



SunCloud P2Pソフト ウェア・テクノ ロジー・ホワイト ペーパー

Yichen Wu

2017/03/23

尚云互联

SunCloud P2Pソフトウェア・テクノロジー・ホワイトペーパー（中国語）

バージョン番号です。2.2.0

シンセン・シャンユン・インターネット・テクノロジー社

バージョン情報

最新版です。

-APIです。

-C++（Win32/Linux/Android/iOS/OSX）。2.2.4.0

-C（Linux）です。2.0.0.1

-P2Pサーバー。1.7.0

-Relay Server:.2.0.0



尚云互联

改訂履歴

バージョン	日付	著者	修正
1.0.0	2014/05/01	Charlie Chang	オリジナルバージョン
2.0.0	2016/12/28	Yichen Wu	1. 目次を追加し、コンテンツのアウトラインを削除します。 2. ドキュメントのフォントや段落のレイアウトを変更する。 3. フォワーディングサーバーのインストールとセットアップはサーバープログラムの章で、パラメーターはパラメーターの章で、RelayIPパラメーターの追加と表示イメージの変更を行います。 4. テストの結果を示す写真
2.1.0	2017/02/22	Yichen Wu	1. ConnectionTester、Changing platform testing toolsのスクリーンショットです。 2. P2Pサーバー部の削除—オプション機能章
2.2.0	2017/03/23	Yichen Wu	1. インストールとセットアップ」のフォワーディングサーバーの章を更新する。 start_Relay.shのコードです。
注：このドキュメントを変更するたびに、この表を更新してください。			

シンセン・シャンユン・インターネット・テクノロジー社は、本文書の内容を最終的に解釈し、変更する権利を有します。

カタログ

バージョン情報	1 -
改訂履歴	2 -
DIDパーツ	5 -
DID	
I. DIDコーディングルール	5 -
II. APILicense code.....	5 -
APILicense Code	
III. CRCキー	6 -
IV. デバイス・フラッシュ・メモリ内のパラメータ領域..	6 -
P2PServer部	8 -
-P2PServer	
I. 据付・架設	8 -
II. コンフィギュレーションファイルの設定	9 -
III. CRCキー.....	11 -
IV. ログ	12 -
V. 管理ツール	14 -
フォワーディングサーバー	17 -
I. フォワーディングサーバーの動作説明.....	17 -
II. 搬入・搬出.....	17 -
III. フォワーディングサーバー ID.....	20 -
IV. ログ	20 -
V. デバイスフォワーディング機能.....	21 -
プラットフォーム テストツール 手順	23 -

I. コネクションテスター.....	23 -
II.ListenTester.....	24 -

I. 動作原理.....	27	-
II. 搬入・搬出.....	27	-
リモートマネージメント部.....	30	-
- リモートマネージメント		
I. 動作原理.....	30	-
II. 搬入・搬出.....	31	-
III. セキュリティ.....	31	-
API パーツ	32	-
API		
I. 文字列の初期化.....	32	-
II. 聞く・つながる.....	32	-
III. 書き込みとフロー制御.....	33	-
IV. 読み取りと タイムアウト.....	33	-
Read		
V. ライブラリとバージョンリリース.....	33	-
6. 互換性.....	34	-

DID パーツ

I. DID コーディング ルール

- DIDのフォーマットは、XXXX-123456-ABCDE、です。
 - XXXX: プレフィックスコード（prefix code）、異なる製品、顧客、販売地域などを区別するために使用できる。
 - 異なるプリコードの転送サーバー（デバイス転送を含む）は共有できないため、各プリコードは独立して設定する必要があります。
 - プレフィックスコードが多すぎるとサーバーの効率が悪くなるため、プレフィックスコードの数は多すぎないようにしてください（1つ40を超えないことを推奨します）。プレフィックスコードは大文字で、数字は使用せず、文字「0」は使用しないことを推奨します。プレフィックスコードの最大値はcode7、最小値はcodeです。3

123456- : ランニングナンバー、6桁、000000~999999

- ABCDE: チェックコード

- チェックコードは、プレフィックスとランニングナンバーから独自のアルゴリズムで導き出されます。
 - チェックコードが正しいかどうかをサーバーが確認する
- 同等のDID入力: 以下のDIDは同等です。XXXX-123456-ABCDE、XXXX123456ABCDE、XXXX123456-ABCDEまたはXXXX-123456ABCDE

II. APILicense code

- APILicense Codeは、DID+プライベートアルゴリズムに基づいています

- P2PサーバーでDIDチェックコードを確認する
 - その目的は、DIDが法的に認可されていることを確認するためです。
- APILicenseはAPI内で検証されます。
 - その目的は、APIの正当な利用を保護することです。

■ APILicenseはListen, Listen(DID) のみ
必要です。....., APILicense)

■ クライアントがデバイスに接続するためにAPILicenseを必要としない

■ 戻り値 -
21ERROR_PPPP_INVALID_APILICENSE (-
21) APILicenseの検証が正しくない場合
。

III. CRCキー

- CRCキーは、P2Pプラットフォームの所有者の権利を保護するために設計されたセキュリティメカニズムです。
 - 正しいCRCKeyがないと、正しいDIDを持っていてもP2PServerにログインできません。
 - 本物のデバイスに影響を与える正しいDIDを持つダミーデバイスを避ける
 - True DID: そのDIDは確かに合法で正しいものです（ただし、不正な経路で取得された可能性があります）。
 - 偽装デバイス: 本物のデバイスメーカー（プラットフォームオーナー）によって偽装されていないデバイス
 - 純正品: 純正メーカーが正規のDIDで販売する機器
- お客様がP2Pサーバーに独自のCRCKeyを設定する場合
 - 3つのP2Pサーバーで同じ設定をする必要があります。
 - 長さは自由ですが、6～8文字を推奨します。長すぎるとパフォーマンスに影響します。
 - CRCKeyのことは、サンクラウド・インターネットやデバイスの設計者を含め、誰にも言わないようにしてください。
- CRCキーはListen()時に入力され、APILicenseと一緒に最後のパラメータに "AAAA:CCCCC" というフォーマットで格納されます。ここで、AAAAはAPILicense、CCCCCはCRCキーを

