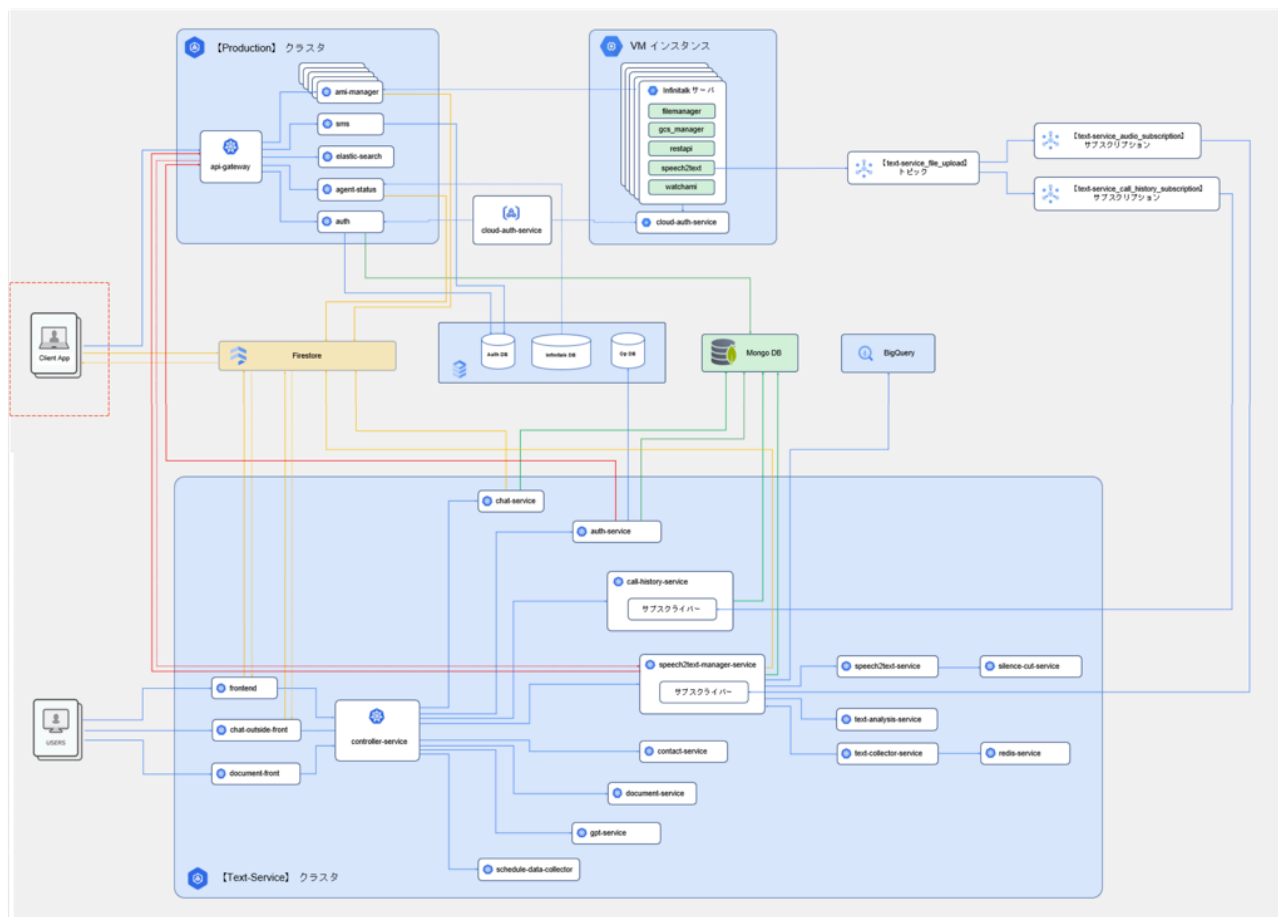


CLIENT APP

Box\CCaaS開発\2.TextService\1.GKE



I. ClientApp 概要

- ClientAppは、SIP/WebRTCを通じて通話の発信および着信を行うためのデスクトップアプリ（Electronjs、Reactjs、Typescript）です。

JsSIPライブラリを使用してSIP登録および通話管理を行います。Asteriskとのpersistent connectを維持して、VOIP機能を実現します。

ClientAppとシステム（asterisk、GCP、firebase、firestore、text service web）との関係性。

クラウド顧客（v5、v8）向けで、Softphoneでの通話や、ClientAppのみインストールして電話帳ポップアップさせる顧客もいます（ClientApp+AgentStatusToolを併用）。

II. 重要な構成要素

1. ClientAppの通信先 (Firewall)

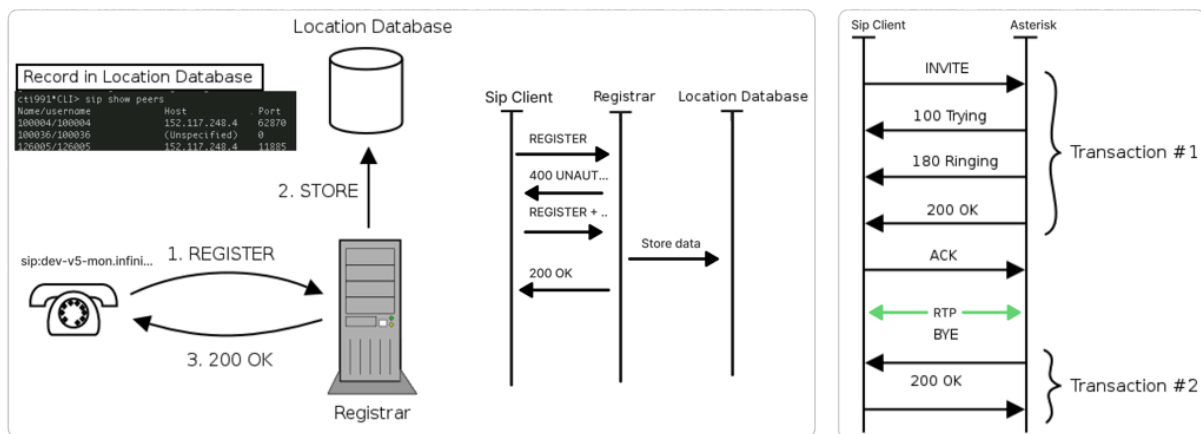
```
1 1. Controllerサーバ:
2 - controller.infinitalk.net:443
3
4 2. Api-gatewayサーバ:
5 - api.infinitalk.net:443
6
7 3. ビデオコール用Asteriskサーバ:
8 - cti-videocall.infinitalk.net:8089
9
10 4. ソフトフォン用のAsteriskサーバ
11 4.1. VPNなしの場合 (現状)
12 - Asteriskサーバドメイン名:8089
13 - Asteriskサーバドメイン名:10000~20000 (SRTP)
14
15 4.2. VPNの場合 (将来機能)
16 - GCP AsteriskサーバプライベートIP:8088
17 - GCP AsteriskサーバプライベートIP: 10000~20000 (SRTP)
18
19 5. Firestore
20 - www.google.com:443
21 - www.googleapis.com:443
22 - firestore.googleapis.com:443
23 - commercial-001.firebaseio.com: 443
24
25 6. 外部チャット用のサーバ
26 - chat.infinitalk.net:443
27
28 7. ネットワーク不安定確認用のサーバ
29 - httpbin.org:443
30
31 8. チャットハドル機能用のスタンサーバ
32 - stun.l.google.com:19302
33 - stun1.l.google.com:19302
34 - stun2.l.google.com:19302
35 - stun3.l.google.com:19302
36 - stun4.l.google.com:19302
```

