.
publichost	Rechnername	Name oder Adresse, unter der die Clients den IPrAcc Server erreichen. Wird in das Client-Paket eingetragen.

Schutz vor Überlastung und Missbrauch (Hintergrund in Kapitel 10)

Einstellung	Standard	Bedeutung
handshaketimeoutms	3000	Gesamtzeit vom Verbindungsaufbau bis zum fertigen Handshake; danach wird ohne Antwort getrennt.
firstbytetimeoutms	1000	Ein echter Client sendet sofort. Verbindungen, die nach dieser Zeit noch nichts gesendet haben, werden getrennt (Schutz vor „Slowloris“).
idletimeoutseconds	90	Kommt von einem Client so lange (Sekunden) nichts – auch kein Ping –, gilt die stehende Verbindung als tot und wird getrennt. Muss deutlich größer sein als keepaliveseconds der Clients (Standard 30).
maxpacketspersec	100	Höchstzahl von Steuer- und Statuspaketen je Verbindung und Sekunde; wer mehr sendet, wird getrennt. Nur in der cfg-Datei. Durchgeleitete Daten sind davon ausgenommen (sie unterliegen der Flusskontrolle).
maxframebytes	65536	Maximale Größe eines Pakets. Wird vor dem Lesen geprüft.
maxconnections	256	Gleichzeitig offene Verbindungen insgesamt. Weil jeder Client eine stehende Verbindung hält, ist das zugleich die Höchstzahl gleichzeitig verbundener Clients; bei mehr Clients erhöhen. Weitere Verbindungen werden sofort geschlossen.
maxperip	8	Gleichzeitige Verbindungen je Absender-IP. Für viele Clients hinter einer gemeinsamen Adresse (NAT) erhöhen.
newconnpersec, newconnburst	5, 10	Neue Verbindungen je Absender-IP: dauerhaft pro Sekunde und kurzzeitig am Stück. (Nach einer Störung bauen alle Clients ihre Verbindung neu auf – bei vielen Clients hinter einer Adresse erhöhen.)
failbanthreshold, failbanwindowsec, failbansec	10, 60, 300	Eine Quelle, die innerhalb von 60 Sekunden 10-mal einen fehlgeschlagenen Handshake verursacht, wird 300 Sekunden gesperrt (0 = nie sperren).
allowcidr,denycidr	leer	Optionale Positiv- bzw. Negativliste von Netzen, kommasetrennt, z.B. 192.168.1.0/24, 10.0.0.5. Ist eine Positivliste angegeben, dürfen nur diese Netze verbinden.

Durchleitung

Einstellung	Standard	Bedeutung
maxstreamsperlink	16	Höchstzahl gleichzeitig durchgeleiteter Verbindungen je Verknüpfung. Nur in der cfg-Datei.

Listen-Richtlinie

Einstellung	Standard	Bedeutung
defaultmaylisten	1	Voreinstellung: Dürfen Clients ohne eigene Einstellung Listen-Ports anbieten? 0 = nur Clients, bei denen es in der Clientliste ausdrücklich erlaubt ist.
defaultlistenports	leer	Voreinstellung: Portbereiche, die Clients ohne eigene Portangabe verwenden dürfen, z.B. 8000–8999, 9100. Leer oder * = alle Ports. Je Client abweichend einstellbar (Abschnitt 3.4).

Auswertung und Schlüssel

Einstellung	Standard	Bedeutung
statuspath	status	Verzeichnis für die Statusdateien (bei einer benannten Instanz <code>status.<Instanzname></code>), relativ zum Programmordner oder absolut.
statusinterval	30	Abstand in Sekunden, in dem jeweils eine neue Statusdatei geschrieben wird.
maxstatusfiles	10	Anzahl der aufbewahrten Statusdateien; ältere werden automatisch gelöscht.
staleafterseconds	120	Ohne Statusmeldung so lange = „überfällig“ (nur Anzeige).
idfromparams	leer	Statusdatei, Spalte 1: ID aus Meldungsfeldern bilden, z.B. so, bq (Kapitel 8). Leer = Anzeigenname des Clients.
msgoutputdir, msgoutputinterval	leer, 60	Meldungsverzeichnis: Lebenszeichen für AccRoutineMonitor (Abschnitt 9.1) und Störungsmeldungen (Abschnitt 9.2). Verzeichnis und Abstand der Lebenszeichen in Sekunden. Leer = aus.
faulttimeoutseconds	120	Störungsmeldung: Ist ein Client, der verbunden war, so lange (Sekunden) getrennt, schreibt der Server eine <code>ExternalErrorMessage</code> in das Meldungsverzeichnis (Abschnitt 9.2). 0 = keine Störungsmeldungen. Wirkt nur mit Meldungsverzeichnis.
journalpath, journalretentiondays	leer, 90	Verbindungsjournal (Abschnitt 3.6): Ordner, in dem der Server je Tag eine Journaldatei schreibt (relativ zum Programmordner oder absolut), und Aufbewahrungszeit in Tagen. Leer = kein Journal.
enrollvalidseconds	600	Gültigkeit eines Einmalcodes in Sekunden (Abschnitt 3.3). Nur in der <code>cfg</code> -Datei.
enrollapprovalseconds	300	Zeit, die Sie nach der Anmeldung eines Geräts zum Bestätigen haben. Nur in der <code>cfg</code> -Datei.
enrollmaxfailures	20	Fehlversuche mit falschen Codes, nach denen alle unbenutzten Codes zurückgezogen werden. Nur in der <code>cfg</code> -Datei.
keyfile	leer	Schlüsseldatei. Leer = <code>%ProgramData%\AccMicroServices\IPrAccServer[.Instanz]\clients.keys</code> . Die Verbindungstabellen und Verknüpfungen liegen im selben Ordner. Änderung erfordert einen Neustart.
messagefile	message.inf	Vorlage der Quittung (Abschnitt 3.9).

Protokollierung (`logfile`, `loglevel`, `logretentiondays`, `logtoconsole`) und Weboberfläche (`adminport` Standard 8110, `adminbindaddress`, `adminuser`, `adminpassword`, `guiautostart`, `help`link u. a.) entsprechen den übrigen AccMicroServices.

3.9 Quittung

Jede Statusmeldung eines Clients wird vom Server mit einer Quittung beantwortet. Ihr Inhalt stammt aus der Vorlage `message.inf` (bearbeitbar im Bereich **Quittungsvorlage**). So sieht man direkt am Client, ob die Verbindung funktioniert und welchen Status der Server hat. In der Vorlage können folgende Platzhalter verwendet werden, die der Server beim Senden durch die aktuellen Werte ersetzt:

Platzhalter ...	... wird ersetzt durch
<code>%message_status%</code>	Ergebnis der Verarbeitung: OK, NO_MESSAGE_UUID (Meldung ohne <code>message_uuid</code>) oder BAD_FRAME (leere Meldung).
<code>%message_uuid%</code>	Die UUID der quittierten Meldung. Der Client prüft sie – eine Quittung mit fehlender oder falscher UUID wird verworfen.
<code>%timecode_utc17%</code>	Uhrzeit des Servers im UTC17-Format (JJJJMMTThhmmsssttt).
<code>%host_name%</code>	Name des Server-PCs.
<code>%host_uuid%</code>	Eindeutige Kennung des Server-PCs.
<code>%host_status%</code>	Status des Server-PCs, normalerweise OK.

```
message_status=%message_status%
message_uuid=%message_uuid%
timecode_utc17=%timecode_utc17%
host_name=%host_name%
host_uuid=%host_uuid%
host_status=%host_status%
```

Die Vorlage darf Escape-Sequenzen wie `\n` für einen Zeilenumbruch enthalten.

3.10 Erreichbarkeit: Port und Firewall

Damit der **IPrAcc** Server die stehenden Verbindungen der Clients annehmen kann, muss er von den **IPrAcc** Clients über eine – ggf. öffentliche – IP-Adresse erreichbar sein. Der frei wählbare **TCP-Port** `listenport` muss von dieser Adresse bis zum IPrAccServer durchgeroutet und an allen Firewalls freigegeben werden, letztlich auch an der Firewall des Betriebssystems, unter dem der Dienst läuft (Port-Forwarding im externen Router). Genau dieser Port (und der Name bzw. die Adresse aus `publichost`) wird im Client-Paket eingetragen. **Alle anderen Ports sollten für von außen eingehende Anfragen gesperrt bleiben.**

Die Clients bauen die Verbindung selbst auf; in den Client-Netzen müssen daher keine eingehenden Ports geöffnet werden. Eine Namensauflösung (DNS) ist nur nötig, wenn die Clients den Server über einen Namen ansprechen – ein Eintrag der IP-Adresse im Client-Paket (`publichost`) funktioniert ebenso und macht **IPrAcc** unabhängig vom DNS (Kapitel 12).

Befindet sich der Server **im Netz der Leitstelle**, genügt es, den Port für die Clients der Kunden freizugeben; die Arbeitsplätze der Leitstelle greifen nicht auf den Server zu, sondern auf die Listen-Ports ihres eigenen Leitstellen-Clients.

3.11 Als Windows-Dienst installieren

Für den Dauerbetrieb läuft der Server als Windows-Dienst. Dadurch wird **IPrAcc** bei jedem Systemstart automatisch mitgestartet und läuft auch, ohne dass sich ein Benutzer am System anmeldet. Am einfachsten geht das im AccMicroServiceKonfigurator (Kapitel 7): Dienst IPrAccServer wählen und **Installieren + Starten** klicken. Alternativ in einer Eingabeaufforderung mit Administratorrechten im Programmordner:

```
IPrAccServer.exe install  
IPrAccServer.exe start
```

`IPrAccServer.exe stop` beendet den Dienst, `IPrAccServer.exe uninstall` entfernt ihn wieder. Der Dienst läuft standardmäßig unter dem Konto LocalSystem; die Schlüsseldatei ist für dieses Konto lesbar.

4 Der IPrAccClient im Client-Netz

Den **IPrAcc** Client finden Sie auf dem PC im jeweiligen Netz im Verzeichnis

C:\EBUES\AccMicroServices\IPrAccClient

Wie beim Server liegen dort die Anwendung (IPrAccClient.exe), die Konfiguration (IPrAccClient.cfg) und die Vorlage der Statusmeldung (message.inf). Sobald Sie Verbindungen eintragen, kommt die Datei connections.tbl mit Ihrer Verbindungstabelle hinzu. Führen Sie IPrAccClient.exe durch Doppelklick aus: Es öffnet sich ein Konsolenfenster, und – bei guiautostart=1 – die Weboberfläche des Clients im Browser (Standard: http://127.0.0.1:8111/).

Der Client läuft im **Netz, dessen Verbindungen er anbietet oder nutzt**: bei einem Kunden auf einem PC, der die Kamera erreicht; in der Leitstelle auf einem PC, den die Arbeitsplätze erreichen. Ein PC kann auch mehrere benannte Instanzen des Clients betreiben (Kapitel 7).

IPrAcc Client Leitstelle

Sendet verschlüsselte Lebenszeichen an einen IPrAcc-Server
Web-GUI mit Status, Konfiguration und Protokollierung

Status

Server	Client-ID	Verbindung	Intervall	Läuft seit	Schlüssel
LAPNAU07:5050	207b9144f0643ea8	verbunden seit 26.10.2026, 17:30:05	30 s	1h 6m 9s	vorhanden
Letzter Versand 26.10.2026, 17:57:06	Nächster Versand 26.10.2026, 17:57:36	Quittung OK	Laufzeit 1 ms	Zeitabweichung 0 ms	Erfolgreich / fehlgeschlagen 80 / 0
Speicher 13 MB (max. 13 MB)	Handles 188	Aktive Verbindungen 2	Datenmenge 1 12 B 12 B		

Letzte Meldung
OK Meldung zugestellt und quittiert (1 ms)

Verbindungen (Freigaben für den Server)

Diese Tabelle nennt dem Server die TCP-Verbindungen dieses Netzes. „Listen“: dieser Client öffnet einen Port und nimmt nacheinander beliebig viele Verbindungen an (IP 0.0.0.0 = alle Netzwerkschnittstellen). „Connect“: dieser Client verbindet sich mit IP und Port (z. B. einer Kamera), wenn der Server-Administrator es verknüpft hat. Ein Eintrag wird erst nutzbar, wenn Sie „Freigabe“ bewusst ankreuzen; ohne Haken meldet der Client ihn nur, der Server darf ihn nicht verwenden. Die UUID bleibt gleich, auch wenn Name oder Adresse geändert werden.

Verbindungsname	IP-Adresse	Port	Protokoll	Art	Erlaubte Absender	Freigabe	Zustand	UUID
AccTcpTester	127.0.0.1	10101	TCP	Listen (Pc)	leer = alle	☒	Port offen (1 Verbindung(en))	f50c88c1-5fd8-44c9-8e79-3f332d194622
Testsender	127.0.0.1	10102	TCP	Connect	-	☒	bereit (1 Verbindung(en))	1a8a62b9-2b13-4ed4-bbe9-c5af1f94a628

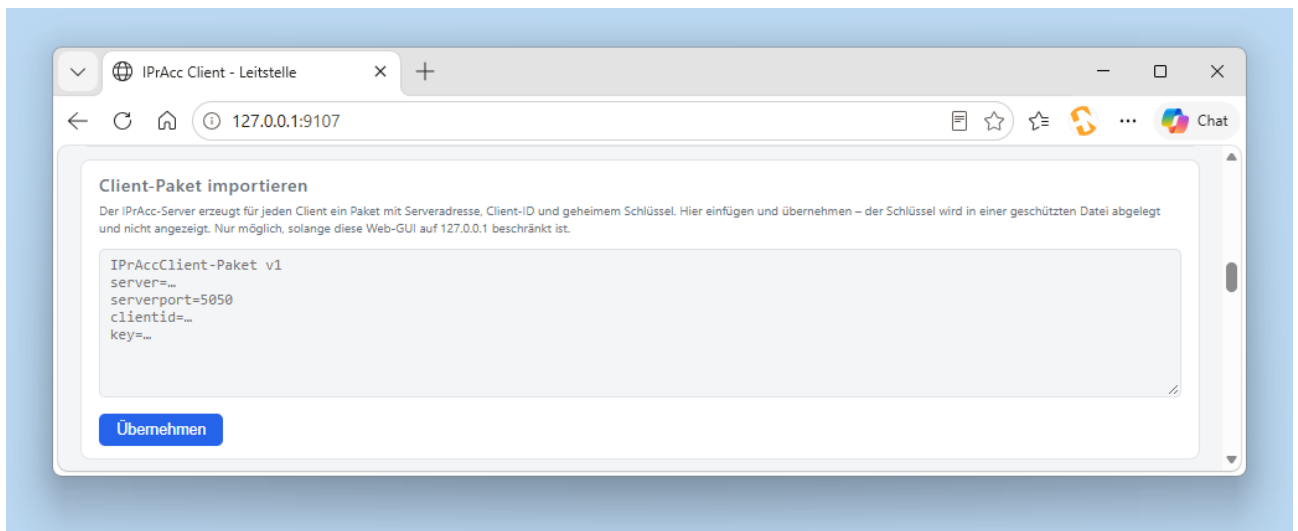
[Zelle hinzufügen](#) [Speichern und melden](#)

Letzte Meldung an den Server: 26.10.2026, 17:30:05 (2 Einträge) - Verbindungstabelle vom Server uebernommen.
Richtlinie des Servers für Listen-Ports: alle Ports sind erlaubt.