IV. デバイス・フラッシュ・メモリ内のパラメータ領域

- サンクラウドのP2Pソフトウェアプラットフォームを使用する機器には、以下のP2P関連の設定領域が必要となります。

- APILicense: 異なるDIDは異なるAPILicensesを持つ
- *CRCKey: CRCKeyはすべてのデバイスで同じです。
- *InitString: サーバーアドレスの暗号化された文字列、全てのデバイスに共通
- *注: 同一プラットフォームの全てのデバイスは、CRCKeyとInitStringが同じなので、プログラムで死守するように書くこともできます。
- 既存のパラメーター領域にDIDしかない場合は、XXXX-123456-ABCDE, AAAA:CCCCC
- DIDと "APILicense: CRCKey" を", "で区切って、Listen("DID")として呼び出します。 , "APILicense: CRCKey")

尚云互联

P2PServer部

I. 据付・架設

- P2Pサーバーは最大3台まで設置できますが、まずは3台すべてを設置することをお勧めします。
- P2P Server仕様の要件。
 - OS: 32または64ビットのLinux OS (Debian、Ubuntu、Redhat、Fedora... など、無用なトラブルを避けるためにUbuntuを推奨しており、本書の内容もUbuntuに基づいて書かれています。)
 - ハードウェア (CPU/メモリ/ストレージ): クラウドホスティングに必要な最低限の仕様で十分ですが、そうでない場合は後から拡張することができます。
 - 帯域幅の大きさが鍵: 1100万人のオンラインユーザーには約100Mbpsの帯域幅が必要
 - 固定されたパブリックIP (IPを移動させないようにするか、それが保証されない場合はドメイン名を使用する)
 - ファイアウォールを使用している場合は、UDPのブロックを解除してください。
 - P2Pサーバーは、バックアップ効果を得るために、異なるサーバールームや異なる地域に設置する必要があります。
- インストール手順
 - ホスティングまたはクラウドホスティング用のP2Pサーバーを3台用意 (クラウドホスティングを推奨)
 - P2P Server プログラムをディレクトリ (例: /root/PPC) に置く (** libmysqlclient.so.18 も同じディレクトリに置く必要があります)。
 - 設定ファイルの変更: 2PPサーバーには**Server.conf**、**RMPassword.conf**、**BlackList.conf**が設定されてい

ロジー・テクノロジー（中国語）
P2P Server 設定ファイルの設定」の項を参照
してください。

- Logディレクトリの作成（mkdir /root/PPC/Log）
- P2Pサーバープログラムが起動後に自動的に実行される
ように、"/root/PPC/PPCS_Server CRCKey &" コマンド
で設定します。

■ 付属の start_Server.sh を使用することをお勧めします（
内容は下記参照）。

■ root権限でのご利用をお勧めします。

■ P2Pサーバーの実行権にご注意ください。

■ CRCKeyはユーザー定義で、6～8文字のアルファベットまたは数字を推奨します。大文字小文字は区別されますので、「P2PServer CRCKey」の項を参照してください。

- start_Server.sh（赤字の部分は実際に使用するものによって異なります。）

```
#/bin/shcd /root/PPC
sl
while [ "$s" != "0" ] do
    if [ $(ps -A | grep -c 'PPCServer') = "0" ]; then
        /root/PPC/PPCServerCRCKey& fi
スリープ完了
5
```

- stop_Server.sh

```
killall satServer;killall PPCServer
```

- 起動時に自動的に実行される（変更される場合があります。詳細はP2Pデプロイメントのドキュメントを参照してください）。

```
#chmod 755 /root/PPC/*
#cd /root/PPC/satServer.sh>/etc/init.d/satPPC.sh#chmod 755/etc/init.d/satPPC.sh
#cd /etc3d/etc3d
#ln -s ../init.d/satPPC.shsatPPC.sh
```

II. コンフィギュレーションファイルの設定

- PPCのP2Pサーバーには、Server.conf、RMPassword.conf、BlackList.confという3つの設定ファイルがあります。

- P2Pサーバープログラムのディレクトリ（例：/root/PPC/）に置く必要があります。
- すべてプレーンテキスト形式ですので、vimを使って修正してください。
- 変更があった場合は、P2Pサーバーを再実行する必要があります

- セキュリティ上の理由から、/root/PPCはrootのみがアクセスできるように設定してください。

- Server.conf: サポートされるプレフィックスコードとランタイムナンバーの最大数の設定

- ファイルの内容が暗号化されたプレーンテキスト文字列であること
- 文字列が無効になるので、サンクラウドに連絡して文字列の内容を入手してください。
- Server.confは必要な設定ファイルで、これがないとP2PServerは実行できません。

- RMPassword.conf: リモートモニタリングのパスワードを設定する

- ファイルの内容はお客様にお任せしますが、少なくともコード8、大文字、小文字、数字の組み合わせを推奨します。
- リモート監視・管理を使用しない場合は、RMPassword.confの

- BlackList.conf: 特定のID番号でのp2p接続を無効にするために使用します。

- 一部のID番号→プレフィックスとランニングナンバーで指定
- 無効にするID番号がない場合は、BlackList.confを使用していません。

- Server.conf（内容は以下の例とは異なるはずです。）

```
root@Debian:~/PPC#  
root@Debian:~/PPC# cat Server.conf  
CALCLCNJBBDILPPEFmcJGF1FdFGEsdfeDFsfDGLofGgOCDcGaFiIeKoAcMgFMFoNgI  
root@Debian:~/PPC#
```

- RMPassword.（注意：あなたのコンテンツは以下の例とは異なるはずです）

```
root@Debian:~/PPC#  
root@Debian:~/PPC# cat RMPasssword.conf  
A1b2c3D4  
root@Debian:~/PPC#
```