2. SIP / JsSIP

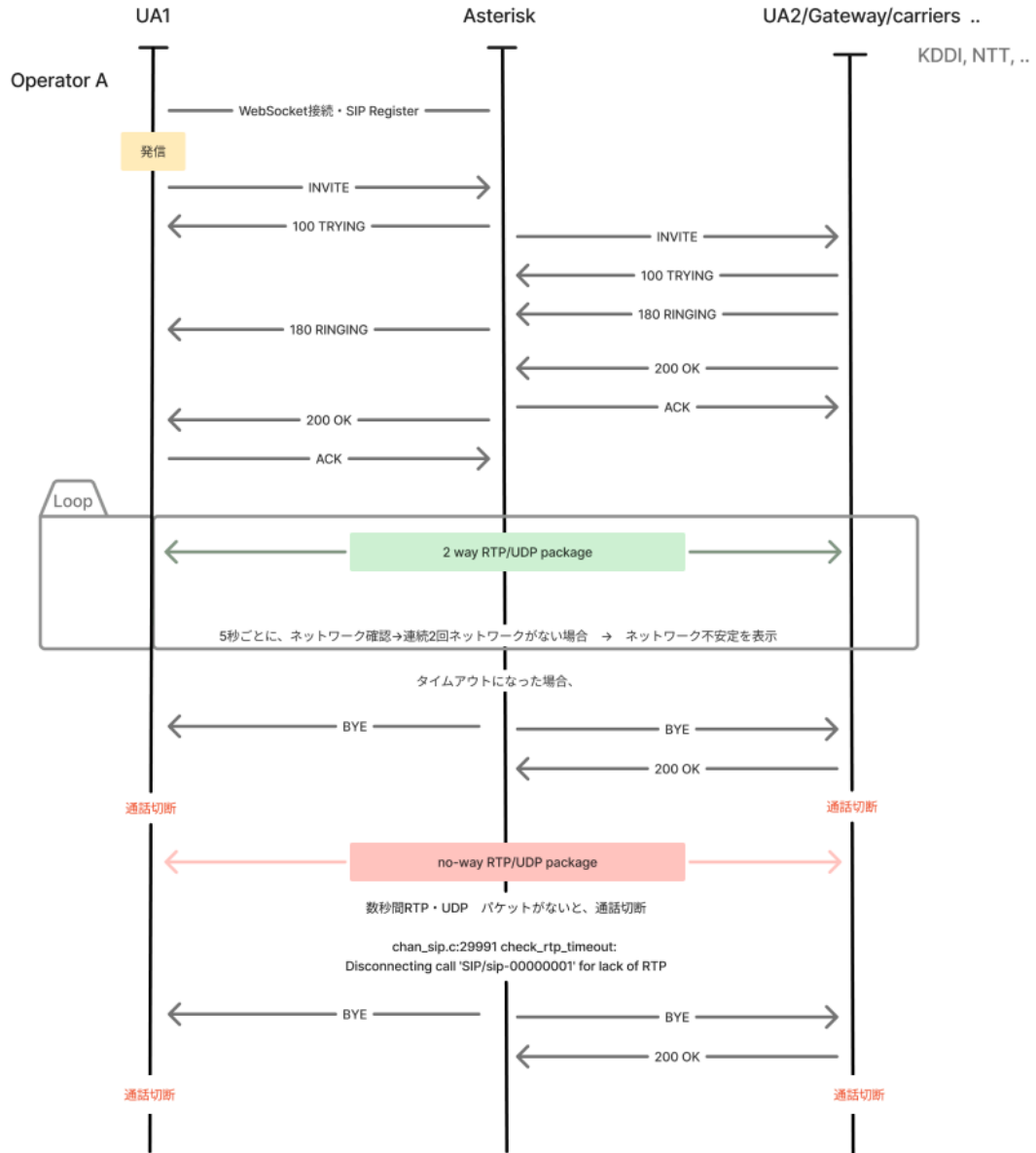
- JsSIPはSIP UAを作成するために使用されます。SIP処理に関するロジックは `src/repositories/infinitalkSIP.ts` にあり、SIP登録、UAイベントのリッスン、通話接続などを含みます。
- JsSIPの関数に関するドキュメントは公式のdocsを参照してください [JsSIP - API](#)
- 主な関数は以下の通りです
 - ClientAppからSoftphoneにログインする際、SipAccountの種類 (VPN/VPNなし) に応じて、Asteriskの8088/8089ポートに接続します。登録後、AsteriskのCLIまたは `infinitalkdb/ast_sip_buddies` で確認可能です。

- **call** : 通常の通話フロー以外にも、自動アナウンス再生するフローがあります。
- **overwriteRemoteDescription** : Asteriskと接続する際、複数のcandidateの中から最も短い経路を自動で選択します。privateIPで通信できる場合は問題ありませんが、できない場合は状態を監視して、必要に応じてcandidateを更新する必要があります。
- **listenCall** : 'incoming', 'progress', 'confirm', 'end', 'failed' ... などのイベントをリッスンする場所です。
- **answer**, **sendDTMF**, **refer**, **hold**, **unhold** ...