4.1 Client-Paket importieren

Server, Port, Client-ID und Schlüssel werden in einem Schritt aus dem Client-Paket übernommen, das der Server beim Anlegen des Clients erzeugt hat (Abschnitt 3.2):

1. Im Bereich **Client-Paket importieren** der Client-Weboberfläche den Paketinhalt einfügen.
2. Auf **Übernehmen** klicken.



Der Client trägt Server, Port und Client-ID in `IPrAccClient.cfg` ein, legt den geheimen Schlüssel in einer geschützten Schlüsseldatei ab und **baut die Verbindung zum Server auf**. Der Schlüssel erscheint nie in der Konfigurationsdatei und wird nie angezeigt:

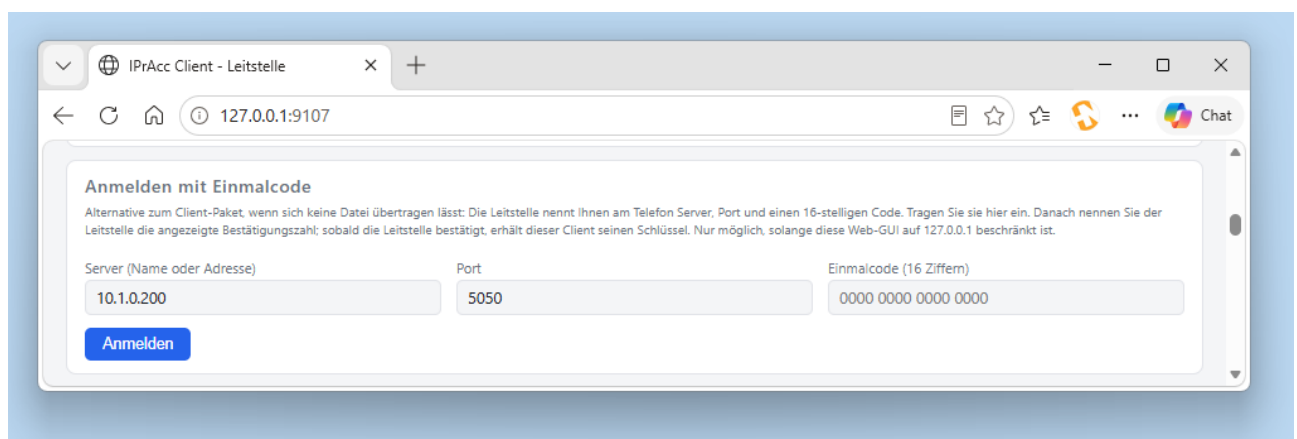
```
C:\ProgramData\AccMicroServices\IPrAccClient\client.key
```

Hinweis: Der Import ist – wie das Anlegen auf dem Server – nur möglich, solange die Weboberfläche des Clients auf 127.0.0.1 beschränkt ist, damit der Schlüssel nicht unverschlüsselt über das Netz geht. Sie können die Schlüsseldatei auch von Hand auf dem System ablegen; sie enthält den Schlüssel als Base64-Text.

4.2 Alternative: Anmelden mit Einmalcode

Wenn sich das Client-Paket nicht übertragen lässt (Kamera, Recorder, Alarmsystem ohne E-Mail und USB), meldet sich der Client mit einem Einmalcode der Leitstelle an. Die Leitstelle erzeugt den Code (Abschnitt 3.3) und nennt Ihnen am Telefon Server, Port und die 16 Ziffern:

1. In der Weboberfläche des Clients im Bereich **Anmelden mit Einmalcode** Server, Port und Einmalcode eingeben und auf **Anmelden** klicken. Leerzeichen und Bindestriche im Code spielen keine Rolle; die letzte Ziffer ist eine Prüfziffer, die Tippfehler sofort erkennt.
2. Der Client verbindet sich und zeigt eine große **Bestätigungszahl** (4 Ziffern). Nennen Sie diese Zahl der Leitstelle am Telefon und warten Sie.
3. Sobald die Leitstelle die Zahl verglichen und bestätigt hat, erscheint **Angemeldet**: Der Client hat seinen Schlüssel erhalten, in einer geschützten Datei abgelegt (er wird nicht angezeigt) und Server, Port und Client-ID in die Konfiguration übernommen. Die stehende Verbindung wird sofort aufgebaut.



Für Geräte ohne Weboberfläche gibt es die Kommandozeile (Instanzen wie üblich mit `--instance=NAME`):

```
IPrAccClient.exe enroll --server=leitstelle.example --port=5050 --code="4821 0937 5516 2084"
```

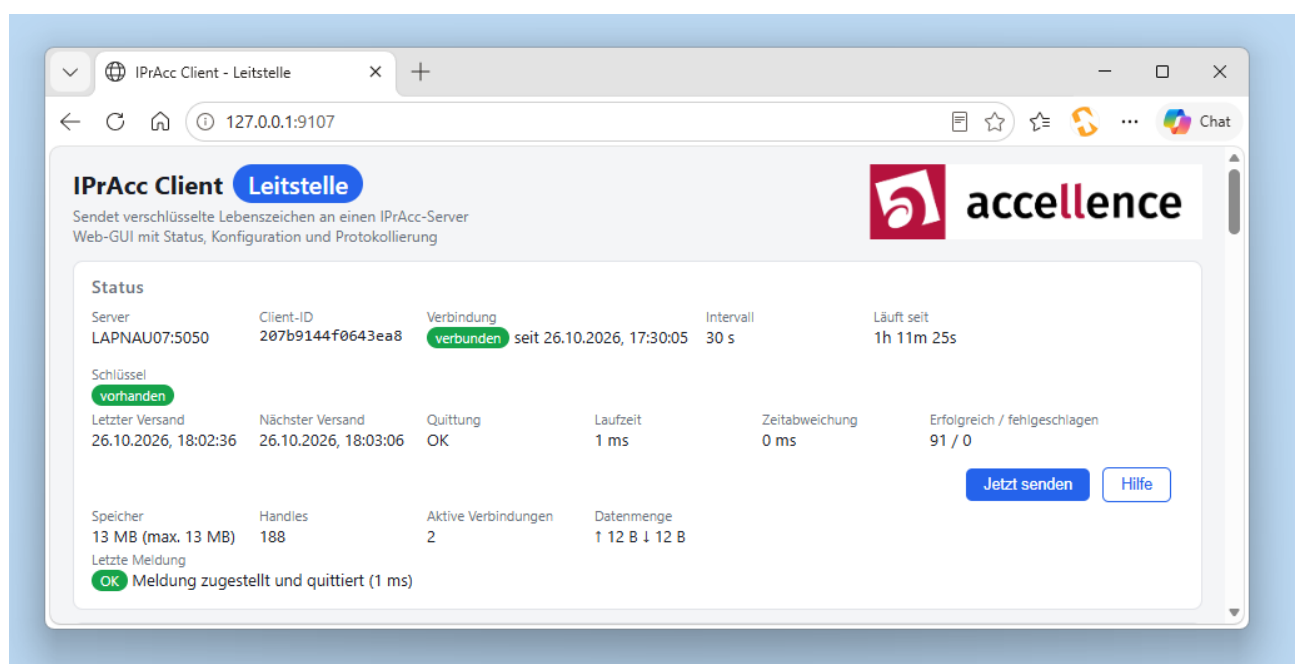
Fehlt `--code`, fragt das Programm danach. Es zeigt die Bestätigungszahl an, wartet auf die Leitstelle und beendet sich mit dem Exit-Code 0 (angemeldet), 1 (abgelehnt, abgelaufen oder fehlgeschlagen) oder 2 (Eingabe fehlerhaft). Der Dienst nimmt die neue Konfiguration und den Schlüssel selbst auf.

Hinweis: Ein falscher, abgelaufener oder bereits benutzter Code sieht für den Client genauso aus wie ein nicht erreichbarer Server („Anmeldung nicht möglich ...“) – so verrät der Server keine gültigen Codes. Prüfen Sie Code, Server und Port, und lassen Sie sich bei Bedarf einen neuen Code geben. In der Weboberfläche des Clients ist die Anmeldung wie der Paket-Import nur möglich, solange sie auf 127.0.0.1 beschränkt ist; die Kommandozeile ist davon nicht betroffen.

4.3 Status und stehende Verbindung

Der Client hält **eine stehende, verschlüsselte Verbindung** zum Server und sendet darüber regelmäßig seine Statusmeldung. Bricht die Verbindung ab – Netz weg, Server-Neustart, Sperre, hängende Verbindung –, baut er sie **selbstständig wieder auf**: zuerst nach `reconnectminseconds` (Standard 1 s), nach jedem Fehlversuch mit doppelter Pause bis höchstens `reconnectmaxseconds` (Standard 60 s), jeweils mit einer Streuung von $\pm 20\%$, damit nicht alle Clients gleichzeitig zurückkehren. Ein Ping-Austausch alle `keepaliveseconds` (Standard 30 s) erkennt tote Verbindungen auch dann, wenn gerade nichts übertragen wird; kommt `idletimeoutseconds` (Standard 90 s) lang nichts vom Server, wird die Verbindung neu aufgebaut.

Im oberen Bereich der Weboberfläche zeigt der Client den Zustand:



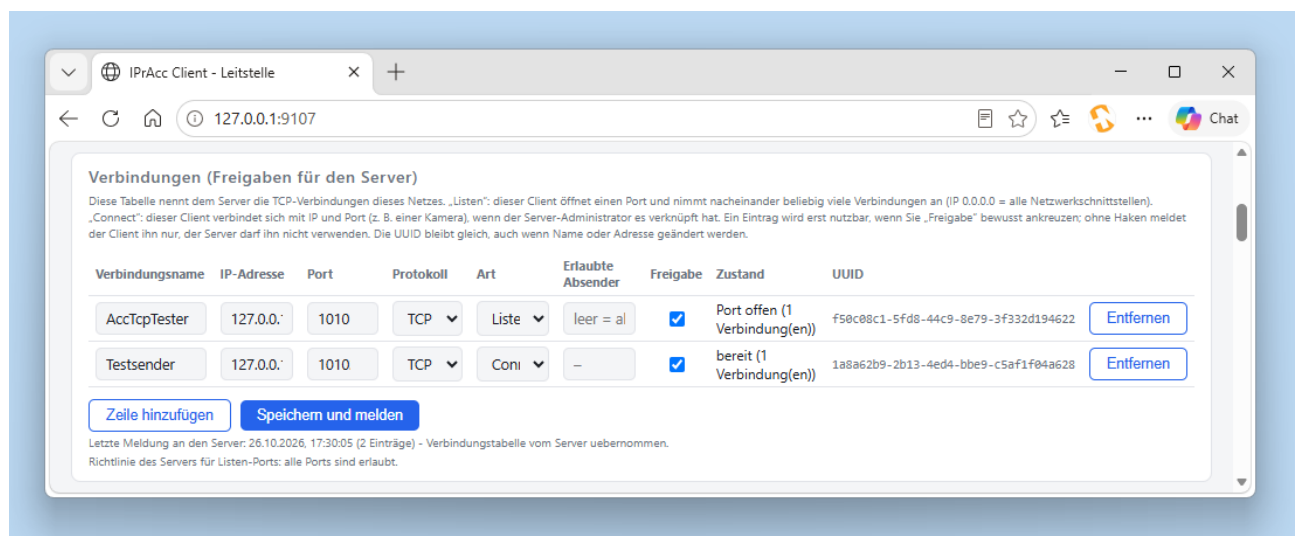
Anzeige	Bedeutung
Server, Client-ID	Ziel und Identität dieses Clients.
Verbindung	verbunden seit ..., oder getrennt mit dem Grund (z.B. „keine Daten vom Server seit 90 s“) und dem Zeitpunkt des nächsten Versuchs.
Schlüssel	vorhanden oder fehlt – Client-Paket importieren.
Letzter / Nächster Versand	Zeitpunkt der letzten Statusmeldung und der nächsten geplanten Meldung.
Quittung	Status aus der Antwort des Servers (<code>message_status</code>), normalerweise OK.
Laufzeit	Dauer von der Statusmeldung bis zur Quittung in Millisekunden – ein Maß für die Qualität der Netzwerkverbindung.
Zeitabweichung	Serverzeit minus eigene Zeit beim Empfang der Quittung (grob, enthält die halbe Laufzeit). Nur zur Information: Die Sicherheit hängt nicht von den Uhren ab. Sind alle PCs z.B. per NTP synchronisiert, zeigt der Wert die Netzwerklaufzeit.
Erfolgreich / fehlgeschlagen	Anzahl der Statusmeldungen seit dem Start des Clients.
Letzte Meldung	Ergebnis im Klartext, bei Fehlern mit Ursache (Kapitel 11).
Speicher, Handles	Systemmonitor: Arbeitsspeicher des Client-Prozesses (in Klammern der Höchstwert seit dem Start) und Zahl der Windows-Handles. Wachsen beide über Tage stetig, melden Sie das dem Support.
Aktive Verbindungen, Datenmenge	Zahl der Verbindungen, die gerade durch den Tunnel laufen, und die seit dem Start zum Server (↑) und vom Server (↓) übertragene Datenmenge.