- BlackList.conf（内容は以下の例とは異なるはずです。）

```
root@Debian:~/PPC#  
root@Debian:~/PPC# cat BlackList.conf  
[GENERAL]  
ListNo=2  
List_0=AABB  
List_1=AACC  
[AABB]  
No=3  
SN_0=000099  
SN_1=000098  
SN_2=000097  
[AACC]  
No=1  
SN_0=000123
```

III. CRCキー

- CRCKeyについて。
 - CRCキーは、サンクラウドのP2Pソフトウェアのユーザーが、偽のデバイス（ただし正しいDIDを持つ）がP2Pサーバーにログインし、本物のデバイスが正常に動作しなくなることを防ぐために設計されています（これを偽デバイス攻撃と呼びます）。
 - 正しいCRCKeyを知らずにサーバーにログインすることはできません。
 - CRCKeyは、CRC Checksumとしてだけでなく、デバイスの登録パケットの内容に直接影響を与えるので、同じDIDでもCRCKeyが違えば、登録パケットの内容は全く違ったものになります。
- CRCキーの設定。
 - この設定は、P2Pサーバーの実行時に実行パラメータとして与えられます。
 - 例：以下の実行コマンドのCRCKeyは “Ab09cD12 ”です。
 - **# /root/PPC/PPC_Server Ab09cD12 &**
 - CRCKeyは大文字と小文字を区別します。

ロジニ・CRCキーの長さ(中国語)に制限はありませんが、長すぎるキーは 2.2.0

パフォーマンスに影響するため、6～8コードを推奨します。

- 機器側のCRCKeyの設定については、DIDの「DID-CRC Key」の項目をご参照ください。

- CRCKeyは、偽装デバイスへの攻撃に対する防御として使用され、デバイスがリリースされると変更できないため、絶対に秘密にしておき、漏洩しないようにしなければなりません。

IV. ログ

- P2PServerは、Server.inf、DataFlow.log、DDHH.logなど、プレーンテキストの内容を持つ複数のログファイルを生成します。
 - Server.inf: P2PServerの実行パラメータを格納
 - ディレクトリの場所: /dev/shm
 - これには、サーバー名、バージョン番号、デバッグレベル、許容される転送サーバーの最大数（個々のプレフィックスについて）、CRCKey、サポートされているプレフィックスおよびプレフィックス数などが含まれます。
 - 情報漏洩を防ぐために、このファイルはrootのみがアクセスできるようにしてください。
 - DataFlow.log: 1時間ごとのデータフローの記録
 - ディレクトリの場所: /root/PPC（**実際のユーザーの使用状況によって異なります。**）
 - 前の時間の値を1時間ごとにオンタイムで記録する In & Out number of Byte.
 - DDHH.log: デバイスの登録、クライアントのオンラインリクエスト、転送サーバーのログイン…。パケットのリアルタイムロギングなど
 - ディレクトリの場所: /root/PPC/Log（**/root/PPCは実際のユーザーの使用状況に応じて変化しますが、Logディレクトリ名は固定です。**）
 - DDHH → DD: 00~31（各月の日）HH:00~23（時間）、すなわち、1時間ごとに1ファイル

ロジエンのウェブサイト（中国語）のログ（プレーンテキスト形式）
■ 上記のログ（中国語）のログ（プレーンテキスト形式）
バージョン番号は2.2.0

、ユーザーが見やすいようにするため）に加えて、P2Pサーバー自身が使用することを目的としたLoginIDs.datというログもあります。

- LoginIDs.dat: すべてのデバイスで最後に成功したログイン情報を保存する

■ ディレクトリの場所: /root/PPC（**実際のユーザーの使用状況によって異なります。**）

- このファイルは自動生成されたものですので、偶発的なエラーを避けるため、ご自身で修正しないでください。
- このファイルを削除してもP2PServerの動作には影響しませんが、すべてのデバイスの最終ログイン時間がわからなくなる可能性があります。

• DDHH.logの内容の説明。

– A. デバイスのログインログです。

■ エラーです。

◆ ログイン [XXXX]’ DID ’from IP: Port

Invalid Prefix not spotted by this server XXXX=Invalid DID このIDはチェックサムのため、無効です。は正しくありません。
Banned DID このDIDはブラックリストに入っています。
Login Out of Date DIDライセンスは期限切れです。

■ 成功しました。

◆ IP: Port, NATType=#, APIVersion: #から
ログイン「DID」。#. #, LocalAddr=IP:
Port

NATType: 1: Port Restricted NAT, 2: Port Restricted NAT, 3: Symmetric NAT
API Version APIのバージョン、###
Local Address ログインデバイスのローカルアドレス

– B. デバイスリレー機能

■ エラーです。

◆ SDev ログイン [XXXX]’ DID ’from IP:Port

XXX= A. デバイスのログインログを参照してください。

■ 成功しました。

◆ IP: ポート’からのSDevログイン’ DID

– C. P2Pリクエストログ

■ エラーです。

◆ P2PReq [XXXX]’ DID ’from IP: Port

XXX= A. デバイスのログインログを参照してください。

◆ P2PReq' DID 'from IP: Port, LocalAddr= IP: Port

- D. リレーサーバーのログイン

■ エラーです。

◆ RSLogin ログイン [XXXX] 'SID' from IP: Port

XXX=A. 機器ログインログを参照SD. リレーサーバーのID

■ 成功しました。

◆ RSLogin Login 'SID' from IP: Port,
BandWidth=##, UserNumber=##"

BandWidthは まだ実装されていませんので、無視してくださいね。 ユーザー番号はまだ実装されていませんので、Ignore!!