• SIPの基本

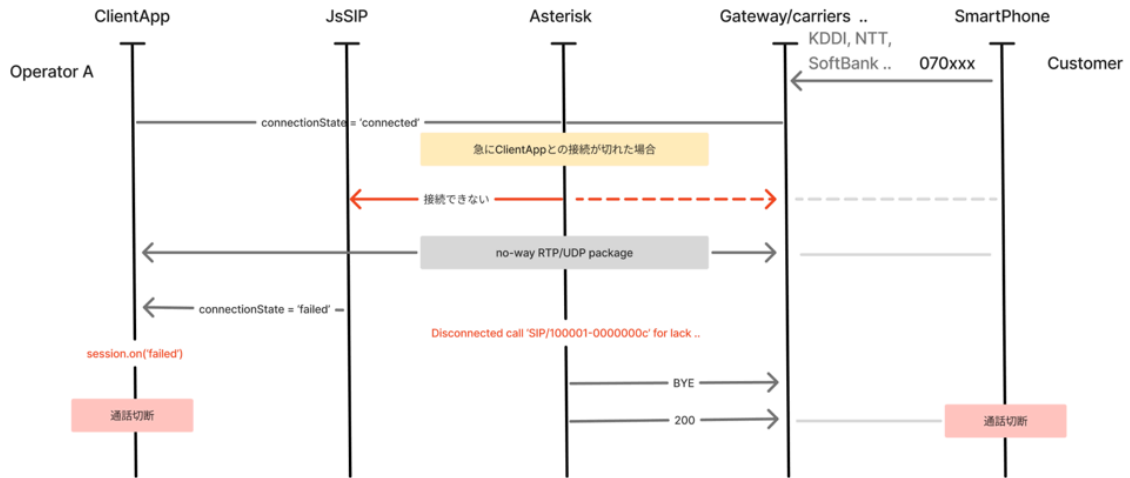


SIP通信



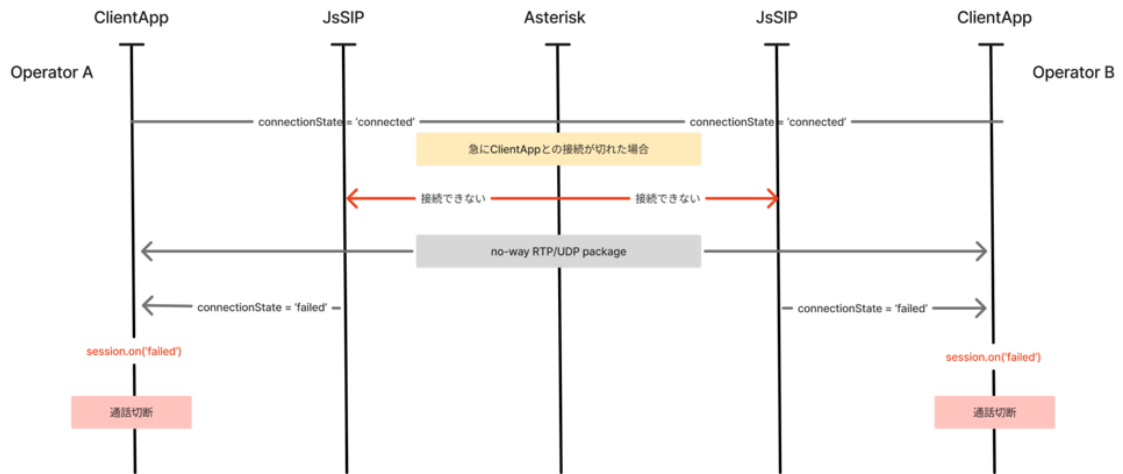
Asteriskとの接続が失う場合

外線発着信

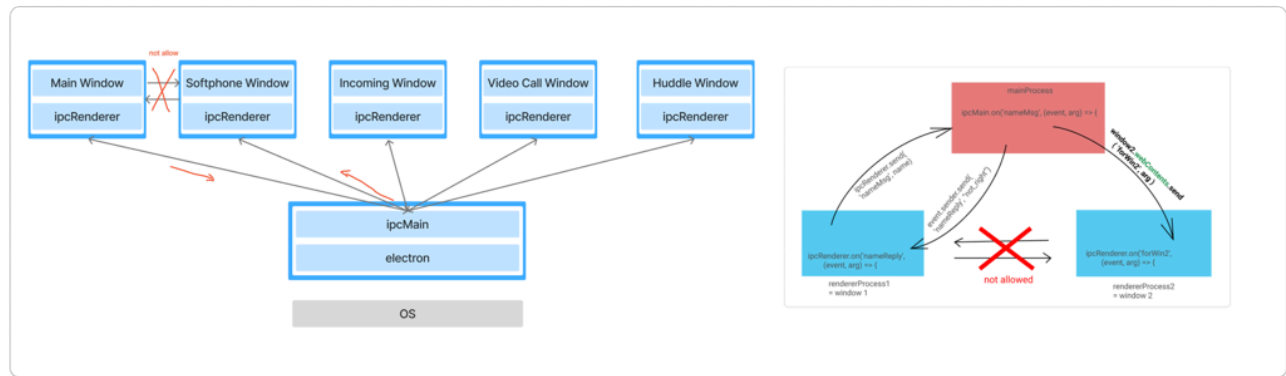


Asteriskとの接続が失う場合

内線発着信



2. Electron関連



- **electron**：メインのロジックは `core.js` にあります。
- **renderer process**：各ウィンドウのことを指し、Main Window（jsipの処理を行う場所）、Softphone Window、Incoming Windowなどが含まれます。
- 複数のRenderer Process間でデータをやり取りする場合は、Main Processを中継として使用します。

renderer process 1 → main process → renderer process 2

📌 各ウィンドウは独自のstoreを持っているため、recoilのデータ使用時は注意が必要です。

例：Main Windowでstateをrecoilに保存している場合、Softphone Windowでそのデータを使いたいときは、Main WindowのデータをSoftphone Windowに送信して更新する必要があります（データはMain Windowのstoreに保持されているため、Softphone Windowでは自動的に取得できません）。

データ送信の仕組みとstate変更イベントの検知を組み合わせて使用します：

renderer process 1：state変更 → 新しいstateをrenderer process 2に送信 → main process 経由 → renderer process 2：stateを更新

III. デバッグ

ClientAppのログ、SIP/RTP、Wireshark、Asterisk CLIなどのツールを活用することは、デバッグ時に非常に重要です。

1. 📝 App Log (React Native Console)

- App Logは、アプリの操作ログを記録します。例：ユーザーのアクション（Softphoneログイン、Call/Answerクリック）、speakerのdeviceId、windowのスリープやサスペンド、SIPフローなど。
- 保存場所： `AppData\Roaming\client-app\logs`
- 通話のUAイベント（connecting、progress、ended、failedなど）、MediaStreamが呼び出されたタイミング、その他の関連ログを活用することで、障害の推測に役立ちます。
- これはカスタムログなので、調査には十分でない場合があります。調査に必要なログを検討し、ログ出力を強化することをおすすめします。

2. 📶 Firestore (Cloud log)

- SIP log： `Storage/Debug/SIP/...`
SIP register、SIP INVITEなど
- Dump Call → `%2300` によりSoftphoneが初期化されたタイミングを特定可能
- RTP log： `Storage/Debug/RTP/...`
jitter、bufferJitterなど音声に関するパラメータを確認可能