Die Schaltfläche **Jetzt senden** löst sofort eine zusätzliche Statusmeldung aus, unabhängig vom Intervall – ideal zum Testen nach Änderungen.

Unter dem Status steht, wann die Verbindungstabelle zuletzt an den Server gemeldet und von ihm übernommen wurde.

4.4 Verbindungen und Freigaben

Im Bereich **Verbindungen (Freigaben für den Server)** legt der Administrator des Client-Netzes fest, **welche TCP/IP-Verbindungen dieses Netzes der Server nutzen darf**. Die Tabelle gehört zu diesem Client und kann für jeden Client anders aussehen.



Spalte	Bedeutung
Verbindungsname	Freier Text (1 bis 100 Zeichen), z.B. Kamera 07: videostream oder Leitstelle:80. Der Name erscheint in der Server-Übersicht und bei den Verknüpfungen.
IP-Adresse	Bei Connect : die Adresse des Geräts in diesem Netz (z.B. 10.1.0.208). Bei Listen : die Adresse der Netzwerkschnittstelle dieses PCs, auf der der Port geöffnet wird – 0.0.0.0 bedeutet alle Schnittstellen. IPv4 und IPv6 sind möglich; Namen sind nicht zulässig.
Port	Portnummer, 1 bis 65535. TCP- und UDP-Port gleicher Nummer sind verschiedene Ports.
Protokoll	TCP (Standard) oder UDP : so verwendet dieser Client die Verbindung an seinem Ende. Bei Listen + UDP öffnet er einen UDP-Port, bei Connect + UDP sendet er Datagramme an die Zieladresse. Die Gegenstelle einer Verknüpfung darf ein anderes Protokoll haben (Abschnitt 5.3).
Art	Connect (verbinden) : der Client baut auf Anforderung die Verbindung zu IP-Adresse und Port auf. Listen (Port öffnen) : der Client öffnet den Port und nimmt nacheinander beliebig viele Verbindungen an.
Erlaubte Absender	Nur bei Listen : die Adressen oder Netze, die sich mit diesem Port verbinden dürfen – IPv4 und IPv6, einzeln oder als Netz, getrennt durch Komma oder Leerzeichen, z.B. 10.0.0.0/24, 192.168.1.5. Leer = alle , die den Port erreichen. Jede andere Quelle wird sofort abgewiesen (TCP: Verbindung geschlossen, UDP: Datagramm verworfen); Server und Gegenstelle erfahren davon nichts. Bei Connect hat die Spalte keine Bedeutung.

Spalte	Bedeutung
Freigabe	Der Haken, mit dem Sie die Verbindung bewusst zur Nutzung freigeben. Neue Zeilen sind nicht freigegeben. Ohne Haken wird die Verbindung dem Server nur gemeldet, aber nie benutzt: Ein Listen-Port bleibt geschlossen, zu einer Connect-Adresse wird keine Verbindung aufgebaut.
Zustand	Port offen (Listen, freigegeben), bereit (Connect, freigegeben), Fehler: ... (z.B. der Port ist bereits belegt – der Client versucht es alle 10 Sekunden erneut) oder – (nicht freigegeben). Dahinter steht die Zahl der gerade durchlaufenden Verbindungen und in eckigen Klammern die Zahl der wegen der erlaubten Absender abgewiesenen Verbindungen.
UUID	Kennung der Verbindung, vom Client vergeben. Bleibt gleich, auch wenn Name, Adresse oder Port geändert werden; der Server ordnet Verknüpfungen über die UUID zu.

So tragen Sie Verbindungen ein:

1. Auf **Zeile hinzufügen** klicken und Name, IP-Adresse, Port, Protokoll und Art eintragen. Bei der Art **Listen** setzt der Client 0.0.0.0 ein, wenn Sie noch keine Adresse angegeben haben.
2. Mit **Freigabe** bewusst die Nutzung freigeben – oder den Haken für den Moment weglassen.
3. Auf **Speichern und melden** klicken. Der Client prüft die Eingaben, speichert die Tabelle in `connections.tbl` und meldet sie sofort an den Server; die Web-GUI zeigt anschließend den Zeitpunkt der Quittung. **Entfernen** löscht eine Zeile (die Änderung wirkt mit dem nächsten Speichern).

Die Freigabe gilt **sofort**: Nimmt der Administrator den Haken zurück und speichert, wird ein offener Listen-Port geschlossen, und laufende Verbindungen dieses Eintrags werden beendet. Damit behält er jederzeit die Hoheit darüber, was in sein Netz gelangt – ohne die Zeile löschen zu müssen. Zusätzlich prüft der Client auch dann noch selbst, wenn der Server eine Verbindung anfordert: Er baut **ausschließlich** Verbindungen zu Einträgen seiner eigenen Tabelle auf, die in diesem Augenblick als **Connect** freigegeben sind – gleichgültig, was der Server verlangt.

Richtlinie des Servers: Der Server-Administrator kann festlegen, ob dieser Client Listen-Ports anbieten darf und auf welchen Ports (Abschnitt 3.4). Der Client erfährt das beim Verbinden; unter der Tabelle steht „Richtlinie des Servers für Listen-Ports: ...“. Ein Listen-Eintrag, der sie verletzt, wird **nicht geöffnet**: Der Zustand zeigt dann „Fehler: Der Port ... liegt laut Richtlinie des Servers nicht im erlaubten Bereich“ bzw. „... erlaubt diesem Client keine Listen-Ports“. Wählen Sie dann einen erlaubten Port, oder bitten Sie den Server-Administrator um eine Anpassung. Solange der Client die Richtlinie noch nicht vom Server erhalten hat (z.B. nach einem Neustart ohne Verbindung), öffnet er seine freigegebenen Listen-Ports wie in der Tabelle eingetragen; durchgeleitet wird über den Server nur, was die Richtlinie erlaubt.

Hinweis: Geben Sie nur Verbindungen frei, deren Nutzung durch die Leitstelle vereinbart und vom Betreiber des Zielgeräts gewollt ist. Ein Listen-Port auf 0.0.0.0 ist für **alle** erreichbar, die den PC im lokalen Netz erreichen; wählen Sie Adresse und Netz so, dass nur die vorgesehenen Arbeitsplätze zugreifen können, und tragen Sie diese unter **Erlaubte Absender** ein (Kapitel 10).

Die Tabelle steht als Textdatei `connections.tbl` im Programmordner (eine Zeile je Verbindung, Spalten durch Tabulator getrennt: UUID, Name, IP-Adresse, Port, Listen oder connect, Freigabe 1/0, Protokoll tcp/udp; fehlt die letzte Spalte, gilt TCP). Wird die Datei von Hand geändert, lädt der Client sie innerhalb weniger Sekunden neu und meldet sie an den Server. Höchstens 100 Verbindungen je Client.

4.5 Inhalt der Statusmeldung

Der Inhalt der Statusmeldung steht in der Datei `message.inf` (im Bereich **Meldungsvorlage** der Weboberfläche bearbeitbar): eine Zeile je Wert in der Form `schlüssel=wert`. Fehlt `message_uuid`, ergänzt der Client es automatisch. Beispiel für ein System mit EBÜS-Bildquelle:

```
timecode_utc17=%timecode_utc17%
message_uuid=%message_uuid%
host_name=%host_name%
host_uuid=%host_uuid%
tcpip_alarm_port=13003
so=Meyer & Söhne GmbH
bq=Recorder 01
bqt=Geutebrück GCore_v8.2
serial_nr=GB0815-1000
```

Feld	Bedeutung
timecode_utc17	Sendezeitpunkt. Sind die Uhren aller Systeme korrekt synchronisiert, lässt sich damit die Laufzeit der Meldung messen.
message_uuid	Eindeutige UUID zur Nachverfolgung jeder Meldung. Die Quittung des Servers muss dieselbe UUID enthalten.
host_name, host_uuid	Name des PCs und eine eindeutige Kennung für jeden PC.
tcpip_alarm_port	Port, zu dem das überwachte System seine Alarmer per TCP/IP an die Leitstelle sendet.
so, bq, bqt	Name des in EBÜS konfigurierten Schutzobjektes, der Bildquelle und der Bildquellentyp.
serial_nr	Seriennummer zur eindeutigen Identifizierung des Systems.

Weitere Felder können in zusätzlichen Zeilen ergänzt werden; die Feldnamen müssen eindeutig sein. Die Datei wird vor jeder Statusmeldung neu eingelesen, Änderungen wirken daher bei der nächsten Meldung. Folgende Platzhalter ersetzt der Client beim Senden automatisch durch die aktuellen Werte:

Platzhalter ...	... wird ersetzt durch
%timecode_utc17%	Uhrzeit im UTC17-Format (JJJJMMThhmmsssttt, www.ebues.de/glossar#UTC17).
%message_uuid%	Eindeutige UUID zur Verwaltung jeder Meldung.
%host_name%	Hostname des PCs, von dem die Meldung verschickt wird.
%host_uuid%	Eindeutige Kennung für jeden PC.
%host_status%	Status dieses PCs, normalerweise OK.

Statt der Datei lässt sich in `IPrAccClient.cfg` auch eine einzeilige Vorlage mit Escape-Sequenzen angeben, z.B.

`message=host_name=%host_name%\nmessage_uuid=%message_uuid%`; sie wird nur verwendet, wenn die Nachrichtendatei fehlt oder leer ist.

4.6 Konfiguration des Clients

Die Einstellungen der Datei `IPrAccClient.cfg` lassen sich im Bereich **Konfiguration** der Weboberfläche bearbeiten (Einstellungen, die nur in der `cfg`-Datei stehen, sind gekennzeichnet). Änderungen wirken innerhalb weniger Sekunden – ein Neustart ist nicht nötig (Ausnahme: Port und Adresse der Weboberfläche). Ändern sich Server, Port, Client-ID oder Schlüsseldatei, baut der Client die Verbindung selbstständig neu auf.

Konfiguration

Verbindung und Meldung

Server (Name oder IP): LAPNAU07 Port: 5050 Client-ID: 207b9144f0643ea8

Intervall (Sekunden): 30 Zeitlimit Verbindungsaufbau (ms): 5000 Zeitlimit Handshake (ms): 5000

Zeitlimit Quittung (ms): 5000 Schlüsseldatei: Nachrichtendatei: message.inf

Lebenszeichen für AccRoutineMonitor

Meldungsverzeichnis (leer = aus): Abstand (Sekunden): 60

Protokollierung

Log-Datei: IPrAccClient.Leitstelle.log Log-Level: info Log-Aufbewahrung (Tage): 30

☒ Ausgabe in Konsole

Web-GUI (Neustart erforderlich für Port/Adresse)

☒ Web-GUI aktiviert

Port: 9107 Bind-Adresse: 127.0.0.1

Benutzer: Passwort: unverändert lassen

☒ Im Konsolenmodus automatisch den Browser öffnen

Fensterbreite (Pixel): 900 Fensterhöhe (Pixel): 900

Hilfe-Link (URL der Online-Dokumentation): <https://www.ebues.de/doc/IPrAcc.pdf>

Speichern

Einstellung	Standard	Bedeutung
server	–	Name oder IP-Adresse des Servers (aus dem Client-Paket).
serverport	5050	Port des Servers (aus dem Client-Paket).
clientid	–	ID dieses Clients (aus dem Client-Paket).
keyfile	leer	Schlüsseldatei. Leer = %ProgramData%\AccMicroServices\IPrAccClient[.Instanz]\client.key.
interval	30	Abstand der Statusmeldungen in Sekunden.
keepaliveseconds	30	Wurde so lange (Sekunden) nichts gesendet, schickt der Client einen Ping. Muss deutlich kleiner sein als idletimeoutseconds des Servers. Nur in der <code>cfg</code> -Datei.

Einstellung	Standard	Bedeutung
idletimeoutseconds	90	Kommt so lange nichts vom Server, gilt die Verbindung als gestört und wird neu aufgebaut. Nur in der cfg-Datei.
reconnectminseconds, reconnectmaxseconds	1, 60	Pause vor dem Wiederaufbau: beginnt bei reconnectminseconds und verdoppelt sich bei jedem Fehlversuch bis reconnectmaxseconds ($\pm 20\%$ Streuung). Nur in der cfg-Datei.
connecttimeoutms, handshaketimeoutms, replytimeoutms	5000	Zeitlimits in Millisekunden für TCP-Verbindungsaufbau, Handshake und Wartezeit auf die Quittung einer Statusmeldung.
messagefile	message.inf	Datei mit der Vorlage der Statusmeldung.
connectionsfile	connections.tbl	Datei mit der Verbindungstabelle (relativ zum Programmordner). Nur in der cfg-Datei.
udpidleseconds	60	Ein UDP-Datenstrom (je Absenderadresse an einem UDP-Listen-Port bzw. je UDP-Ziel) endet, wenn so lange (Sekunden) kein Datagramm in einer der beiden Richtungen kam. Nur in der cfg-Datei.
maxtunnelstreams	32	Höchstzahl gleichzeitig durchlaufender Verbindungen an diesem Client. Nur in der cfg-Datei.
tunnelconnecttimeoutms	5000	Zeitlimit für den Aufbau einer Verbindung zu einem Connect-Eintrag. Nur in der cfg-Datei.
msgoutputdir, msgoutputinterval	leer, 60	Lebenszeichen für AccRoutineMonitor (Abschnitt 9.1): Verzeichnis und Abstand in Sekunden. Leer = aus.

Protokollierung und Weboberfläche (adminport Standard 8111) entsprechen den übrigen AccMicroServices.