- E. リレーリクエスト

■ エラーです。

◆ RLYReq [XXXX]' DID 'from IP: Port

XXX= A. デバイスのログインログを参照してください。

■ 成功しました。

◆ IP:Port' からのRLYReq' DID、RLYAddr=IP:Port

RYAddr: リレーサービスプロバイダーのIP: ポート

- F. リレーサーバーリストリクエストログ

■ エラーです。

◆ ListReq1 [XXXX]' DID 'from IP: Port

XXX= A. デバイスのログインログを参照してください。

■ 成功しました。

◆ 成功したときのログはありません。

V. 管理ツール

- P2PServerには2種類の管理ツールがあります。

- PPC_Management: ローカルマシンでの管理機能を提供する

使用方

- 必要であれば、P2P ServerのホストマシンにWebServerをインストールし、CGIを使ってPPC_Managementプログラムを実行し、以下のように結果を出力することもできます。/Log/DID_Dump.htmlをWeb経由で出力することができるので、Webを使ってリモートで問い合わせをすることができます。

- RemoteCommander、RemoteMonitor: リモートマネージメント

- リモートコマンダー

- 使用方法

使用方法: `./RemoteCommander HostName Password`

P2Pサーバーの動作状態を取得する。

使用方法: `./RemoteCommander HostName Password 2`

ログイン機器の統計データを取得する。

使用方法: `./RemoteCommander HostName Password Prefix SN`

SNのサーティファイドデバイスのログイン状態をプレフィックス付きで取得する 使

方: `./RemoteCommander HostName Password 4`

サーバーへのEXITコマンド送信

使用方法: `./RemoteCommander HostName Password 5`

サーバーからのServer ofの取得

使用方法: `./RemoteCommander HostName Password 6 Server Of File Name`

Server ofをServerに更新

- パスワードは、「P2Pサーバー - 設定ファイルの設定」の項で説明したように、RMPassword.confで設定します。
- ユーザーが複数のP2Pサーバーを持っている場合は、ヒューマンエラーを避けるために、RemoteCommanderで制御・管理することをお勧めします。

- RemoteMonitorです。

- これは、RemoteCommanderからコントロール機能を取り除くバージョンで、管理者権限を持っていないが、リモートサーバーで何が起きているかを知りたい人のための

チャージャー

忘れられた？

■ 使用方法

使用方法。

`./root/.SunCloud/Server/HostName/Passwd/GetP2P`サーバー

の動作情報です。

使用方法: `./root/.SunCloud/Server/HostName/Passwd/GetP2P` 取得方法

SunCloud P2Pソフトウェア・テクノ
ロジー・ホワイトペーパー（中国語）
ログインデバイスの統計データ

バージョン番
号です。2.2.0

使用方法: AddHostNamePasswd RefixSN
接続付きのSNのcertainvicのログイン状態を取得する。



尚云互联

フォワーディングサーバー

I. フォワーディングサーバーの動作説明

- Relay Serverは、P2P Serverがオンラインになると積極的に報告するので、固定のインターネットIPは必要なく、いつでも比較的安価なトラフィックを持つクラウドホストに切り替えることができます。
- フォワーディングサーバーはUDPポートを使用してフォワーディングサービスを提供しているため、フォワーディングサーバーホスト上のファイアウォールで双方向のすべてのUDPポートのブロックを無効にしなければ、サービスを提供できません（事業者の外部ファイアウォールには特に注意してください）。
- フォワーディングサーバーは、実際にはLinux上の実行プログラムに過ぎません。
 - 1台のホストで複数のフォワーダーを動作させることができ、これは複数のフォワーディングサーバーを持っているのと同じです（もちろん、総トラフィックは制限されます）。
 - 各転送サービスプログラムは、1つのDIDプレフィックスのみをサポートし、異なるDIDプレフィックスを持つ転送を相互にサポートすることはできません。例えば、AABBプレフィックスの転送サーバーは、AACCプレフィックスのデバイスに対して転送サービスを提供することはできません。フォワーディングサーバーID」の項
 - フォワーディングサーバーは必要に応じて追加 / 削除 / 削除が可能で、理論上のサーバー数に制限はなく、追加 / 削除 / 削除は即時（数分以内）に行われます。1
 - Add: クラウドホスト上で新しいRelay Serverプログラムを実行する
 - Delete: クラウドホスト上のRelay Serverプログラムの実行を終了させる
 - 移動: 旧クラウド・ホストで終了し、新クラウド・ホスト（同一

II. 搬入・搬出

- インストール手順
 - 3264ビットx86 Linuxの物理ホストまたはクラウドホストの準備、不要なトラブルを避けるためにUbuntuを推奨、本マニュアルの内容はUbuntuをベースにしています)

- PPCRelay Serverプログラムを任意の場所（例：`:/root/PPC`）に配置します。

- Logディレクトリを作成します。

```
#mkdir /root/PPC/RSLog
```

- リレープログラムの実行

- 付属のstart_Relay.shを使用することをお勧めします。
- root権限でのご利用をお勧めします。
- リレープログラムの許可について

```
root@ubuntu:PPC# ./PPCS_Relay
Usage: ./PPCS_Relay SID SID Port BandWidth MaxUser P2PServer1 P2PServer2 P2PServer3 [DelayTime_ms] RelayIP
SID: The SID of this Relay Server
Port: The Relay Service Port
BandWidth: Bandwidth of this relay server, in unit of Kbps, if unknow type 0
MaxUser: Max number of relay service user, if unlimit type 0
P2PServer1: Domain name of 1st P2P Server 0
P2PServer2: Domain name of 2nd P2P Server 0
P2PServer3: Domain name of 3rd P2P Server 0
DelayTime_ms: The response delay time in ms. Delay time is to fine ture server' response
The smaller the quicker server response a relay request, thus, increase bandwidth usage
Default value: 0 (if not specified), suggest value : 200, max value: 1000
Example: ./PPCS_Relay RLY-0000-XXXX 12300 10000 50 127.0.0.1 127.0.0.1 127.0.0.1 200 127.0.0.1
```

• プログラムパラメータの説明

- Usage: PPCS_Relay SID Port BandWidth MaxUser P2PServer1 P2PServer2 P2PServer3 [DelayTime_ms] [RelayIP].