3. JsSIP Debug Log

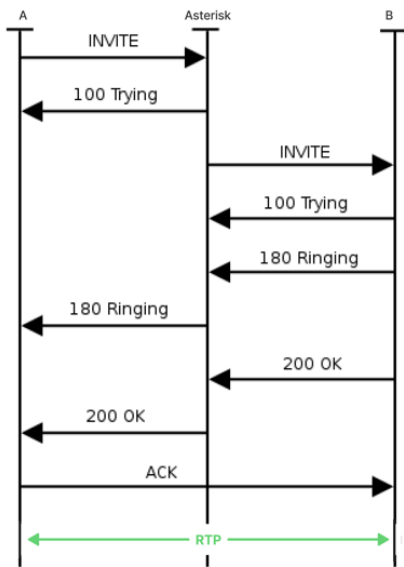
- ローカル開発時には `JsSIP.debug.enable('JsSIP:*')` を有効にして、アプリから発生するすべてのSIPの動作をconsoleで確認できます。

SIPログには、以下のような重要な情報が含まれています：

Asteriskから受信またはアプリから送信されたメッセージ、IP/Port、linkedId、contact-info、アプリバージョン、SDP、candidateなど。

BYEやACKのメッセージには、`reason`（例：16 Normal Clearing）が含まれる場合もあります。

```
1 JsSIP:RTCSession receiveInvite response: 180 Ringing
2 JsSIP:RTCSession sendDTMF() | tones: 1 +5s
3
4 JsSIP:InviteClientTransaction Timer B expired for transaction z9hG4bK4295174 +6s
5 JsSIP:RTCSession confirmed
6 JsSIP:RTCSession ended
7
```



SIP message info

SDP

```

<--- SIP read from WS:152.117.248.4:55411 --->
INVITE sip:09040591977@cti99.infnitalk.net SIP/2.0
Via: SIP/2.0/WS ct:99.infnitalk.net;branch=29h6dK7379927
Max-Forwards: 69
To: <:sip:09040591977@cti99.infnitalk.net>
From: "JMS小佐野(rtc)" <:sip:126076@cti99.infnitalk.net>;tag=7n51ejk2Zm
Call-ID: 54oh9m6192pi14ecfm77
CSeq: 2036 INVITE
Authorization: Digest algorithm=MD5, username="126076", realm="asterisk",
nonce="65da58dd", uri="sip:09040591977@cti99.infnitalk.net",
response="7d9f52cb192f981159f14879b18338d7"
Contact: <:sip:126076@cti99.infnitalk.net;transport=ws;ob>
Content-Type: application/sdp
Allow: INVITE,ACK,CANCEL,BYE,UPDATE,MESSAGE,OPTIONS,REFER,INFO,NOTIFY
Supported: ice,replaces,outbound
User-Agent: Infnitalk ClientApp Ver1.19.0
Content-Length: 2033

v=0
o=- 2885682843760910970 2 IN IP4 127.0.0.1
s=-
t=0
a=group:BUNDLE 0
a=extmap-allow-mixed
a=msid-semantic: WMS 74cb798-a1cb-4d49-ac3a-b5081b0c785e
m=audio 61913 UDP/TLS/RTP/SAVPF 111 63 103 104 9 0 8 106 105 13 110 112 113 126
c=IN IP4 192.168.128.140
b=RTCP:61914 IN IP4 192.168.128.140
a=candidate:2484721060 1 udp 2122260223 192.168.128.140 61913 typ host
generation 0 network-id 1
a=candidate:2484721060 2 udp 2122260222 192.168.128.140 61914 typ host
generation 0 network-id 1
a=candidate:3667536212 1 tcp 1518280447 192.168.128.140 9 typ host tcptype
active generation 0 network-id 1
a=candidate:3667536212 2 tcp 1518280446 192.168.128.140 9 typ host tcptype
active generation 0 network-id 1
a=ice-ufrag:j/95
a=ice-pwd:p13ELJ90KxjrD0hpIM54dJ3
  
```

4. Wireshark

- SIP、SDP、RTPパケットの送信方向や内容を分析します。

例えば、アプリのログではパケットを送信したと記録されているのに、Asteriskが受信していない場合：

→ パケットが本当にアプリから送信されたか、どのアドレスに送られたか、インターネットでブロックされたか、Asteriskに届いたか、パケットの内容などを確認します。

```

1 sip || rtp
2 ip.addr == 34.27.10.214 && tcp.port == <port RTP>
3
  
```

No.	Time	Source	Destination	Protocol	Length	Info
5524	17:05:17.723897	10.128.0.51	192.168.128.124	UDP	224	10034 → 8000 Len=182
5525	17:05:17.725597	192.168.128.124	163.220.246.42	TLSv1.2	394	Application Data
5526	17:05:17.732371	163.220.246.42	192.168.128.124	TLSv1.2	127	Application Data
5527	17:05:17.735689	10.128.0.51	192.168.128.124	UDP	224	10034 → 8000 Len=182
5528	17:05:17.740000	192.168.128.124	10.128.0.51	UDP	224	8000 → 10034 Len=182
5529	17:05:17.745639	192.168.128.124	163.220.246.42	TLSv1.2	398	Application Data
5530	17:05:17.748056	163.220.246.42	192.168.128.124	TLSv1.2	148	Application Data
5531	17:05:17.755173	10.128.0.51	192.168.128.124	UDP	224	10034 → 8000 Len=182
5532	17:05:17.763596	163.220.246.42	192.168.128.124	TLSv1.2	148	Application Data

5. RTP / Media Debug

音声に関する問題（音が出ない、音声の遅延など）が発生した場合、以下を確認します：

- RTPログ：RTPパケットが送受信されているかどうか確認
- アプリ動作のログ：マイク/スピーカーに関するログを確認
- 通話中のストリームログ：途中で切断されたか、過負荷状態か、開放されていないかなどを確認