4.7 Als Windows-Dienst installieren

Wie der Server wird auch der Client im AccMicroServiceKonfigurator mit **Installieren + Starten** als Windows-Dienst eingerichtet (Kapitel 7) oder in einer Eingabeaufforderung mit Administratorrechten im Programmordner:

```
IPrAccClient.exe install
IPrAccClient.exe start
```

Der Dienst startet mit dem System und läuft ohne angemeldeten Benutzer. Läuft der Dienst nicht unter dem Standardkonto LocalSystem, muss dieses Konto die Schlüsseldatei lesen können. Nach dem Start baut der Dienst die stehende Verbindung selbst auf und hält sie – auch nach Störungen, ohne dass jemand eingreifen muss.

5 Verbindungen nutzen: Durchleitung im Detail

Dieses Kapitel beschreibt, was geschieht, wenn eine Verbindung durchgeleitet wird, und worauf Sie beim Einsatz achten sollten.

5.1 Verbindungsaufbau

Sobald eine Verknüpfung **bereit** ist (Abschnitt 3.6), steht der Listen-Port des einen Clients offen. Wer sich mit diesem Port verbindet, wird über den Server mit der Gegenstelle verbunden:



Zwischen 2 und 4 laufen die Daten über die stehenden, verschlüsselten Verbindungen der beiden Clients zum Server.
In der Leitstelle ist nur die Adresse des Listen-Ports bekannt – die IP-Adresse der Kamera im Kundennetz bleibt dort unsichtbar.

1. Ein Arbeitsplatz (hier: der Browser) verbindet sich mit der Adresse des Listen-Ports im Netz der Leitstelle, z.B. 192.168.1.5:80.
2. Der Leitstellen-Client nimmt die Verbindung an und fragt den Server, ob der Listen-Port verknüpft und bereit ist.
3. Der Server prüft die Verknüpfung und beide Freigaben und fordert beim Client der Gegenstelle die Verbindung an.
4. Der Client der Gegenstelle prüft **selbst noch einmal**, dass der Eintrag freigegeben ist, und baut die Verbindung zu seiner IP-Adresse und seinem Port auf, z.B. 10.1.0.208:80.
5. Sobald die Verbindung zur Kamera steht, schaltet der Server durch. Ab da werden die Daten in beide Richtungen übertragen.

Scheitert einer der Schritte (z.B. die Kamera ist nicht erreichbar, die Verknüpfung nicht bereit), wird die Verbindung am Listen-Port sofort wieder geschlossen; der Grund steht im Log des Clients. Ist der Leitstellen-Client gerade nicht mit dem Server verbunden, schließt er Verbindungen am Listen-Port sofort wieder.

5.2 Mehrere Verbindungen

Der Listen-Port bleibt offen. Es können **beliebig viele Verbindungen nacheinander oder gleichzeitig** über denselben Port aufgebaut werden – jede wird als eigene Verbindung mit eigener Kennung durchgeleitet. Das entspricht dem normalen Verhalten eines Servers im Netz und ist z.B. nötig, wenn Browser beim Aufschalten einer Kamera mehrere Verbindungen parallel öffnen. Die Obergrenzen sind `maxstreamperlink` am Server (je Verknüpfung, Standard 16) und `maxtunnelstreams` am Client (insgesamt, Standard 32). Wird ein Limit erreicht, werden weitere Verbindungen abgewiesen, bis wieder Platz ist. Bei UDP zählt jede Absenderadresse als eine Verbindung (Abschnitt 5.3).

5.3 UDP und Umsetzung zwischen TCP und UDP

Jede Verbindung der Tabelle verwendet an **ihrem** Ende TCP oder UDP (Spalte **Protokoll**). Durch den Tunnel selbst ändert sich nichts: Der Server leitet Datenpakete weiter, ohne zu wissen, ob sie an den Enden als TCP oder UDP erscheinen. Dadurch sind alle Kombinationen möglich:

Listen-Ende	Connect-Ende	Wirkung
TCP	TCP	Normale TCP-Durchleitung (Abschnitte 5.1 und 5.2).
UDP	UDP	Durchleitung von UDP. Jedes Datagramm kommt als ein Datagramm an; die Grenzen bleiben erhalten.
UDP	TCP	Umsetzung UDP → TCP: Die Datagramme eines Absenders werden als Bytefolge in eine TCP-Verbindung zur Gegenstelle geschrieben. Was die Gegenstelle per TCP zurücksendet, kommt beim Absender als Datagramme an.
TCP	UDP	Umsetzung TCP → UDP: Die Bytes der TCP-Verbindung werden als Datagramme an das UDP-Ziel gesendet (je Datenblock von höchstens 16 KB eines). Die Datagramme des Ziels kommen als TCP-Daten zurück.

So arbeitet ein **UDP-Ende**:

- **UDP-Listen-Port:** Jeder Absender (IP-Adresse und Port) bekommt einen eigenen **Datenstrom**. Antworten der Gegenstelle gehen an genau diesen Absender zurück – so funktionieren auch mehrere Absender gleichzeitig. Datagramme, die ankommen, während die Gegenstelle noch angesprochen wird, werden kurz zwischengespeichert (bis zu 32).
- **UDP-Ziel (Connect):** Der Client sendet die Datagramme an IP-Adresse und Port der Zeile und leitet alles zurück, was das Ziel an ihn schickt. Ein Verbindungsaufbau findet nicht statt; der Datenstrom ist sofort offen.
- **Ende eines UDP-Datenstroms:** UDP kennt kein Verbindungsende. Ein Datenstrom endet, wenn `udpidleseconds` (Standard 60 s) lang kein Datagramm kam, wenn die Freigabe zurückgenommen wird oder das TCP-Ende der Verknüpfung seine Verbindung schließt. Das nächste Datagramm desselben Absenders beginnt einen neuen Datenstrom.
- **Größe:** Ein Datagramm darf bis zu 60000 Bytes groß sein; größere werden verworfen.

Wichtige Eigenschaften und Grenzen:

- Der Tunnel zwischen Client und Server ist eine **TCP-Verbindung**. Datagramme bleiben deshalb **in der richtigen Reihenfolge** und gehen auf der Strecke nicht verloren; bei einer Störung der Internetverbindung verzögern sich aber alle Daten dieser Verbindung gemeinsam. Für Echtzeitdaten (Video, Audio) kann das zu spürbarer Verzögerung führen.
- UDP kann einen Absender nicht bremsen: Ist eine Verbindung überlastet (die Flusskontrolle hat keinen Platz mehr), werden Datagramme **verworfen** – wie bei UDP üblich.
- Bei der Umsetzung **TCP** → **UDP** gehen die Grenzen der TCP-Daten verloren: Die Datagramme entsprechen den Datenblöcken, die der Client gerade gelesen hat, nicht den ursprünglichen Nachrichten. Das passt nur zu Protokollen, die damit zurechtkommen oder eine eigene Rahmung haben. Bei **UDP** → **TCP** verschwinden die Datagrammgrenzen im Bytestrom.
- Broadcast und Multicast werden nicht durchgeleitet, nur Datagramme von und zu einzelnen Adressen.

Hinweis: Ein UDP-Port kennt keinen Verbindungsaufbau: Jeder, der den Port erreicht, kann Datagramme einspeisen, und Absenderadressen lassen sich fälschen. Wählen Sie die Adresse des UDP-Listen-Ports und ggf. die Firewall des PCs so, dass nur die vorgesehenen Absender den Port erreichen, und tragen Sie die erlaubten Absender in der gleichnamigen Spalte der Verbindungstabelle ein (Abschnitt 4.4, Kapitel 10).

5.4 Datenübertragung und Grenzen

- **Flusskontrolle:** Jede Verbindung wird einzeln und je Richtung gesteuert. Ein langsamer Empfänger bremst nur seine eigene Verbindung, andere Verbindungen und die stehenden Verbindungen der Clients bleiben davon unberührt. Es wird nirgends unbegrenzt gepuffert.
- **Beenden:** Beendet eine Seite ihre Verbindung (Halbschließen), wird das an die Gegenstelle weitergegeben, sodass auch Protokolle funktionieren, bei denen ein Ende zuerst „fertig“ ist. Bricht eine Verbindung ab, endet nur diese eine.
- **Protokolle:** TCP und UDP (Abschnitt 5.3). Nicht durchgeleitet werden Broadcast, Multicast und Protokolle ohne Ports (z.B. ICMP/Ping).
- **Verschlüsselung:** Die stehenden Verbindungen zwischen Client und Server sind verschlüsselt (Kapitel 10). Die Strecke zwischen Arbeitsplatz und Leitstellen-Client sowie zwischen Kunden-Client und Kamera liegt im jeweiligen lokalen Netz und ist so geschützt, wie es das verwendete Protokoll (z.B. HTTP oder HTTPS) und das Netz vorsehen.
- **Leistung:** Der Durchsatz hängt von der Bandbreite der Internetanbindungen der Beteiligten und der Leistung des Servers ab. Solange Verbindungen laufen, arbeitet der Server mit kurzen Takten (5 ms), damit die Daten ohne spürbare Verzögerung weitergereicht werden.

5.5 Anwendungsbeispiele

Aufgabe	Umsetzung
Kamera im Browser aufschalten	Verknüpfung Listen 0.0.0.0:80 (Leitstelle) mit Connect 10.1.0.208:80 (Kunde); im Browser <code>http://<Adresse des Leitstellen-Clients>:80/</code> aufrufen.
Viele Kameras eines Kunden	Je Kamera eine Connect-Verbindung beim Kunden und je einen eigenen Listen-Port in der Leitstelle (z.B. 8001, 8002, 8003 ...) mit je einer Verknüpfung.
Videorecorder-Software der Leitstelle	In der Leitstellen-Software als Adresse des Recorders die Adresse des Listen-Ports eintragen.
Alarm- oder Konfigurationsports	Jeder weitere TCP-Port eines Geräts erhält wie oben eine eigene Connect-Verbindung und einen eigenen Listen-Port.
UDP-Dienst im Kundennetz (z.B. Syslog, SNMP-Traps, RTP-Video)	Connect-Verbindung mit Protokoll UDP beim Kunden, Listen-Port mit Protokoll UDP in der Leitstelle, beide verknüpfen.
UDP in TCP umsetzen	Listen-Port mit Protokoll UDP in der Leitstelle, Connect-Verbindung mit Protokoll TCP beim Kunden (oder umgekehrt) – so erreicht z.B. ein UDP-Sender einen TCP-Empfänger.

Weitere Hinweise zu typischen Einsatzszenarien finden Sie im nächsten Kapitel.

6 Anwendungsszenarien

6.1 Leitstelle mit vielen Kunden

Typischer Fall: Eine Leitstelle bewacht die Liegenschaften vieler Kunden. Sie betreibt **einen IPrAcc Server** und einen **IPrAcc Client** (oder mehrere) in ihrem Netz. Jeder Kunde erhält einen Client mit eigenem Schlüssel; der Kunden-Administrator trägt dort die Geräte ein, die die Leitstelle erreichen soll, und gibt sie frei. Die Leitstelle legt am Server je Gerät eine Verknüpfung an. Sie entscheidet damit, **was** sie aufschaltet; der Kunde entscheidet, **was** er freigibt – beides muss zusammenpassen.

Bewährt hat sich ein festes Schema für die Listen-Ports der Leitstelle, z.B. je Kunde ein Portbereich (Kunde 1: 8101–8199, Kunde 2: 8201–8299 ...) und aussagekräftige Verbindungsnamen („Müller, Kamera 07: Videostream“).

6.2 Gleiche Adressbereiche bei mehreren Kunden

Bei einem VPN führt es zu Adresskonflikten, wenn mehrere Kunden dasselbe Netz verwenden (z.B. 192.168.1.0/24 oder 10.1.0.0/24) – das kommt in der Praxis ständig vor. Bei **IPrAcc** arbeitet jeder Client in seinem eigenen Adressbereich: In der Leitstelle sind nur die Adressen der Listen-Ports sichtbar, die die Leitstelle selbst festlegt. Zwei Kameras mit der gleichen IP-Adresse in zwei Kundennetzen sind deshalb über zwei verschiedene Listen-Ports problemlos erreichbar.

6.3 Gezielte Kopplung zweier Netze

Server und Clients können auch auf demselben PC oder in einem gemeinsamen Netz betrieben werden, um gezielt einzelne Verbindungen zwischen mehreren Netzen zu ermöglichen, die von diesem PC aus erreichbar sind (selektive Netzwerkkopplung): Jede Verbindung muss einzeln eingetragen, freigegeben und verknüpft werden; es entsteht keine allgemeine Netzverbindung.

6.4 Zwei Netze, die sich nicht erreichen können

Zwei Netze A und B liegen hinter Firewalls oder Routern, die keine eingehenden Verbindungen zulassen, und Rechner in A sollen Dienste in B nutzen – vielleicht auch umgekehrt. Ein VPN scheidet aus (Adresskonflikte, Aufwand, Sicherheitsbedenken), und im Internet soll kein Port geöffnet werden. Mit **IPrAcc** genügt **ein Server**, der von beiden Netzen aus erreichbar ist (z.B. in einer DMZ oder beim Dienstleister), und **je ein Client in jedem Netz**; beide Clients bauen die Verbindung nach außen auf.

- **A nach B:** Der Client in B trägt den Dienst als Connect-Eintrag ein und gibt ihn frei; der Client in A trägt einen Listen-Eintrag ein (der Port, an dem die Anwender in A den Dienst erreichen) und gibt ihn frei. Der Server-Administrator verknüpft beide.
- **B nach A:** Dasselbe in Gegenrichtung: Connect-Eintrag im Client in A, Listen-Eintrag im Client in B, eine zweite Verknüpfung.
- Jeder Client kann beide Arten von Einträgen enthalten, und jede Verbindung wird einzeln freigegeben. Es entsteht keine allgemeine Netzverbindung: Nur die eingetragenen und freigegebenen Dienste sind erreichbar. Verkettete Server, wie sie bei der Vorgängerversion IPrAccTcp nötig waren, entfallen.