- SID: このリレーサーバーのSID（詳細は「[フォワーディングサーバー](#)」参照）
- 転送先サーバーID」の章)
- ポート: リレーサービスのポート
- BandWidth: このリレーサーバーの帯域幅、単位はKbps、unknowタイプの場合は0（現在は予約済みのパラメータで、番組の機能としては実用的ではありません。）
- MaxUser: unlimitタイプが0の場合のリレーサービスユーザーの最大数（リレーサービスの最大数を制限するために使用します）
- P2PServer1: 1台目のP2PサーバーのIPまたはドメイン名

- P2PServer2: 2台目のP2PサーバーのIPまたはドメイン名
- P2PServer3: 第3のP2PサーバーのIPまたはドメイン名
- DelayTime_ms: 応答の遅延時間（単位: ms）。

小さければ小さいほど、中継要求に対するサーバーの応答が早くなり、その結果、帯域幅の使用量が増加します。初期値：0（指定しない場合）、推奨値：最大200, 値1000

- RelayIP: Local IP（このパラメーターを設定すると、P2PサーバーはまずフォワーディングサーバーのIPを取得します。したがって、P2Pサーバーとフォワーディングサーバーが同じクラウドホストまたは同じサーバールームにある場合には設定する必要があります。）

- 例: ./PPCS_Relay RLY-0000-XXXXX 12300 10000 50
s1.p2p.com s2.p2p.com s3.p2p.com 200140.112.123.45
- MaxUserは、フォワーディングサービスの最大数を制限するためのものです。
 - に設定されている場合、最大1001001つの同時転送サービスのみが提供されることを意味します。
 - このパラメータは、最大帯域幅の使用量を制限するために使用できます。特に、最大許容帯域幅が設定されている特定の共有クラウドホストでは便利です（帯域幅の使用過多により「無制限」が無効になることが多い場合）。
- Delay_Timeは、サービスリクエストを転送する際の応答時間を調整するためのもので、デバイスの転送を優先させたり、複数のサーバー間で負荷を分散させたりするために使用されます
- start_Relay.sh（以下のコードは、異なるプレフィックスを持つサーバーで実行されます）

```
#!/bin/sh#!/bin/shcd
/ct/PCs1
while[ "$$" != "0" ] do
    if [ $(selfip"ge" |grep-c 'ACE(0000)')="1" ]; then
/ct/PPCRelayACE(0000)ADE 100100100
2SaveP12Save122Save13RelayP200 fi
    if [ $(selfip"ge" |grep-c 'BC(0000)')="1" ]; then
/ct/PPCRelayBC(0000)ADA 100100100
2SaveP12Save122Save13RelayP200 fi
    if [ $(selfip"ge" |grep-c 'DA(0000)')="1" ];
```

```
/root/.PCHRelay/DMA000000EB 1200500500
```

P2Sev1P2Sev2P2Sev3RelayP20& fi

スリープ完

了5

- ```
killall startRelay.sh killall HCRelay
```

### III. フォワーディングサーバーID

- 配信するDIDプレフィックスを指定する=SIDのプレフィックスを指定する

- サーバーが正規のものであることを確認する。SIDはチェックコード付きのDIDに似ており、SIDの構造: XXXX-123456-ABCDE

- XXXX :プレフィックスコード

- ## ■ 123456ストリーマー

- ## ■ ABCDE: チェックコード

- なぜフォーディングサーバーIDにもチェックコードが必要なのか?

- フォワーディングサーバーがチェックデジットを持っていない場合、Fake Relay Server攻撃を受ける可能性があります。

- フォワーディングサーバーのSIDについては、サンククラウドにお問い合わせください。

#### IV. ログ

- DDHH. logなどのプレーンテキストのログファイルが生成



- SIDの違いを区別するために、Relay.inf、DataFlow.log  
の後にSIDを記載しています。  
"Relay.inf\_XXXX-123456-ABCDE", "DataFlow.log\_XXXX-1

- Relay.inf: Relay Serverの実行設定の内容が格納されている
  - ディレクトリの場所: /dev/shm
  - その内容は、バージョン番号、デバッグレベル、実行パラメータなどです。
  - 情報漏洩を防ぐため、このファイルはrootのみが読めるように設定してください。
- DataFlow.log: 1時間ごとのデータフローの記録
  - ディレクトリの場所: /root/PPC ( **実際のユーザーの使用状況によって異なります。** )
  - 前の時間の値の1時間ごとのオンタイムの記録 In & Out Byte.
- DDHH.log: フォワーディングサービスの使用ログ、エラーログなど。
  - DDHH.logです。
    - ディレクトリの場所: /root/PPC/RSLog ( **/root/PPCは実際のユーザーの使用状況に応じて変化し、RSLogのディレクトリ名は固定されます。** )
    - DDHH->DD: 00~31 (各月の日付) HH:00~23 (時間)、つまり1時間に1ファイル
- ログが直接読める

## V. デバイスフォワーディング機能

- ネットワークの状態が許す限り、デバイスがデバイス転送機能 (APIfunction-> Share\_Bandwidth(1)) を有効にしていれば、そのデバイスは転送サービスを提供できるデバイス (スーパーデバイス) としてP2Pサーバーに登録されます。
- デバイスフォワーディング機能の説明
  - 各スーパーデバイスは、1つのクライアントに対してのみフォワーディングサービスを提供できます。
  - スーパーデバイスはいつでもフォワーディングサービスを中断することができる (Share\_Bandwidth(0))

ロジック・プラットフォーム（中国語）  
サービスは、P2P APIではなく、<sup>2.2.0</sup>

デバイスのファームウェアによって制御されます（APIは機能を提供するだけで、有効にするかどうか、無効にするかどうかはデバイスのプログラム次第です）。

- 推奨される使用方法：クライアントがオンラインになっていないときは（Share\_Bandwidth(1)）をオンにし、クライアントがオンラインになったらオフにします。

- 一部の地域では、転送機能がユーザーの承認を得ていないと違法となる場合があります
- 端末の転送機能は、プライバシーの問題はないのか？
  - 理論的にはそうではありません。SunCloud P2P APIはサイドロギングを提供していないため、ユーザーはAPIを通じてデバイスが転送したパケット内容にアクセスすることはできず、ロギングも行われません。
  - 製品のプロトコルがすでに暗号化されている場合、デバイスの転送によってプライバシーが侵害される可能性は低くなる