また、Wiresharkを使用してRTPの方向を確認し、どこへ送信されているか、Asteriskに届いているか、NATにブロックされているかなどを調べます。

6. Asterisk CLI

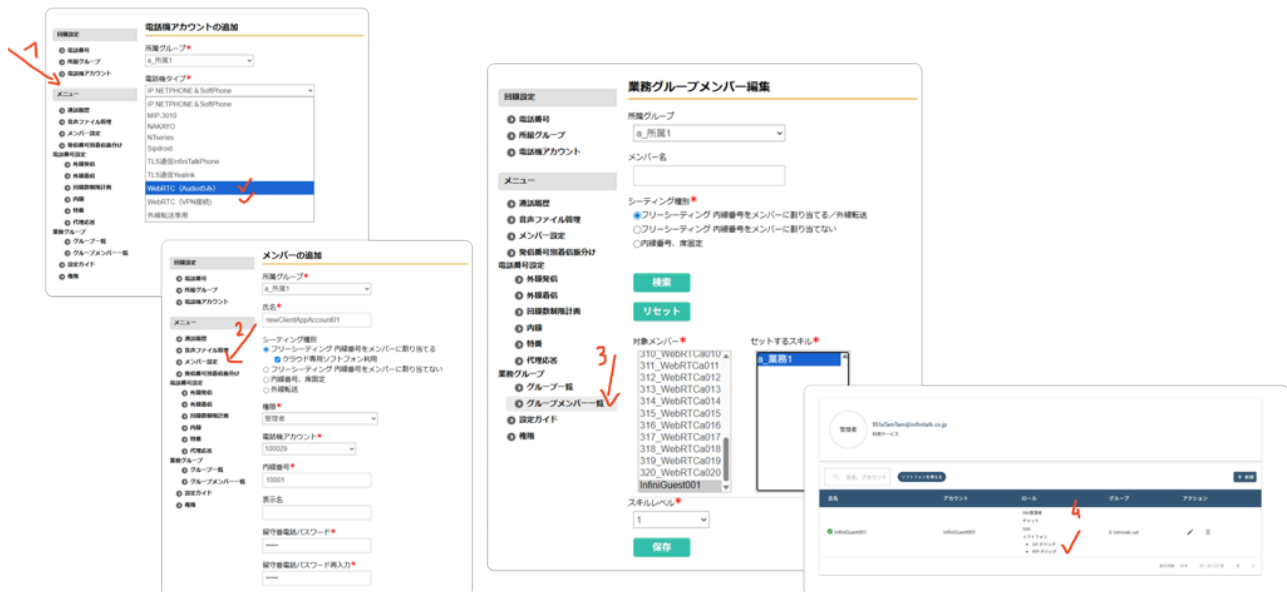
```
1 asterisk -rvvvvvvvvvvvvvvvv
2 core show channels
3 bridge show all
4 dialplan show
5 sip show peers
6 sip show peer 10001
7 sip set debug on
8 rtp set debug on ..
```

IV. Một số tính năng quan trọng

1. エージェント状態変更

Softphoneを使用するには、SoftphoneLicenseを持つアカウントでログインする必要があります。

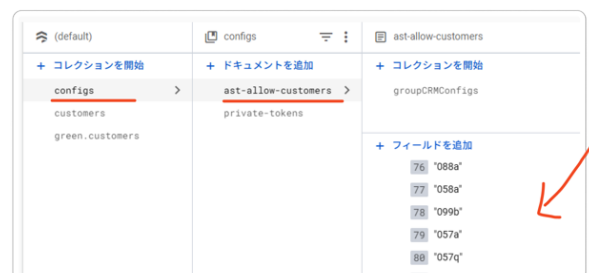
以下は、アカウントを作成し、ユーザーにSoftphoneの使用権限を付与するフローです。



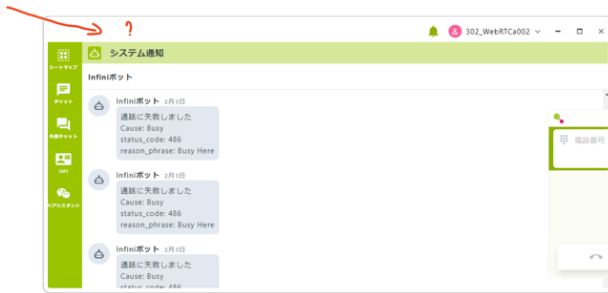
ただし、開発環境では以下のような現象がよく発生します：

アカウントを作成し、ライセンスを付与した後でも、AgentStatusバーが表示されない。

→ Firebaseの `ast-allow-customers` コレクションに



customerId を追加しているかを確認してください。



2. Softphone：発信・着信・電話帳ポップアップ表示

a. 発信・着信

・ 発信：

番号を入力して発信、tel-app/080111…での発信、SeatMapやモニタービューからの発信が可能。

・ 着信：

- 。待機中：内線・外線の着信どちらも受信可能
- 。ワーク、離席 など：内線着信のみ受信可能
- 。呼出中、発信中、通話中、受信中 …：着信を受け取れない状態

0 待機中

1 ログオフ

2 ワーク

3 離席

7 自動ワーク

78 発信:呼出中

79 発信:通話中

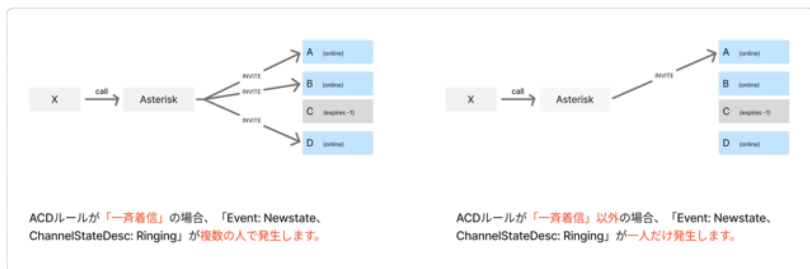
88 ACD以外着信:
呼出中

89 ACD以外着信:
通話中

98 ACD着信:呼出
中

99 ACD着信:通話
中

- ・ Asteriskからの着信には、ACDルールやダイヤルプランなどにより、主に2種類のIncomingパターンがあります。



b. ポップアップ表示・対応履歴

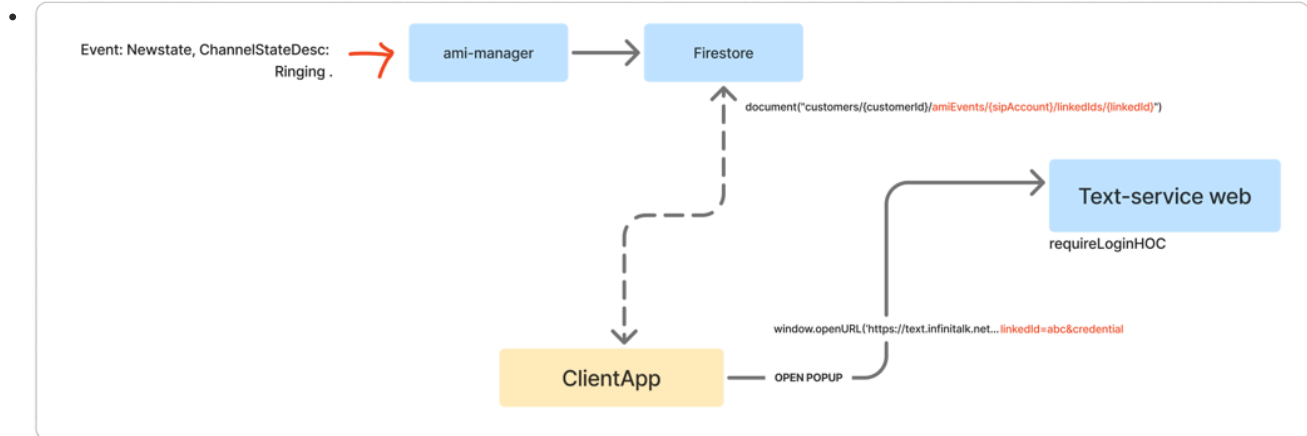
ポップアップ機能の仕組みは、Web側（text-service）は `loginHOC` をラップし、アプリ側は `window.open(url)` を使ってブラウザを開きます。