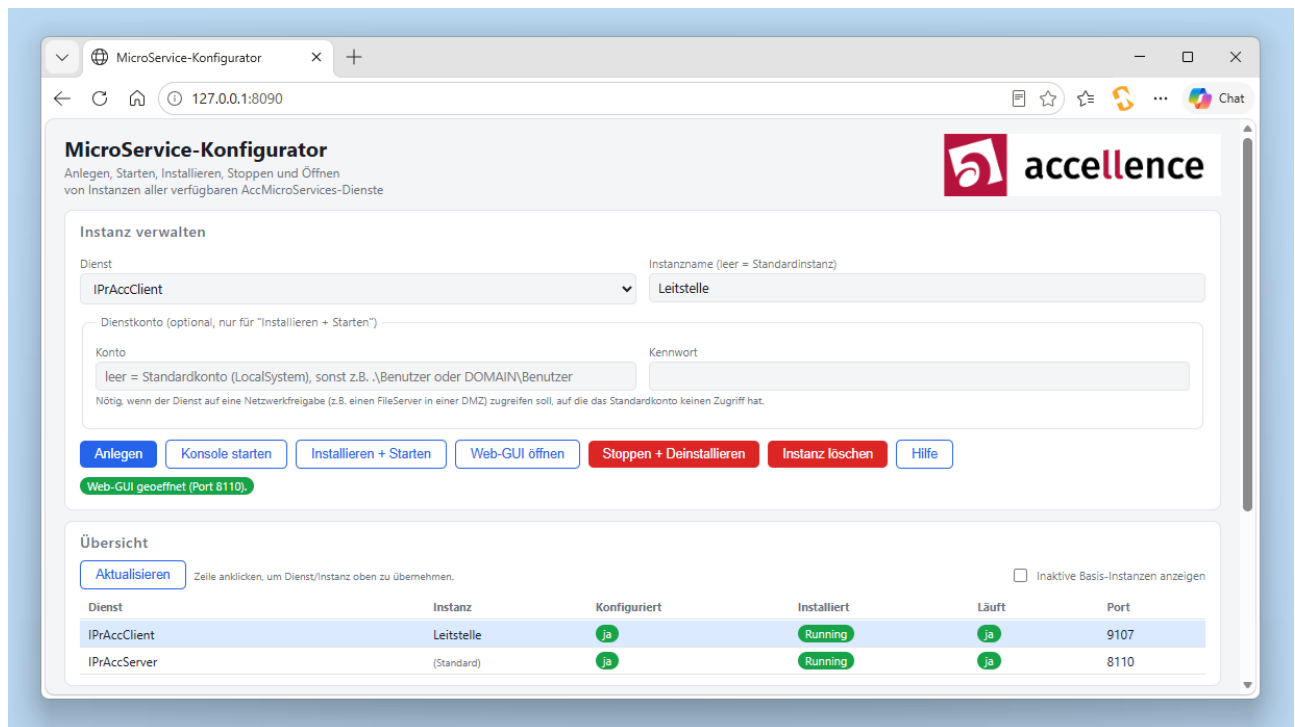
6.5 Mehrere Server und Instanzen

Ein **IPrAcc** Server kann bei Bedarf mehrfach als benannte Instanz gestartet werden – z.B. je Mandant oder Netz ein eigener Server mit eigenen Clients, eigener Schlüsseldatei und eigenem Statusverzeichnis. Entsprechend kann auf einem PC je Zielsever eine benannte Instanz des **IPrAcc** Clients laufen (AccMicroServiceKonfigurator, Kapitel 7).

Für erhöhte Ausfallsicherheit können mehrere Clients ihre Statusmeldungen über unterschiedliche Netze an verschiedene **IPrAcc** Server senden. Dank des Parameters `messagefile` genügt es, die Meldungsdaten nur einmal zentral bereitzustellen: Alle Instanzen können auf dieselbe Datei zeigen.

7 Verwaltung mit dem MicroServiceKonfigurator

IPrAccServer und IPrAccClient erscheinen im AccMicroServiceKonfigurator in der Liste **Dienst** und in der Übersicht. Dort lassen sie sich wie alle AccMicroServices anlegen, in der Konsole starten, als Windows-Dienst installieren und stoppen oder deinstallieren, ihre Weboberfläche und diese Dokumentation öffnen (Details siehe `AccMicroServices.docx`).



Besonderheiten bei **IPrAcc**:

- **Zwei Ports je Server-Instanz:** Neben dem Port der Weboberfläche (adminport) öffnet der Server den Port für die Clients (listenport). Legt der Konfigurator eine benannte Instanz an, vergibt er für beide automatisch freie Ports aus seinem Portbereich. Passen Sie den Port für die Clients bei Bedarf in der Weboberfläche an (z.B. auf 5050) und geben Sie ihn in der Firewall frei; das Client-Paket enthält jeweils den aktuellen Wert.
- **Mehrere Instanzen:** Mit benannten Instanzen betreiben Sie z.B. je Mandant oder Netzwerk einen eigenen Server (eigene Clients, eigene Schlüsseldatei, eigenes Statusverzeichnis) beziehungsweise auf einem PC einen Client je Zielserver.
- **Instanz löschen:** Beim Löschen einer benannten Instanz entfernt der Konfigurator auch deren Schlüsselverzeichnis (`%ProgramData%\AccMicroServices\IPrAccServer.<Instanz>`). Alle zugehörigen Clients verlieren damit ihren Zugang, ebenso gehen die gemeldeten Verbindungstabellen und die Verknüpfungen verloren. Die unbenannte Basis-Instanz kann nicht gelöscht werden.

8 Format der Statusdateien

Der **IPrAcc** Server empfängt regelmäßig Statusmeldungen von vielen **IPrAcc** Clients. Um stets den aktuellen Zustand aller Systeme bereitzustellen und zugleich den Speicherplatz zu begrenzen, legt er die Daten in Statusdateien rollierend ab (mit automatischer „Rotation“). So stehen immer die letzten Statusstände zur Verfügung, ältere werden automatisch gelöscht.

Wie funktioniert die rollierende Speicherung?

- In regelmäßigen Abständen (`statusinterval`, z.B. alle 30 Sekunden) schreibt der Server eine neue Statusdatei in das konfigurierte Verzeichnis (`statuspath`).
- Sobald die Anzahl der Statusdateien `maxstatusfiles` überschreitet, werden die ältesten automatisch gelöscht.
- Zur eindeutigen Erkennung enthält der Dateiname den Zeitpunkt, zu dem die Datei gespeichert wurde, im UTC17-Format, und die Endung `.ipraccstatus`, z.B. `20261025214648030.ipraccstatus`.
- Jede Statusdatei enthält, zeilenweise gegliedert nach ID, die zuletzt mit dieser ID empfangene Statusmeldung, jeweils mit IP-Adresse und Zeitstempel.

Vorteile: Die Auswertung kann immer auf die letzten, konsistenten Statusstände zugreifen – am sichersten auf die zweitneueste Datei, die garantiert vollständig geschrieben ist. Die Dateien bleiben lange genug erhalten, um sie sicher auswerten zu können, und der Speicherplatz bleibt begrenzt.

Beispiel für den Inhalt einer Statusdatei (hier mit `idfromparams=so,bq`; TAB steht für ein Tabulatorzeichen):

```
Meyer & Söhne GmbH,Recorder 01 TAB 192.168.1.101 TAB 20261025174634001 TAB  
timecode_utc17=20261025174634001\nmessage_uuid=...\ntcpip_alarm_port=13003\nso=Meyer & Söhne  
GmbH\nbq=Recorder 01\nbqt=Geutebrück GCore_v8.2\nserial_nr=GB0815-1000  
Alpha AG,Recorder 02 TAB 192.168.1.102 TAB 20261025174634123 TAB  
timecode_utc17=20261025174634123\nmessage_uuid=...\ntcpip_alarm_port=13004\nso=Alpha  
AG\nbq=Recorder 02\nbqt=Geutebrück GCore_v8.2\nserial_nr=GB0815-1001
```

Jede Zeile enthält, durch Tabulator getrennt:

1. die ID des Systems (siehe unten),
2. die IP-Adresse, von der die Statusmeldung dieses Systems empfangen wurde,
3. den Empfangszeitpunkt der letzten Meldung im UTC17-Format,
4. die Meldung selbst (mehrzeilig; `\n` steht dabei für einen Zeilenumbruch, `\t` für ein Tabulatorzeichen, `\\` für einen Backslash).

Die Datei kann einfach von anderen Programmen ausgewertet werden. In jeder neuen Statusdatei stehen die jeweils letzten Meldungen aller Systeme, von denen seit dem Start des Servers Meldungen empfangen wurden.

Welche ID steht in Spalte 1? Standardmäßig der Anzeigename, den Sie beim Anlegen des Clients vergeben haben – er steht fest im Server und kann von einem Client nicht verändert werden. Möchten Sie die ID aus Feldern der Meldung bilden (z.B. `so` und `bq` für EBÜS-Bildquellen), tragen Sie die Feldnamen kommasetrennt bei `idfromparams` ein, z.B. `idfromparams=so,bq`. Fehlt ein Feld in einer Meldung, wird ersatzweise der Anzeigename verwendet.

Hinweis: Felder der Meldung stammen vom Client selbst. Ein kompromittierter Client könnte darin eine fremde ID angeben und so einen anderen Eintrag überschreiben. Wo das eine Rolle spielt, lassen Sie `idfromparams` leer und vergeben die Anzeigenamen so, dass Ihre Auswertung sie verwenden kann.

9 Erkennen und Melden von Systemausfällen

Der **IPrAcc** Server dokumentiert für jedes System anhand der ID die aktuelle IP-Adresse, von der die Statusmeldung empfangen wurde, den exakten Zeitpunkt der letzten Meldung sowie alle weiteren Felder, die in der Datei `message.inf` des jeweiligen **IPrAcc** Clients konfiguriert sind.

Über die Zeitstempel lassen sich Ausfälle einzelner Systeme oder ganzer Netzsegmente zuverlässig erkennen und dokumentieren: Bleibt die Meldung eines Systems länger als erwartet aus, ist es oder die Netzwerkverbindung zu ihm gestört. In diesem Fall kann die Leitstelle entsprechende Alarmer generieren und alle nötigen Maßnahmen einleiten. Die Weboberfläche des Servers zeigt solche Clients bereits als **Getrennt** oder **überfällig** an. Die stehende Verbindung ist dabei selbst ein Prüfstein: Solange der Client verbunden ist und seine Statusmeldungen quittiert werden, funktioniert auch der Netzwerkweg zwischen Kundennetz und Leitstelle, den durchgeleitete Verbindungen und Alarmer nehmen.

9.1 Lebenszeichen für AccRoutineMonitor

Damit auch der **IPrAcc**-Dienst selbst überwacht werden kann, schreiben **IPrAcc** Server und **IPrAcc** Client auf Wunsch ein regelmäßiges Lebenszeichen in einen Ordner, den das Überwachungswerkzeug AccRoutineMonitor beobachtet. Dazu tragen Sie in der Weboberfläche (Bereich **Konfiguration**, Gruppe **Lebenszeichen für AccRoutineMonitor**) oder in der `cfg`-Datei ein Meldungsverzeichnis (`msgoutputdir`) und den Abstand in Sekunden (`msgoutputinterval`, Standard 60) ein. Ist das Verzeichnis leer, wird nichts geschrieben.

In diesem Abstand entsteht eine kleine Datei mit dem Namen `<Kennung>_<UTC17-Zeitstempel>.msg` (z.B. `IPrAccServer_20261025123015123.msg`; bei einer benannten Instanz mit deren Namen). Der Zeitstempel im Dateinamen ist wichtig: AccRoutineMonitor erkennt eine Datei nur dann als neu, wenn sich der Name ändert. Der Inhalt hat ein von AccRoutineMonitor fest vorgegebenes Format:

```
RoutineMessage
InstanceName=<Kennung>
TimeStamp=<UTC17-Zeitstempel>
IntervalTime=<msgoutputinterval>
--- eof ---
```

- Der **IPrAcc** Server schreibt das Lebenszeichen nur, solange sein sicherer Listener läuft, also solange er Verbindungen der Clients annehmen kann. Kann der Port nicht geöffnet werden (z.B. weil er belegt ist), bleibt die Datei aus, und AccRoutineMonitor schlägt Alarm; sobald der Server den Port wieder öffnen konnte (Wiederholung alle 10 Sekunden), erscheinen die Dateien wieder.

- Der **IPrAcc** Client schreibt das Lebenszeichen, solange er läuft – unabhängig davon, ob der Server gerade erreichbar ist. Ob der Client verbunden ist, zeigt der Server selbst (Zustand „Getrennt“, Kapitel 3).

Hinweis: Die Dateien sammeln sich im Meldungsverzeichnis an. Wie bei AccFileCleaner und AccFileMover räumt eine AccFileCleaner-Instanz, die auf dieses Verzeichnis zeigt, sie wieder auf.

9.2 Störungsmeldung bei Ausfall eines Clients

Der Server überwacht seine Clients. Damit ein Ausfall in der Leitstelle auffällt – z.B. in EBÜS –, schreibt er ihn in dasselbe Meldungsverzeichnis wie das Lebenszeichen (msgoutputdir), und zwar im selben Format und nach demselben Mechanismus wie AccFileMover. Ist das Verzeichnis das Alarmverzeichnis einer Bildquelle (z.B. C:\EBUES\AccAlarmServer\FtpRoot_Leitstelle\IPrAcc_Server), zeigt EBÜS die Meldungen an; für das Lebenszeichen gibt es dort außerdem dieROUTINERUF-Überwachung.