## プラットフォーム テスト ツール プログラム

### I. コネクションテスター

- Linuxベースのスタンドアローンコマンドラインプログラム
  - Linuxベース：動作にはx8632ビットのLinux OSが必要です。
  - スタンドアローン：P2P APIライブラリ（すでに含まれている）なしで独立して実行可能
  - コマンドライン：コマンドラインモードでの操作
- 用途
  - DID機器が接続可能かどうかを判断するために使用します。
    - エクストラネット接続またはイントラネット接続
  - また、デバイスが利用できない場合に、そのDIDが正規のDIDであるかどうかをテストするためにも使用できます。
  - フォワーディングサーバーが動作しているかどうかのテスト
  - 長期間にわたる機器のオンライン安定性のテスト
- 使用方法

```
dugu@ubuntu:/mnt/hgfs/Workstation/P2P文件/Tester$./ConnectionTester
ConnectionTester Verion: 2.4.0 Build 16/11/28 17:17
API Version: 2.2.4.0
Usage: ./ConnectionTester Mode DID InitString [Repeat]
Mode: 0, No local LAN search, P2P then Relay for remote
 1, Local LAN search, P2P then Relay for remote
 2, No local LAN search, force Relay for remote
 3, Local LAN search, force Relay for remote
 4, Do Network Detect Only
 6, No local LAN search, force Server Relay for remote
Repeat: Repeat times.(eg: 100), If not speified, connect once.
```

- 例
  - DIDのエクストラネット接続テストを行う

```
dugu@ubuntu:Tester$./ConnectionTester 0 PPCS-014103-WBPKR EBGAEIBIKHJJGFJKEOGCF
AEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
DID=PPCS-014103-WBPKR, RemoteAddr=61.141.152.217:62302, Mode=P2P, Time=0.662 (Sec)
```

- イントラネット上のDIDのオンラインテスト

```
dugu@ubuntu:Tester$./ConnectionTester 1 PPCS-014103-WBPKR EBGAEIBIKHJJGFJKEOGCF
AEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
DID=PPCS-014103-WBPKR, RemoteAddr=192.168.30.128:25590, Mode=P2P, Time=1.518 (Sec)
```

ロジニ・デバイスが存在しない場合でも、ConnectionTesterでテストす  
ることができます。

```
dugu@ubuntu:Tester$./ConnectionTester 0 PPCS-014103-WBAKR EBGAEIBIKHJJGFJKEOGCF
AEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
Connect failed : Invalid Device ID
dugu@ubuntu:Tester$./ConnectionTester 0 PPC-014103-WBPKR EBGAEIBIKHJJGFJKEOGCFA
EHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
Connect failed : Prefix of Device ID is not accepted by Server
dugu@ubuntu:Tester$./ConnectionTester 0 PPCS-014103-WBPKR EBGAEIBIKHJJGFJKEOGCF
AEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
Connect failed : Device is not on line now, nor login in the past 5 minutes
dugu@ubuntu:Tester$
```

错误的DID      前置码不支持      表示DID合法

- 長時間のテストによる機器のオンライン安定性

```
dugu@ubuntu:Tester$./ConnectionTester 0 PPCS-014103-WBPKR EBGAEIBIKHJJGFJKEOGCF
AEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM 100
```

- ConnectionTesterを使用して、フォーワーディングサーバーが動作しているかどうかをテストする

```
dugu@ubuntu:Tester$./ConnectionTester 6 PPCS-014103-WBPKR EBGAEIBIKHJJGFJKEOGCF
AEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
DID=PPCS-014103-WBPKR, RemoteAddr=:10019, Mode=RLY, Time=2.426 (Sec)
dugu@ubuntu:Tester$./ConnectionTester 6 PPCS-014103-WBPKR EBGAEIBIKHJJGFJKEOGCF
AEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
Connect failed : Connection time out, probably the device is off line now
dugu@ubuntu:Tester$ clear
dugu@ubuntu:Tester$
```

转发服务器正常      转发服务器异常

## II. ListenTester

- Linuxベースのスタンドアローンコマンドラインプログラム
  - Linuxベース：動作にはx8632ビットのLinux OSが必要です。
    - スタンドアローン：P2P APIライブラリ（すでにラップされている）なしで独立して実行可能
    - コマンドライン：コマンドラインモードでの操作
    - 用途
      - ◆ ネットワーク環境の検出に使用
      - ◆ DIDがP2Pサーバーに正常にログインできるかどうかをテストするために使用します。



- ◆ ConnectionTesterを使用して、APILicenseが正しいかどうかを確認します。

- 使用方法

```
dugu@ubuntu:Tester$./ListenTester_PPCS
ListenerTester Verion: 2.0.0 Build 16/12/04 12:22
API Version: 2.2.4.0
Usage: ./ListenTester_PPCS DID APILicense InitString
```

- 注意: ListenTesterは数**60**秒間待機し、クライアントが接続されていない場合はタイムアウトが表示されます。

- 例

- ネットワーク環境の検出に使用

```
dugu@ubuntu:Tester$./ListenTester_PPCS PPCS-014103-WBPKR HKGUCG:EasyView EBGAE
IBIKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
----- NetInfo: -----
Internet Reachable : YES
P2P Server IP resolved : YES
P2P Server Hello Ack : YES
Local NAT Type : Port-Restricted Cone
My Wan IP :
My Lan IP : 192.168.30.128

Start Listen('PPCS-014103-WBPKR', 60, 0, 1, 'HKGUCG:EasyView')...
```

- DIDがP2Pサーバーに正常にログインできるかどうかをテストするために使用します。

```
dugu@ubuntu:Tester$./ListenTester_PPCS PPCS-014103-WBPKR HKGUCG:EasyView EBGAE
IBIKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
----- NetInfo: -----
Internet Reachable : YES
P2P Server IP resolved : YES
P2P Server Hello Ack : YES
Local NAT Type : Port-Restricted Cone
My Wan IP :
My Lan IP : 192.168.30.128

Start Listen('PPCS-014103-WBPKR', 60, 0, 1, 'HKGUCG:EasyView')...
Got Server Response...
Listen failed (-3): Listen time out, No client connect me !!
dugu@ubuntu:Tester$./ListenTester_PPCS PPCS-014103-WBPKR HKGUCG:EasyView EBGAE
IBIKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
----- NetInfo: -----
Internet Reachable : YES
P2P Server IP resolved : YES
P2P Server Hello Ack : YES
Local NAT Type : Port-Restricted Cone
My Wan IP :
My Lan IP : 192.168.30.128

Start Listen('PPCS-014103-WBPKR', 60, 0, 1, 'HKGUCG:EasyView')...
No Server Response!!!
```

登录成功

没登录成功, CRCKey错误?