そのURLには、`credential` が含まれており、自動で `loginHOC` にログインできます。

具体的には、`ami-manager` がAsteriskからの通話イベントを受け取り、通話データを `linkedID` ごとにFirestoreへ保存します。

ClientAppは `customer/{customerId}/amiEvents/` コレクションをスナップショット監視し、`window.open(url)` を実行してポップアップを開きます。

開かれたWebでは、`loginHOC` により `credential` を使って自動ログインが行われます。



- ポップアップ表示設定および通話履歴保存について（グループコードの一致が必要）

The screenshots show the application interface for managing group codes and call history. The left screenshot shows the 'グループコード' (Group Code) settings with a red checkmark. The middle screenshot shows the 'グループコード' settings with a red checkmark. The right screenshot shows the '通話履歴' (Call History) table with a red checkmark.

日時	グループ名	発信者	発信番号	種別	通話先	通話時間	通話状態	通話履歴
2023/03/11 11:10:00	01-10000000	01-10000000	01-10000000	入電	01-10000000	00:00	通話中	通話履歴
2023/03/11 11:10:00	01-10000000	01-10000000	01-10000000	入電	01-10000000	00:00	通話中	通話履歴
2023/03/11 11:10:00	01-10000000	01-10000000	01-10000000	入電	01-10000000	00:00	通話中	通話履歴
2023/03/11 11:10:00	01-10000000	01-10000000	01-10000000	入電	01-10000000	00:00	通話中	通話履歴
2023/03/11 11:10:00	01-10000000	01-10000000	01-10000000	入電	01-10000000	00:00	通話中	通話履歴

- ・グループコードは必ず一致させる必要があります。
- ・通話履歴はグループコード単位で取得されます。
- ・すべての設定が完了しているにもかかわらずポップアップが動作しない場合は、ami-managerのイメージ名を確認してください。

V. あまり使われない機能と設定に関する注意点

1. CRMアカウントでのアプリ利用

- ・CRM単体の顧客：CRM、text-serviceのみを購入し、Infinitalkサーバを購入していない顧客です。
- ・ClientAppでは大きな変更はなく、Chat、ChatOutside、SMS機能は通常通り使用可能です。
- ・SeatMapやSoftphoneのAgentStatusは表示されません。
- ・通話はできません。
- ・CRMアカウントはClientAppのみログイン可能で、SeatMap Appへのログインはできません。

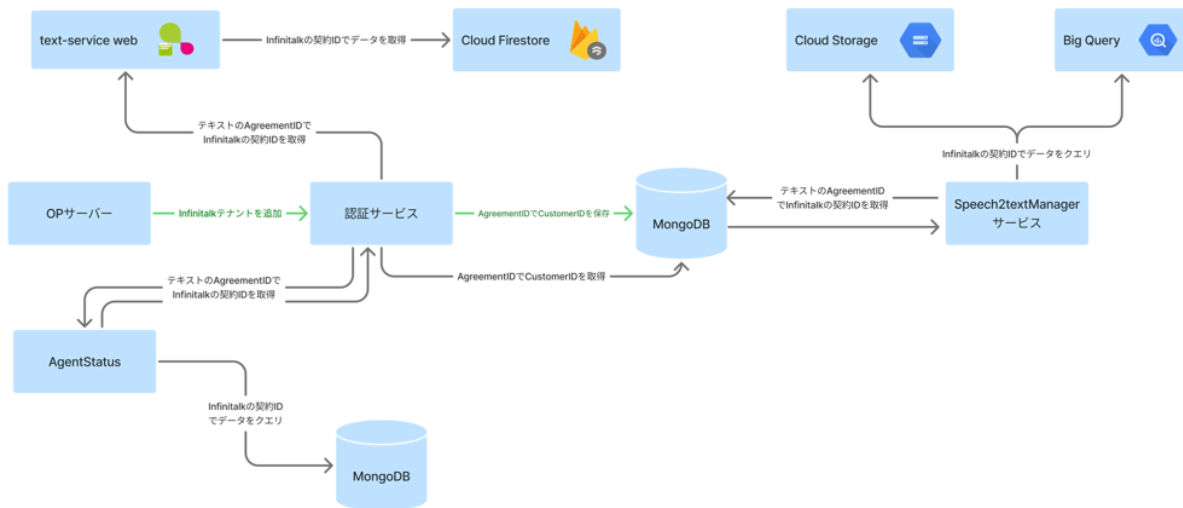
設定手順

- ・CRM単体の顧客を作成します。
- ・CRM単体の顧客を作成する際、Infinitalkの契約IDがないため、text-service側でAgreementIDを生成します。
- ・text-serviceのWebサイトに初めてログインすると、自動的にCompanyIDが生成されます。

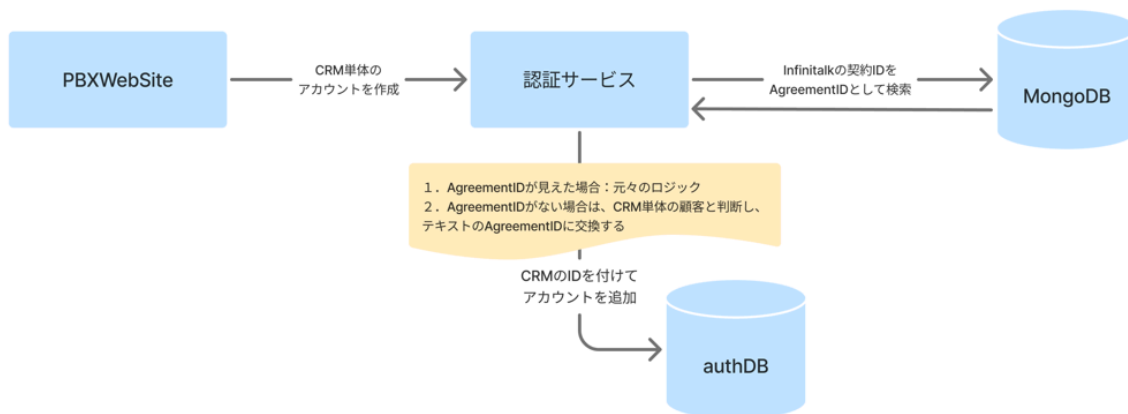


- ・CRM単体の顧客にInfinitalkを追加:

- 。Infinitalkテナントを追加する際に、Infinitalkの契約IDをCustomerIDとしてMongoDBに保存します。
- 。Infinitalkサーバに連携がある機能を利用する時に、MongoDBに見てテキストのAgreementIDでInfinitalkの契約IDを取得して、データをクエリします。-> Infinitalkテキストを追加した後、InfinitalkDBのデータを取得する必要がある場合は、まずテキストのAgreementIDでInfinitalkの契約IDを取ってから、正常にデータをクエリすることが出来るようになります。



CRM単体の顧客にInfinitalkを追加した後、PBXアカウントの対応



契約IDがない
AgreementIDはCRM....の形式である

クラウドサービス顧客一覧	請求期間	請求データの取り込み	アプリ/デバイス利用状況	顧客属性	契約状況	契約一覧	テキストサービス利用
346	GCP-owners-テナントA	基本プラン	065a	2021/04/21	NTTコム		
644	GreenCRM単体利用動作確認	基本プラン	400a	2024/02/20	CRM1708391749		
632	Green電話テスト用の顧客	基本プラン	400a	2023/12/12	NTTコム		
522	ママスクエアHWP(スカイネット)	基本プラン	085b	2022/07/27	IPS(SIPトラング)		
282	HRソリューションズ株式会社	基本プラン	054r	2020/09/06	プラスデル		

Infinitalkサーバを追加した後のCRMの顧客は..

クラウドサービス顧客一覧	請求期間	請求データの取り込み	アプリ/デバイス利用状況	顧客属性	契約状況	契約一覧	テキストサービス利用
346	GCP-owners-テナントA	基本プラン	065a	2021/04/21	NTTコム		
644	GreenCRM単体利用動作確認	基本プラン	400b	2024/02/20	CRM1708391749		
632	Green電話テスト用の顧客	基本プラン	400a	2023/12/12	NTTコム		
522	ママスクエアHWP(スカイネット)	基本プラン	085b	2022/07/27	IPS(SIPトラング)		
282	HRソリューションズ株式会社	基本プラン	054r	2020/09/06	プラスデル		

📌 手動でサーバーをCRMに紐付ける

理由：OP側ではデータがぐちゃぐちゃに作られており、Infinitalk側は頻繁にビルドし直すため、テナントが削除される。しかしOP側では削除されないため、残骸が残ってしまう。

- OP側で既に使用中のものを一時的に利用する
- Superadminにて、現在使用中のものと同じテナントIDの空テナントを作成する（例：954d）
- InfinitalkのDBに接続（pgAdmin使用）

tenants テーブルの **customer** カラムを更新

```
1 UPDATE tenants SET customer = 'xxxxxx' WHERE id = xxxx;
```

ast-configs テーブルに設定を追加

```
1 INSERT INTO ast_configs (cat_metric, var_metric, commented, filename, category, var_name, var_val)
2 VALUES
3 (0, 0, 0, 'extensions.conf', 'globals', 'Line_d', '30'),
4 (0, 0, 0, 'extensions.conf', 'globals', 'Phone_d', '30'),
5 (0, 0, 0, 'extensions.conf', 'globals', 'CC_d', '30'),
6 (0, 0, 0, 'extensions.conf', 'globals', 'MIS_d', '30'),
7 (0, 0, 0, 'extensions.conf', 'globals', 'IVR_d', '30'),
8 (0, 0, 0, 'extensions.conf', 'globals', 'LD_d', '30');
9
```

- Firebaseの設定を追加：Config → FireStore版AST（V2.1.0.0～）
- 新しく作成したテナントでInfinitalkに初回ログイン
- 電話番号を追加： **10.128.0.53/infinitalk/exten/add** にアクセスし、適当な電話番号を追加

2. VPN

VPN利用の場合はIPアドレスとなる為、IPアドレス指定でソフトフォンが利用

電話機アカウントの追加

所属グループ*
a_所属1

電話機タイプ*
 IP NETPHONE & SoftPhone
 IP NETPHONE & SoftPhone
 MIP-3010
 NAKAYO
 NTseries
 Sipdroid
 TLS通信InfiniTalkPhone
 TLS通信Yealink
 WebRTC (Audioのみ)
WebRTC (VPN接続)
 外線転送専用

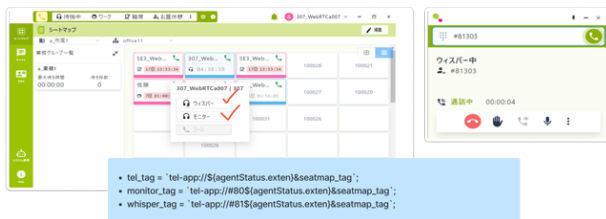
VPN利用の場合はIPアドレスとなる為、IPアドレス指定でソフトフォンが利用

- domain : Private IP
- port : 8088
- transport = ws

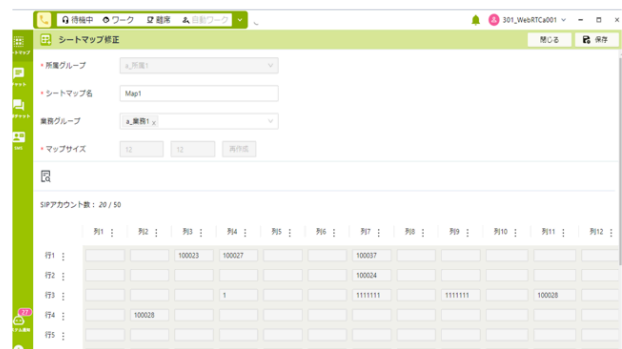
3. SeatMap, Threshold, Monitor, Chat Outside (Direct/General)

a. SeatMap, Threshold, Monitor, Whisper ..