- **Ausfall:** Bleibt ein Client, der verbunden war, länger als `faulttimeoutseconds` (Standard 120 Sekunden) getrennt, legt der Server **einmalig** eine Datei mit der ersten Zeile `ExternalErrorMessage` ab. Die Meldung nennt den Clientnamen, seine zuletzt bekannte Adresse und den Server, z.B. „Client 'Kunde Müller' (zuletzt 203.0.113.7) ist seit mehr als 2 Minuten vom IPrAccServer LEITSTELLE getrennt“.
- **Rückkehr:** Verbindet sich der Client wieder, folgt eine Datei mit der ersten Zeile `StatusMessage` („... ist wieder verbunden (Ausfall: ...)“).
- **Karenzzeit:** Kurze Unterbrechungen – ein Neustart des Clients, ein Neustart des Server-Listeners nach einer Konfigurationsänderung – bleiben unter der Karenzzeit und werden nicht gemeldet. Wählen Sie sie so, dass ein einfacher Neustart noch keine Meldung auslöst. Eine Verbindung, die ohne Abbruch „hängt“, wird nach `idletimeoutseconds` (Standard 90 Sekunden) als getrennt erkannt, die Karenzzeit beginnt dann erst.
- **Keine Störung:** Gesperrte oder gelöschte Clients gelten nicht als Störung. Gemeldet werden nur Clients, die seit dem Start des Servers verbunden waren.
- **Ein/Aus:** Die Funktion wirkt nur, wenn ein Meldungsverzeichnis eingetragen ist. Mit `faulttimeoutseconds=0` werden keine Störungsmeldungen geschrieben (das Lebenszeichen bleibt davon unberührt).

Inhalt einer Störungsmeldung (Beispiel, Dateiname <Kennung>_<UTC17-Zeitstempel>.msg):

```
ExternalErrorMessage
IPrAccServer
Client 'Kunde Müller' (zuletzt 203.0.113.7) ist seit mehr als 2 Minuten vom IPrAccServer
LEITSTELLE getrennt
--- eof ---
```

Die erste Zeile kennzeichnet die Art der Meldung (`StatusMessage`, `ExternalErrorMessage`, ...) wie bei den übrigen AccMicroServices; die zweite Zeile ist die Kennung der Instanz (bei einer benannten Instanz mit deren Namen).

10 Datensicherheit

Die Verbindung zwischen **IPrAcc** Client und **IPrAcc** Server nutzt die gemeinsame Sicherheitsschicht der AccMicroServices. Sie ist bewusst auf bewährte Standards aufgebaut, statt eigene Verfahren zu erfinden:

Schutzziel	Umsetzung
Vertraulichkeit und Integrität	Alle übertragenen Daten – Statusmeldungen wie durchgeleitete Verbindungen – werden mit TLS 1.3 und AES-256-GCM verschlüsselt und gegen Veränderung gesichert. Ältere TLS-Versionen werden abgelehnt.
Authentifizierung	Jeder Client hat eine eigene ID und einen eigenen zufälligen 256-Bit-Schlüssel (Pre-Shared Key). Beide Seiten beweisen im Handshake den Besitz dieses Schlüssels – ein falscher Server wird vom Client ebenso erkannt wie ein falscher Client vom Server, ganz ohne Zertifikatsverwaltung.
Schutz vor Wiedereinspielen (Replay)	Jede Verbindung beginnt mit einem frischen Handshake mit neuen Zufallswerten; ein mitgeschnittener Datenverkehr lässt sich nicht wiederholen. Die Sicherheit hängt nicht von den Uhren der Systeme ab – Zeitstempel in den Meldungen dienen nur der Information. Die Quittung des Servers ist über die message_uuid an die gesendete Meldung gebunden.
Vorwärtsgeheimnis (Forward Secrecy)	Der Sitzungsschlüssel entsteht durch einen Schlüsselaustausch (ECDHE) pro Verbindung. Wird ein Schlüssel später bekannt, lässt sich der damals mitgeschnittene Verkehr trotzdem nicht entschlüsseln.
Einzelne Clients sperren	Ein Client lässt sich im Server sofort sperren oder löschen; seine bestehende Verbindung wird dabei getrennt. Ein gesperrter oder gelöschter Zugang, ein unbekannter Client und ein falscher Schlüssel sind von außen nicht zu unterscheiden – ein Angreifer kann gültige Client-IDs nicht durch Probieren herausfinden.
Anmeldung mit Einmalcode	Einmaliger, 10 Minuten gültiger 16-stelliger Code mit Prüfziffer; der Schlüssel entsteht erst nach der Bestätigung durch den Administrator (Bestätigungszahl aus der verschlüsselten Verbindung); der Code wird nie gespeichert oder protokolliert und nach vielen Fehlversuchen zurückgezogen.
Freigabe durch den Administrator des Client-Netzes	Ein Client benutzt ausschließlich Verbindungen, die in seiner eigenen Tabelle stehen und von seinem Administrator ausdrücklich freigegeben wurden. Neue Einträge sind nicht freigegeben. Die Freigabe lässt sich jederzeit zurücknehmen und wirkt sofort.
Verknüpfung durch den Server-Administrator	Der Server leitet nur Verbindungen durch, für die der Server-Administrator eine Verknüpfung angelegt hat. Ein Client kann nur die Gegenstellen erreichen, die ihm dort zugeordnet wurden – nie ein beliebiges Ziel.
Erlaubte Absender an Listen-Ports	Der Administrator des Client-Netzes legt je Listen-Eintrag fest, welche Adressen oder Netze sich verbinden dürfen (IPv4/IPv6, einzeln oder als Netz). Jede andere Quelle wird sofort abgewiesen, noch bevor der Server oder die Gegenstelle etwas davon erfahren; bei UDP wird das Datagramm verworfen. Eine nicht lesbare Liste lässt niemanden durch. Änderungen wirken sofort und beenden laufende Verbindungen nicht mehr erlaubter Quellen.
Listen-Richtlinie des Servers	Der Server-Administrator bestimmt, welche Clients Listen-Ports anbieten dürfen und auf welchen Ports (z.B. keine privilegierten Ports). Der Server setzt das bei jeder Verknüpfung und jeder Verbindung durch und teilt es den Clients mit, die nicht erlaubte Ports gar nicht erst öffnen.
Prüfung an mehreren Stellen	Bereitschaft und Freigaben werden vom Server laufend (jede Sekunde), vom Listen-Client und – als letzte Instanz – vom Client der Gegenstelle geprüft. Fällt eine Voraussetzung weg, enden laufende Verbindungen sofort.

Schutz vor Überlastung und Angriffen: Unzulässige Verbindungen werden mit geringstmöglichem Ressourcenverbrauch ohne Antwort beendet, bevor Verschlüsselungsaufwand entsteht:

- Verbindungen, die den Handshake nicht rechtzeitig abschließen (z.B. langsame „Slowloris“-Angriffe) oder keine gültigen Daten senden, werden nach wenigen Sekunden getrennt.
- Die Zahl gleichzeitiger Verbindungen ist insgesamt und je Absender-IP begrenzt, ebenso die Rate neuer Verbindungen je IP.
- Quellen, die wiederholt Handshakes verfehlen, werden vorübergehend gesperrt; optional lassen sich Netze per Positiv- oder Negativliste freigeben bzw. ausschließen.
- Die Größe eines Pakets ist begrenzt und wird geprüft, bevor es gelesen wird; ein angemeldeter Client darf nur eine begrenzte Zahl von Steuerpaketen je Sekunde senden, und durchgeleitete Verbindungen unterliegen einer Flusskontrolle sowie Obergrenzen je Verknüpfung und Client.
- Fehlgeschlagene Verbindungsversuche werden im Protokoll nur zusammengefasst und mengenbegrenzt gemeldet; Angaben aus unbeglaubigten Verbindungen werden bereinigt, damit sich nichts in das Protokoll einschleusen lässt.

Schutz der Weboberflächen: Die Weboberfläche läuft im Browser des Administrators und muss auch dort geschützt werden – eine fremde Webseite, die gerade im selben Browser geöffnet ist, darf sie nicht fernsteuern können. **IPrAcc** Server und **IPrAcc** Client prüfen deshalb den Host-Namen jeder Anfrage (nur 127.0.0.1, localhost und [::1]; Schutz vor „DNS-Rebinding“), lehnen Anfragen fremder Seiten ab und verlangen für jede Änderung ein Sicherheits-Token, das nur die eigene, vom Dienst ausgelieferte Seite kennt. Die Antworten enthalten Sicherheits-Header (kein Einbetten in fremde Seiten, kein Zwischenspeichern). Dateipfade (Log, Nachrichtendatei, Statusverzeichnis) lassen sich über die Weboberfläche nur auf relative Pfade unterhalb des Programmordners ändern – die Dienste laufen mit weitreichenden Rechten, ein beliebiger Pfad würde sonst das Überschreiben beliebiger Dateien erlauben. Die Schlüsseldatei und absolute Pfade lassen sich nur direkt in der cfg-Datei einstellen. Der Zugriff auf die Weboberfläche über den AccWebUiGateway ist möglich, das Erzeugen und Importieren von Schlüsseln jedoch nicht.

Hinweis: Die Weboberfläche ist standardmäßig ohne Anmeldung nur vom selben Rechner aus erreichbar. Auf einem Rechner, an dem sich mehrere Benutzer anmelden (z.B. Terminalserver), kann deshalb jeder dieser Benutzer die Oberfläche bedienen. Setzen Sie dort adminuser und adminpassword.

Hinweis: Eine verteilte Überlastung des Netzwerks (DDoS) kann keine Anwendung allein abwehren – dafür sind Firewall und Netzbetreiber zuständig. **IPrAcc** sorgt dafür, dass der Server selbst bei vielen unzulässigen Verbindungen nicht zusätzlich belastet wird und die zulässigen Clients weiter bedient werden.

Schutz der Schlüssel: Die Schlüssel stehen nicht in Konfigurationsdateien, sondern in eigenen Schlüsseldateien im Ordner %ProgramData%\AccMicroServices\..., auf die nur SYSTEM, die Administratoren und das Dienstkonto zugreifen können. Sie werden in keiner Weboberfläche angezeigt – mit Ausnahme des Client-Pakets, einmalig beim Anlegen, und nur, solange die Weboberfläche auf 127.0.0.1 beschränkt ist. Behandeln Sie die Schlüsseldatei des Servers und Sicherungen davon wie ein Passwort.

Sicherheitshinweise für den Einsatz

- Mit **IPrAcc** werden TCP/IP-Verbindungen ermöglicht, die zuvor nicht möglich waren. Klären Sie deshalb vorab, ob alle Betroffenen (z.B. Ihre Kunden) damit einverstanden sind, und **dokumentieren Sie dieses Einverständnis**.
- Die zusätzlichen Verbindungsmöglichkeiten können auch ein potenzielles Sicherheitsrisiko darstellen, wenn z.B. Unbefugte auf Daten und Geräte zugreifen könnten. Einrichtung und Betrieb sollten daher nur durch qualifiziertes Fachpersonal erfolgen, das die Auswirkungen auf die Netzwerksicherheit versteht und geeignete Vorkehrungen trifft.
- Ein **Listen-Port** ist für alle erreichbar, die den PC im lokalen Netz erreichen. Wählen Sie die Adresse des Listen-Ports (0.0.0.0, 127.0.0.1 oder eine bestimmte Schnittstelle) und ggf. die Firewall des PCs so, dass nur die vorgesehenen Arbeitsplätze zugreifen können – und tragen Sie diese Arbeitsplätze zusätzlich unter **Erlaubte Absender** ein: Das ist die einfachste Absicherung und gilt auch für Quellen, die die Firewall nicht unterscheidet.
- Ein **UDP-Listen-Port** kennt keinen Verbindungsaufbau: Wer ihn erreicht, kann Datagramme einspeisen, und Absenderadressen lassen sich fälschen. Beschränken Sie den Zugriff über die erlaubten Absender des Eintrags, die Adresse des Ports und die Firewall des PCs auf die vorgesehenen Absender.
- Geräte, die über **IPrAcc** erreichbar gemacht werden, sollten selbst ausreichend geschützt sein (starke Passwörter, aktuelle Firmware), denn wer den Listen-Port erreicht, erreicht das Gerät.
- Verwenden Sie, wo möglich, verschlüsselte Protokolle (z.B. HTTPS) auch auf den Strecken innerhalb der lokalen Netze.
- Geben Sie am Server nur den einen Port `Listenport` für eingehende Verbindungen frei; alle anderen Ports sollten gesperrt bleiben.

Was IPrAcc nicht verhindert: Ein Client, dessen Rechner kompromittiert wurde, kann bis zu seiner Sperrung weiter (auch falsche) Meldungen senden und freigegebene Verbindungen seines Netzes anbieten – sperren Sie ihn dann sofort. Wird der Server-PC selbst kompromittiert, sind alle seine Client-Schlüssel betroffen und Angreifer könnten Verknüpfungen anlegen, die aber nur auf freigegebene Verbindungen der Clients wirken; legen Sie dann neue Clients an.

11 Digitale Souveränität

Durch Plattform-Unabhängigkeit (Windows, Linux) und die Unabhängigkeit von anfälligen zentralen Diensten wie dem DNS trägt **IPrAcc** zu mehr digitaler Souveränität bei. Alle Komponenten – Server, Clients, Schlüssel und Freigaben – liegen in Ihrer Hand, es gibt keinen Anbieter in der Mitte und keine Cloud, von der Verfügbarkeit oder Datensicherheit abhängen.

IPrAcc

- funktioniert auch bei Manipulationen, Überlastung (DDoS) oder Ausfall des DNS (man denke als Beispiel nur an die Störung bei Cloudflare vom 3.12.2025), weil die Clients den Server auch direkt über seine IP-Adresse erreichen (`publichost`);
- aktualisiert eine – z.B. nach der täglichen Zwangstrennung – geänderte IP-Adresse eines Kundennetzes wesentlich schneller als das DNS: Der Client baut die Verbindung nach einer Trennung selbst neu auf, und der Server kennt sofort die neue Absenderadresse (im ungünstigsten Fall kann es bis zu zwei volle Tage dauern, bis jeder DNS-Server im Internet die alte IP-Adresse vergessen hat);
- lässt jedem Beteiligten die Kontrolle über sein eigenes Netz: Was freigegeben wird, bestimmt der Administrator des jeweiligen LANs.