- ConnectionTesterでは、APILicenseが正しいかどうかを確認するために

```
dugu@ubuntu:Tester$./ListenTester_PPCS PPCS-014103-WBPKR HKGUCG:EasyView EBGAEIBIKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
----- NetInfo: -----
Internet Reachable : YES
P2P Server IP resolved : YES
P2P Server Hello Ack : YES
Local NAT Type : Port-Restricted Cone
My Wan IP : 61.141.152.217
My Lan IP : 192.168.30.128

Start Listen('PPCS-014103-WBPKR', 60, 0, 1, 'HKGUCG:EasyView')...
Got Server Response...
DID : PPCS-014103-WBPKR
Remote Addr : 192.168.30.128:19383
Connection mode: P2P
```

```
dugu@ubuntu:Tester$./ConnectionTester 1 PPCS-014103-WBPKR EBGAEIBIKHJJGFJKEOGCFAEPHPMAHONDGJFPBKCPAJJMLFKBDBAGCJPBGOLKIKLKAJMJKFDOOFMOBECEJIMM
DID=PPCS-014103-WBPKR, RemoteAddr=192.168.30.128:15641, Mode=P2P, Time=0.214 (Sec)
```

接続が成功したということは、APIlicenseが正しいということです。

# ウォッチドッグ

## I. 動作原理

- Watch Dogは、P2Pサーバーやフォーディングサーバーなどのサーバーが正常に動作しているかどうかを監視するために使用されます。
  - 一定の間隔（約151秒）で、WatchDogは指定されたサーバーにパケットを送信する
  - 受け取ったサーバーはACKパケットで応答する
  - 連続して5分間ACKを受信しない場合、サーバーに例外が発生したと判断する（Server Down）
- WatchDogは、Up / Downイベントをすべてのサーバーに通知する手紙を送ります。
- WatchDog自体の故障に備えて、複数のWatchDogを設定可能
- WatchDogメインフレームの仕様要件
  - 32 安定したインターネット接続が可能なBit x86 Linuxサーバ
  - ウェブサーバ
  - UDPファイアウォールなし

## II. 搬入・搬出

- Step 1:クラウドホストの準備（以下のインストール手順は全てamazonEC2, t1.micro, AMI Linux 32bitの場合で、手順に従うことをお勧めします）、Webサーバのインストール、ポートの80オープン設定を行います。
  - `yum install httpd`
- Step2 : WatchDog プログラムを /root/WatchDog/ の下に配置します。
- Step 3: WatchDogのコンフィグレーション・ファイルであるMailList.cfgとWDT.cfgの内容を変更します。

ロジック・ファイル（中国語）  
MailList.cfg: サーバアップ/ダウンのメール通知メールボックス  
の設定

- WDT.cfg: 監視するサーバーIP、ポートの設定
- ステップ4: WatchDogプログラムを実行し、起動時に自動的に実行する  
ように設定します。

- root権限でご利用ください。
- WatchDogはテキストファイルを生成します。  
/var/www/cgi-bin/Wdt\_Status.logの場合。  
/var/www/cgi-bin/ディレクトリが存在しません、手動でmkdirしてください。  
/var/www/cgi-bin/
- Step5 : サーバーの状態を監視するためのServerMonitor.cgiの修正と設定
  - ServerMonitor.cgiの実行権にご注意ください。
  - を有効にするようにServerMonitor.cgiの内容を変更してください。  
/var/www/cgi-bin/Wdt\_Status.logがブラウザに出力されます。
  - var/www/cgi-bin/Wdt\_Status.logを誰でも読めるように設定してください。

■ Chmod /var/www/cgi-bin/Wdt\_Status.log644

- コンフィグファイルです。

- MailList.cfgの内容

```
/root# cat MailList.cfg
123.com
123.com
123.com
```

- WDT.cfgコンテンツ（フォーマット：IP:Port,Descriptions）

```
/root# cat WDT.cfg
123.com:12345 Server1
123.com:12345 サーバー2
123.com:12345 サーバー3
123.com:12345 :1001、PCリレーサーバー1 [PC]
123.com:12345 :1001、PCリレーサーバー2 [PC]
```

- ServerMonitor.cgiです。

- ServerMonitor.cgiは、WatchDogから監視しているサーバーの状態をいつでも確認できるように設計されています。

■ [http://your.watchdog.ip\\_or\\_domain/cgi-bin/ServerMonitor.cgi](http://your.watchdog.ip_or_domain/cgi-bin/ServerMonitor.cgi)

SunCloud P2Pソフトウェア・テクノロジー・ホワイトペーパー (中国語)  
ServerMonitor.cgiの内容です。  
#linah

バージョン番号  
です。2.2.0

SunCloud P2Pソフトウェア・テクノロジー・ホワイトペーパー (中国語)

バージョン番号です。2.2.0

```

echo 'html'
Echo 'htmlpair="olett"olett="ex/html,cast#8"'
echo 'htmlpair="efish"olett="0,ufegbinSvnmio.org"'
echo 'htmlpair="efish"olett="0,ufegbinSvnmio.org"'
echo 'title=PCサービの監視title=led echo 'htmlサービの稼動状況html。'
echo 'pre'
cat WLSauslg
echo '<pre>html</pre>' exit 0

```

- なお、ServerMonitor.cgiの実行権限、ディレクトリの場所、および  
/var/www/cgi-bin/Wdt\_Status.logへの読み取りアクセス。