- ウィスパー、モニターク: オペレータが通話中の時（権限：管理者）



- シートマップ編集は、SV（管理者）のみ編集可能
- シートマップのドラッグドロップは、SV（管理者）の動作可能

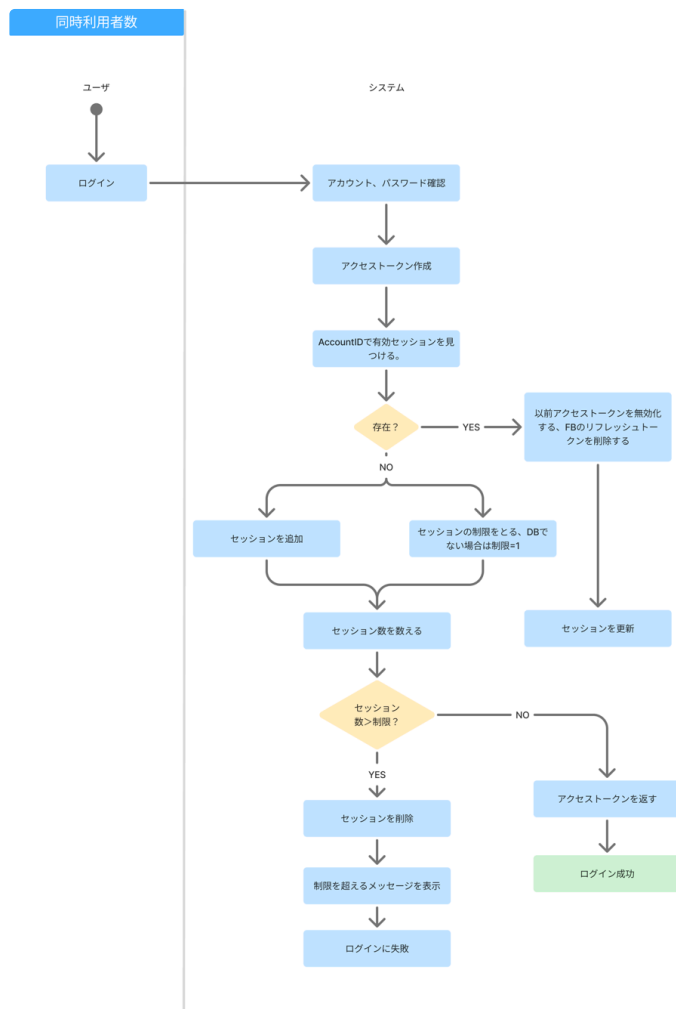
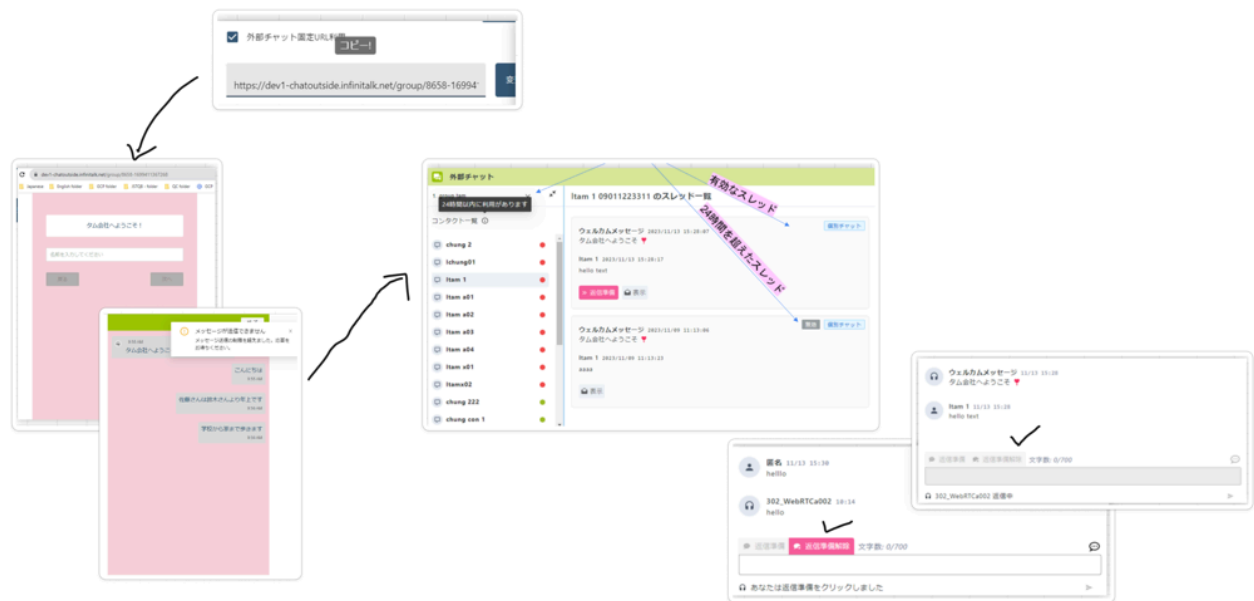


b. Chat Outside

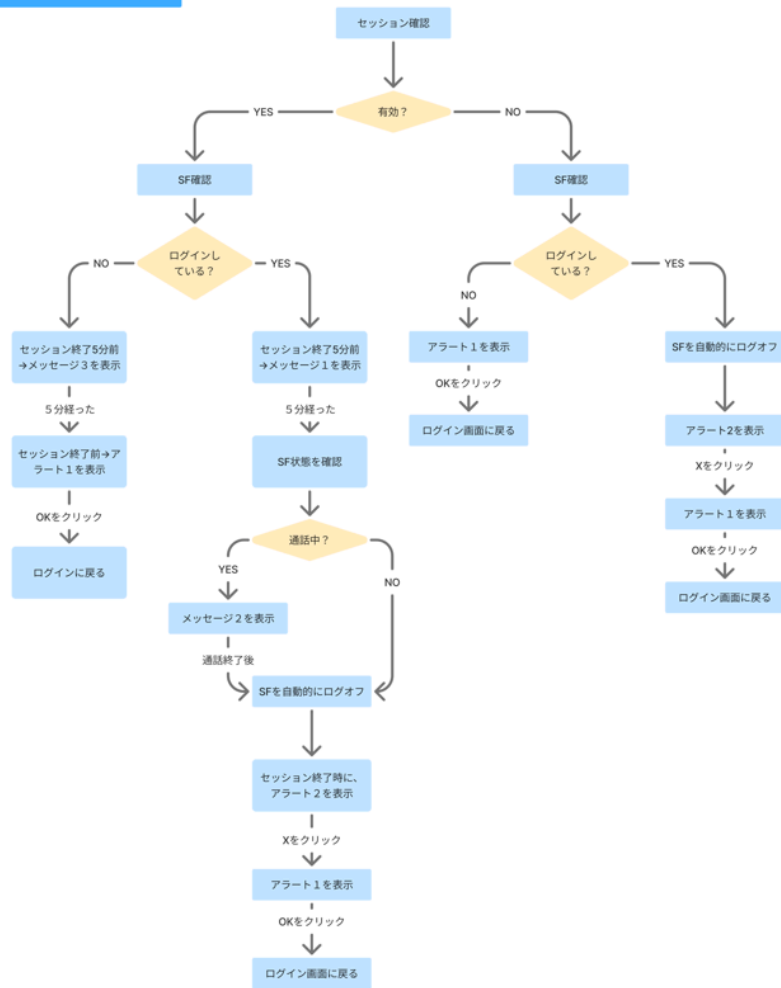
一時的なチャット用リンクを提供し、顧客とチャットができます。

2種類あります：固定URLと個別URL

- **固定URL**：特定のグループに対する固定のURLで、コンタクト登録の設定があります。顧客はこのURLにアクセスしてWebインターフェース上でメッセージを送信します。
- **個別URL**：オペレーターのSMS機能から顧客ごとに作成されます。顧客はこのURLにアクセスしてWebインターフェース上でメッセージを送信します。



セッション切れる対応



4. Call List, Call Announce, Auto Call, Auto Answer, Contact Notification, GPT, ..