12 Download und Updates

IPrAcc Client und **IPrAcc** Server sind Bestandteil der **AccMicroServices** und werden mit dem EBÜS-Setup ins Verzeichnis `C:\EBUES\AccMicroServices` installiert. Aktuelle Versionen stehen mit den EBÜS-Setups unter www.ebues.de/setup bereit. Die Versionsnummer finden Sie in der Fußzeile dieses Dokuments und in der Startmeldung der Dienste.

13 Lizenz und Freischaltung

Der **IPrAcc** Server wird – wie die übrigen Produkte von Accellence – mit einem **Freischaltsschlüssel** freigeschaltet. Die Clients benötigen keine eigene Lizenz; maßgeblich ist die Lizenz des Servers. Die Anwendung heißt bei der Schlüsselerzeugung „IPrAcc2“ (Schlüsselversion 001).

13.1 Testmodus

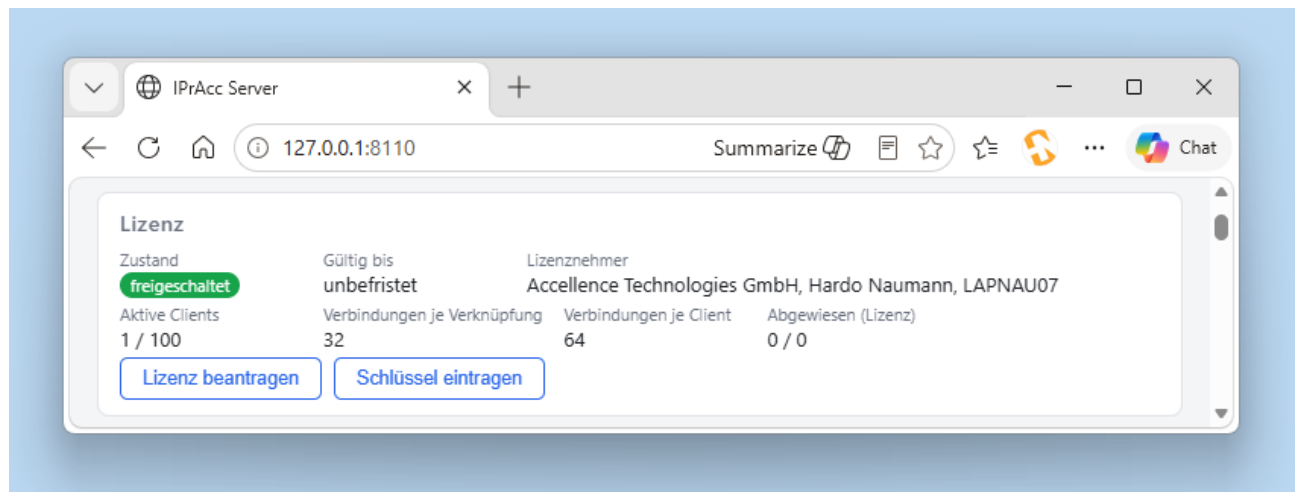
Ohne gültigen Schlüssel läuft der Server im **Testmodus**. Er ist voll funktionsfähig, aber im Umfang begrenzt:

- max. 2 gleichzeitig verbundene Clients
- max. 4 gleichzeitige Verbindungen je Verknüpfung
- max. 4 gleichzeitige Verbindungen je Client

Es dürfen mehr Clients angelegt sein, als gleichzeitig verbunden sein dürfen; die überzähligen Clients werden abgewiesen (sie versuchen es später erneut). Der Testmodus hat kein Ablaufdatum.

13.2 Die Karte „Lizenz“

Die Karte **Lizenz** in der Server-Weboberfläche zeigt den aktuellen Zustand:

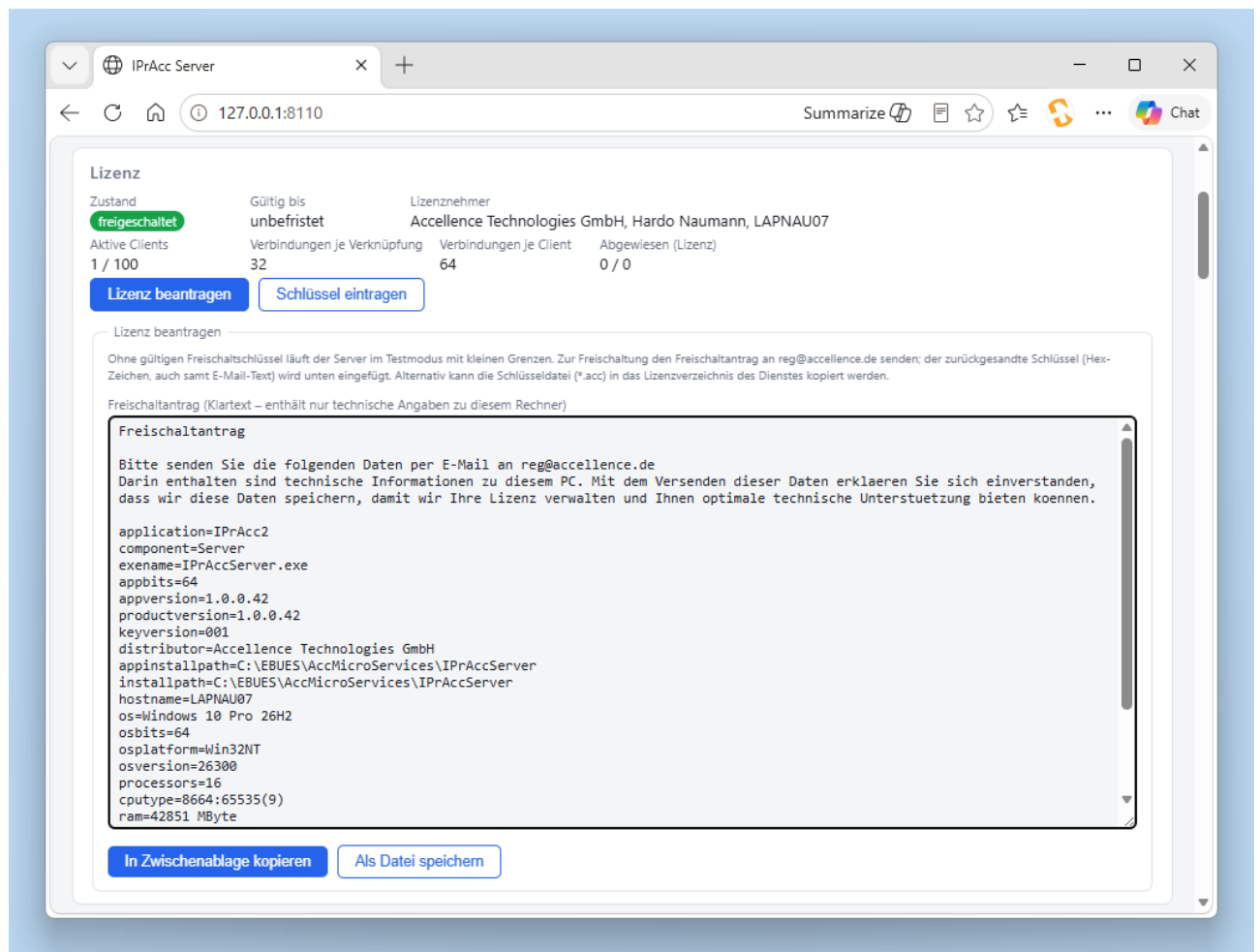


Anzeige	Bedeutung
Zustand	freigeschaltet oder Testmodus
Gültig bis	Ende des Lizenzmonats oder „unbefristet“. Läuft die Lizenz Ende des Monats ab, erscheint ein Hinweis, rechtzeitig einen neuen Schlüssel zu beantragen
Lizenznehmer	Der im Schlüssel eingetragene Lizenznehmer
Aktive Clients	Anzahl aktuell verbundener Clients / lizenzierte maximale Anzahl Sind mehr Clients angelegt als lizenziert, erscheint ein Hinweis auf die nötige Lizenzerweiterung
Verbindungen je Verknüpfung	Höchstzahl gleichzeitiger Verbindungen durch eine Verknüpfung Eine in der Konfiguration eingestellte kleinere Grenze (`maxstreamsperlink`) gilt weiterhin
Verbindungen je Client	Höchstzahl gleichzeitiger durchgeleiteter Verbindungen an einem Client (Listen- und Connect-Seite zusammen)
Abgewiesen (Lizenz)	Seit dem Start des Servers wegen der Lizenzgrenzen abgewiesene Clients und Verbindungen

13.3 Lizenz beantragen

Die Eingabefelder erscheinen im Sinne einer übersichtlichen Bedienoberfläche und klaren Benutzerführung erst, wenn Sie sie bewusst öffnen:

1. Auf **Lizenz beantragen** klicken. Es erscheint der ****Freischaltantrag**** dieses Rechners. Er ist ein Klartext und enthält nur technische Angaben zu diesem Rechner (u. a. einen daraus gebildeten Fingerabdruck), keine Daten Ihrer Kunden oder Verbindungen.

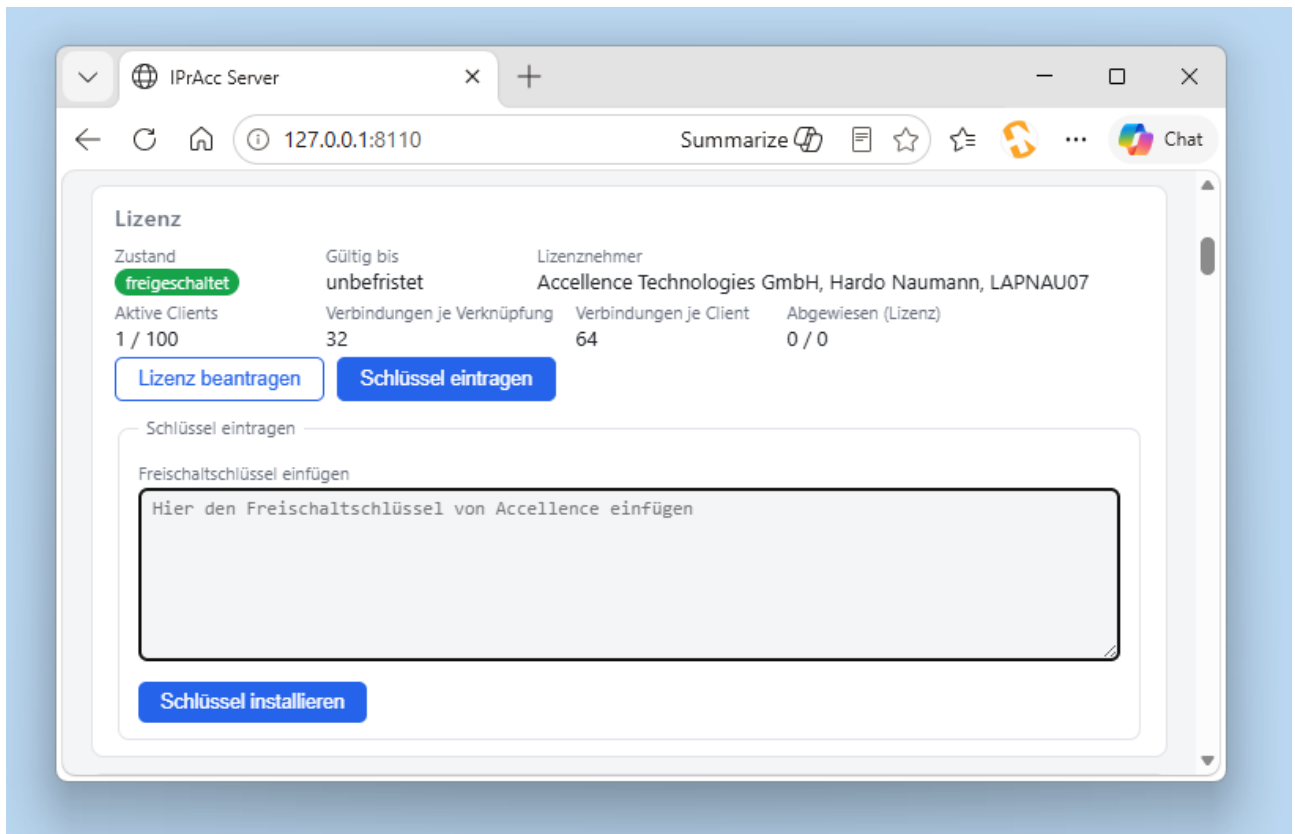


2. Mit **In Zwischenablage kopieren** oder **Als Datei speichern** sichern Sie den Antrag und senden ihn per E-Mail an reg@accellence.de. Alternativ liegt der Antrag als Datei 'FreischaltAntrag.txt' im Lizenzverzeichnis des Dienstes (siehe unten).

3. Accellence erzeugt aus dem Antrag den Freischaltsschlüssel und sendet ihn Ihnen zurück. Der Schlüssel besteht aus Hex-Zeichen; er darf auch samt dem umgebenden E-Mail-Text eingefügt werden.

13.4 Lizenzschlüssel installieren

4. Auf **Schlüssel eintragen** klicken, den Schlüssel in das Feld einfügen und auf die Schaltfläche **Schlüssel installieren** klicken.



Der Server prüft den hier eingetragenen Schlüssel und schaltet sich frei, wenn er für diesen Rechner gültig ist; die Karte zeigt sofort den neuen Zustand. Ein ungültiger oder für einen anderen Rechner ausgestellter Schlüssel wird mit Angabe des Grundes abgewiesen.

Ein erneuter Klick auf denselben Button oder ein Klick auf den anderen Button schließt das Eingabefeld wieder.

Alternative: Schlüsseldatei ablegen

Statt den Schlüssel in der Weboberfläche einzufügen, können Sie die Schlüsseldatei (Endung `.acc``) in das Lizenzverzeichnis des Dienstes legen:

```
`C:\ProgramData\AccMicroServices\IPrAccServer\license\`
```

(Bei einer benannten Instanz heißt der Ordner ``IPrAccServer.<Instanzname>``.) Der Server erkennt die Datei innerhalb von etwa 5 Sekunden. Das ist praktisch, wenn der Server per Fernwartung betreut wird und die Weboberfläche nicht erreichbar ist.