# リモートマネジメン ト部

## I. 動作原理

- リモート管理とは、P2Pサーバーの操作を遠隔地のホストから制御または表示する機能です。
  - リモート管理プログラム：RemoteCommander & RemoteMonitor
    - RemoteCommander：コントロールとビューイングの両機能
    - RemoteMonitor：表示のみ
- リモートマネジメントを実行するために必要なパスワード
  - P2Pサーバー - 設定ファイル」のRMPassword.confを参照してください。
- 使用方法
  - リモートコマンダー  
使用方法：./RemoteCommander Server Host Name Password  
Get P2P Server running information  
使用方法：./RemoteCommander Server Host Name Password --> ログイン機器の統計データを取得する。  
使用方法：。  
/ RemoteCommander Server Host Name 3 Password Prefix SN --> SNの特定のデバイスのログイン状態をプレフィックス付きで取得する場合 使い方：./RemoteCommander Server Host Name Password SendXTCommand to Server  
使用方法：./RemoteCommander Server Host Name Password5 --> Get Server.conf from Server  
使用方法：./RemoteCommander Server Host Name Password6 Server Conf File Name --> Update Server.conf

### - リモートモニター

- 使用方法：./RemoteMonitor Server Host Name1 Password --> P2Pサーバー稼働情報を取得する。
- 使用方法：./RemoteMonitor Server Host Name2 Password --> ログイン機器の統計データを取得する。
- 使用方法：。  
./RemoteMonitor Server Host Name Password Prefix SN --> プレフィックスcを持つSNの特定のデバイスのログイン状態を取得するには

## II. 搬入・搬出

- 要件：リモートマネジメントを導入するホストは、以下の仕様を備えている必要があります。
  - OS:x86 Linux , 32ビット, インターネット接続可
  - UDPパケットの外部への送信と受信を許可する
- RemoteCommander & RemoteMonitorの実行
  - RemoteCommander、RemoteMonitorは標準のLinuxconsoleモードです。
  - シェルから直接実行する

## III. 安全性

- リモート管理機能は、サーバーのセキュリティに関わるため、特に注意が必要です。
  - RMPassword.confのパスワードを公開しない
  - 疑わしい場合は、すぐにRMPassword.confのパスワードを変更し、変更後にP2Pサーバーを再起動してください。
  - RemoteCommanderには制御する機能があるので、最高権限を持つ管理者のみが使用してください。



## APIパーツ

### I. 文字列の初期化

- 初期化文字列またはサーバーアドレス文字列
  - は、暗号化された内容のアルファベット文字列
    - 写真提供: Shangun
    - コンテンツを変更すると、APIが正常に動作しなくなります
  - P2Pサーバーのドメイン名またはIPを指定する際に使用する（InitString）
    - PPCS\_Initialize("XXXXXXXXXXXX")
    - PPCS\_ConnectByServer(....., "xxxxxxxxxx")
    - PPCS\_NetworkDetectByServer(.....), "xxxxxxxxxx")

### II. 聞く・つながる

- TCP/IPソケットのListen & Connectに相当する機能です。
  - デバイスはListen()を呼び出し、クライアントのConnect()を待ちます。
  - Listen()とConnect()はどちらもDIDを必要とします。
- Listen & Connectは、接続が完了するまで、タイムアウトが発生するまで、または異常なエラーが発生するまで、プログラムの実行をブロックします。
  - オンライン補完では、セッションID >= 0が返されます。
    - PPCS\_Check()では、Sessionの詳細を確認することができます。
    - セッションには、データのRead/Writeに使用できる8Channelがあり、Channelは互いに独立しています。
    - PPCS\_Close()はセッションを閉じることができます。
    - PPCS\_ForceClose()は、データが渡されたかどうかに関わらず、Sessionを強制的に終了させます。

ロジニ・タイムアップや失敗(中国語)した場合は、マイナスの値が返されます。

### III. 書き込みとフロー制御

- PPCS\_Write()は直ちに返されます。
  - これは、APIには転送を担当する内部のThreadがあるからです。
- PPCS\_Write()で書き込まれたデータは配信が保証されます。
  - セッションが切断されない限り（タイムアウト、リモートまたはローカルクローズ）
- PPCS\_Check\_Buffer()では、Write inデータの送出残量を知ることができます。
- フローコントロールの実践：
  - PPCS\_Write()の前にライトバッファの量をチェックする
  - バッファの数がある値以上の場合、書き込みを一時停止し、Sleepして待つ
  - LiveVideoアプリケーションの場合は、Videoframeを捨てることを検討する
    - これは、この時点でのネットワークの伝送レートが、Video

### IV. 読み取りとタイムアウト

- PPCS\_Read()にはタイムアウト時間が設定されており、タイムアウト時間が経過するか、読み込まれるデータの期待される長さが満たされるまでブロックされます。
  - Return Timeout データの一部がすでに読み込まれている可能性があり、読み込まれたデータの正確な長さを知るためには、返されたDataSizeを決定する必要があります。

### V. ライブラリとバージョンリリース

- PPCS\_GetAPIVersion()は、APIのバージョン番号を取得できます。
- PPCS\_GetAPIVersion() Returns 0x01020304 -> Version 1.2.3.4
- 今後1.0.0.0は、APIがCかC++かを区別するために、バージョン番号の最

SunCloud P2Pソフトウェア・テクノロジーの  
後の中点を使用します。(中国語)

バージョン番号  
です。2.2.0

- X.X.X.0 -> C++バージョン
- X.X.X.1 -> Cバージョン

## 6. 互換性

- 特に断りのない限り、SunCloud P2P APIは前方互換性があります。
    - 新しいクライアントは古いデバイスにも対応
    - 古いクライアントも新しいデバイスに対応
  - 例えば、以下のように。
    - アプリ1.5.1版で対応可能な0.2.8デバイス
    - アプリ0.2.8版で対応可能な1.5.1デバイス
- これは、アプリをアップデートする必要がなく、新しいバージョンのデバイスに対応できることを意味しています。

尚云互联