13.5 Was der Schlüssel festlegt

Der Schlüssel enthält die Begrenzungen der Lizenz:

Parameter	Bedeutung
maxclients	Anzahl der Clients, die gleichzeitig aktiv (d.h. verbunden) sein dürfen. Es dürfen mehr Clients angelegt sein.
maxstreamsperlink	Gleichzeitige Verbindungen durch eine Verknüpfung
maxtunnelstreams	Gleichzeitig durchgeleitete Verbindungen an einem Client
expiration	Ablaufmonat im Format `JJJJMM` (der letzte Monat, in dem die Lizenz gültig ist, z.B. `202610` für eine Lizenz bis Ende Oktober 2026) oder `NO` für eine unbefristete Lizenz

Fehlt ein Parameter im Schlüssel, gilt der Wert des Testmodus.

13.6 Gültigkeit, Ablauf und Rückfall

- Ein Schlüssel gilt für **einen Rechner** und **eine Dienstinstanz**. Laufen mehrere benannte Instanzen auf einem Rechner, benötigt jede einen eigenen Schlüssel.
- Der Ablauf wird laufend geprüft. Nach dem Ablauf – oder wenn die Schlüsseldatei entfernt wird – fällt der Server in den **Testmodus** zurück: Die über die Testgrenze hinausgehenden Clients werden getrennt, der Freischaltantrag liegt wieder im Lizenzverzeichnis. Bereits angelegte Clients, Verknüpfungen und Einstellungen bleiben erhalten.
- Wird ein neuer Schlüssel installiert, gelten die neuen Grenzen sofort; die vorher abgewiesenen Clients verbinden sich selbstständig wieder.
- Ändert sich die Hardware des Rechners wesentlich (Mainboard, Prozessor, Festplatte), kann der Schlüssel ungültig werden. Senden Sie dann einen neuen Freischaltantrag.
- Der Schlüssel überlebt Neustarts des Dienstes.

13.7 Hinweise für den Betrieb

HINWEIS: Eine **Lizenerweiterung** ist nötig, wenn mehr Clients gleichzeitig verbunden sein sollen als lizenziert sind. Die Karte Lizenz zeigt dann einen Hinweis; im Log des Servers steht, welche Clients wegen der Lizenzgrenze abgewiesen wurden. Der betroffene Client nennt den Grund in seiner Statusanzeige („Lizenzgrenze des Servers erreicht“).

HINWEIS: Verbindungen, die wegen der Grenzen „je Verknüpfung“ oder „je Client“ abgewiesen werden, erscheinen im **Verbindungsjournal** als `REFUSED` mit der Begründung (Abschnitt „Aktive Verbindungen, Statistik und Verbindungsjournal“).

14 Was tun wenn ...

Prüfen Sie zuerst die Statusanzeige und das Log der betroffenen Weboberfläche (Bereich **Log** am Ende der Seite) beziehungsweise die Konsolenausgabe.

Typische Meldungen des Clients im Feld **Verbindung** und im Log:

Meldung	Ursache und Abhilfe
„Server und Client-ID sind nicht konfiguriert“	Es wurde noch kein Client-Paket importiert (Abschnitt 4.1).
„Schlüsseldatei ... nicht lesbar“ bzw. „enthält keinen gültigen 256-Bit-Schlüssel“	Das Client-Paket fehlt oder wurde unvollständig kopiert. Paket erneut importieren; ist es nicht mehr vorhanden, auf dem Server einen neuen Client anlegen.
„Schlüsselverzeichnis kann nicht abgesichert werden – gehört einem Benutzer, der weder SYSTEM noch Administrator ist“	Der Dienst startet nicht (Windows meldet beim Starten einen Fehler). Der Ordner %ProgramData%\AccMicroServices (bzw. ein Unterordner) gehört einem normalen Benutzer; IPrAcc verwendet solche Ordner nicht, weil ein Benutzer dort die Zugriffsrechte wieder ändern und die Schlüssel lesen könnte. Geht der Ordner an ein Administrator-Konto oder SYSTEM, erscheint die Meldung nicht. Abhilfe: In einer Eingabeaufforderung als Administrator <code>takeown /F "%ProgramData%\AccMicroServices" /A</code> ausführen (Besitzer = Administratoren) und den Dienst erneut starten – oder den Ordner löschen, wenn er nur Testdaten enthält.
„Sername ... konnte nicht aufgelöst werden“	Der Name aus server ist im Netz des Clients unbekannt. Namen prüfen oder in publichost des Servers eine IP-Adresse verwenden.
„Verbindung abgelehnt oder nicht erreichbar“ / „Zeitüberschreitung beim Verbindungsaufbau“	Der Server läuft nicht, oder Port bzw. Adresse stimmen nicht, oder eine Firewall/ein Router blockiert den Port (Abschnitt 3.10). Prüfen Sie im Server-Status, ob Sichere Verbindungen aktiv sind. Der Client versucht es automatisch weiter.
„Handshake fehlgeschlagen ...“	Client-ID oder Schlüssel stimmen nicht, der Client wurde gesperrt oder gelöscht, oder die Quelle wurde wegen vieler Fehlversuche vorübergehend gesperrt bzw. durch eine Zugriffsbegrenzung abgewiesen. Der Server unterscheidet das nach außen bewusst nicht; im Server-Log und im Bereich Clients sehen Sie die Ursache.
„Anmeldung nicht möglich: Handshake fehlgeschlagen ...“ (Anmeldung mit Einmalcode)	Der Code ist falsch, abgelaufen oder schon benutzt, oder Server und Port stimmen nicht bzw. sind nicht erreichbar. Der Client kann das nicht unterscheiden (Absicht). Eingaben prüfen, bei Bedarf einen neuen Code in der Leitstelle erzeugen lassen.
„Die Leitstelle hat die Anmeldung abgelehnt“ / „Die Zeit ... ist abgelaufen“	Die Bestätigungszahl stimmt nicht überein oder die Leitstelle hat nicht rechtzeitig bestätigt (<code>enrollapprovalseconds</code>). Neuen Code erzeugen und das Telefonat wiederholen.
„Getrennt: keine Daten vom Server seit ... s“ / „keine Quittung vom Server“	Die Verbindung ist gestört (Netz unterbrochen, Server überlastet oder neu gestartet). Der Client baut sie selbstständig neu auf; die Zeit bis zum nächsten Versuch steht in der Anzeige. Bleibt es dabei, Netz und Server prüfen.
„Getrennt: Der Server hat die Verbindung beendet“	Der Server hat die Verbindung bewusst beendet (z.B. weil der Client gesperrt wurde oder der Server die Konfiguration neu geladen hat). Der Client versucht es automatisch erneut.
„Fehler: Der Server erlaubt diesem Client keine Listen-Ports.“ / „Fehler: Der Port ... liegt laut Richtlinie des Servers nicht im erlaubten Bereich.“ (Zustand eines Listen-Eintrags)	Die Listen-Richtlinie des Servers (Abschnitt 3.4) verbietet diesem Client das Listen bzw. diesen Port. Der Server-Administrator kann es in der Clientliste erlauben, oder Sie wählen einen Port im erlaubten Bereich (die Richtlinie steht unter der Verbindungstabelle).
„Quittung abgelehnt: message_uuid ...“	Die Quittung gehört nicht zur gesendeten Meldung. Das sollte nie vorkommen und deutet auf eine Manipulation oder einen Fehler der Quittungsvorlage des Servers hin.

Probleme bei den Verbindungen und der Durchleitung:

Beobachtung	Ursache und Abhilfe
Die Verknüpfung bleibt auf „Listen-Port am Client nicht freigegeben“ bzw. „Gegenstelle am Client nicht freigegeben“	Der Administrator des betreffenden Client-Netzes muss in der Tabelle Verbindungen den Haken Freigabe setzen und auf Speichern und melden klicken. Das kann der Server-Administrator nicht für ihn tun.
Zustand eines Listen-Ports am Client: „Fehler: ...“	Der Port ist bereits von einem anderen Programm belegt oder die Adresse gehört nicht zu diesem PC. Anderen Port bzw. eine Adresse dieses PCs wählen; der Client versucht es alle 10 Sekunden erneut.
Am Listen-Port wird die Verbindung sofort wieder geschlossen	Die Verknüpfung ist nicht bereit (siehe Zustand am Server), die Adresse des Absenders steht nicht in den Erlaubten Absendern des Eintrags (in der Spalte Zustand zählt „[n abgewiesen]“; im Log steht „... abgewiesen: Die Adresse steht nicht in der Liste der erlaubten Absender“ – den Absender in die Liste aufnehmen oder die Liste leeren), der Client ist nicht mit dem Server verbunden, das Limit gleichzeitiger Verbindungen ist erreicht, oder die Gegenstelle (z.B. die Kamera) ist nicht erreichbar. Die Ursache steht im Log des Listen-Clients und der Gegenstelle („Durchleitung: ...“).
Die Gegenstelle erreicht die Kamera nicht („Verbindung zu ... fehlgeschlagen“)	IP-Adresse oder Port des Connect-Eintrags stimmen nicht, das Gerät ist ausgeschaltet oder blockiert den PC des Clients. Prüfen Sie vom PC des Clients aus, ob das Gerät unter dieser Adresse erreichbar ist.
Das Bild der Kamera erscheint nicht, obwohl die Verbindung steht	Das Gerät verwendet für das Bild weitere Verbindungen (z.B. UDP oder andere Ports). Tragen Sie die zusätzlichen Ports ebenfalls als Verbindungen ein (bei UDP mit Protokoll UDP an beiden Enden) bzw. stellen Sie das Gerät auf Übertragung per TCP um.
UDP-Daten kommen nicht an oder nur gestört	Prüfen Sie, ob beide Enden das gewünschte Protokoll haben (Spalte Protokoll) und ob der Absender die Adresse des UDP-Listen-Ports verwendet (nicht die der Gegenstelle). Ist die Verbindung überlastet, werden Datagramme verworfen; ein Datagramm über 60000 Bytes wird verworfen. Bei längeren Pausen (mehr als <code>udpidleseconds</code>) beginnt der Datenstrom neu.
Verknüpfung lässt sich nicht anlegen: „... darf laut Richtlinie keine Listen-Ports anbieten“ bzw. „... liegt nicht im Portbereich ...“	Der Listen-Eintrag verletzt die Listen-Richtlinie des Clients (Abschnitt 3.4). Passen Sie die Richtlinie in der Clientliste an, oder lassen Sie am Client einen erlaubten Port eintragen.
Beim Verknüpfen wird eine Verbindung nicht zur Auswahl angeboten	Der Client hat sie noch nicht gemeldet (Client nicht verbunden, oder auf Speichern und melden wurde nicht geklickt) oder sie hat die falsche Art (Listen/Connect).

Mögliche Probleme des Servers:

Beobachtung	Ursache und Abhilfe
Status zeigt „inaktiv“ mit Fehlermeldung zum Port	Der Port <code>listenport</code> ist bereits belegt oder die Adresse ungültig. Anderen Port wählen und speichern – der Server versucht es automatisch erneut.
Ein Client steht auf „Getrennt“ oder „überfällig“	Der Client läuft nicht (Dienst gestoppt, System aus, Netz gestört) oder wird abgewiesen. Im Client-Log und im Server-Log nachsehen. Die Frist stellt <code>staleafterseconds</code> ein.
Clients werden nach einer Weile getrennt und bauen neu auf	<code>keepaliveseconds</code> der Clients ist nicht deutlich kleiner als <code>idletimeoutseconds</code> des Servers, oder ein Router zwischen Client und Server beendet untätige Verbindungen früher. <code>keepaliveseconds</code> verkleinern.
Statusdateien werden nicht geschrieben	<code>statuspath</code> ist nicht beschreibbar (Rechte des Dienstkontos prüfen). Das Server-Log nennt den Pfad.
Browser meldet „403 Forbidden“ bzw. „Unerwarteter Host-Header“	Die Weboberfläche wurde über einen anderen Namen als <code>127.0.0.1</code> oder <code>localhost</code> aufgerufen (z.B. über den Rechnernamen). Rufen Sie sie über <code>http://127.0.0.1:<Port>/</code> auf; für den Zugriff aus dem Netz den <code>AccWebUiGateway</code> verwenden.
„403 Forbidden: Sicherheits-Token fehlt oder ist ungültig“ (in der Weboberfläche)	Die Seite wurde geladen, bevor der Dienst neu gestartet wurde; das Sicherheits-Token gilt nur für einen Dienstprozess. Die Weboberfläche erneuert es beim ersten Fehler selbst und wiederholt die Aktion; bleibt die Meldung bestehen, die Seite mit F5 neu laden.
Änderung eines Pfades wird abgelehnt	Über die Weboberfläche sind nur relative Pfade unterhalb des Programmordners zulässig (Kapitel 10). Absolute Pfade direkt in der <code>cfg-</code> Datei eintragen.
Viele Clients hinter einer Adresse werden abgewiesen	<code>maxperip</code> bzw. <code>newconnpersec/newconnburst</code> sind für die gemeinsame Adresse (NAT) zu niedrig eingestellt.
Neue Clients werden abgewiesen, obwohl der Server läuft	<code>maxconnections</code> (Standard 256) ist erreicht: Jeder Client hält eine stehende Verbindung. Wert erhöhen.

15 Support / Hotline

Weitere Informationen zu **IPrAcc** finden Sie stets aktuell unter www.ipracc.com

Haben Sie noch Fragen oder Wünsche zu **IPrAcc**?

Dann wenden Sie sich bitte

- telefonisch unter 0511 - 277.2490
- per E-Mail an support@accellence.de

an unsere Hotline. Wir sind Werktags von 9:00-17:00 Uhr zu erreichen.

Wir wünschen Ihnen viel Erfolg bei Ihrer Arbeit mit **IPrAcc** und stehen für Ihre Wünsche und Fragen jederzeit gern zu Ihrer Verfügung.