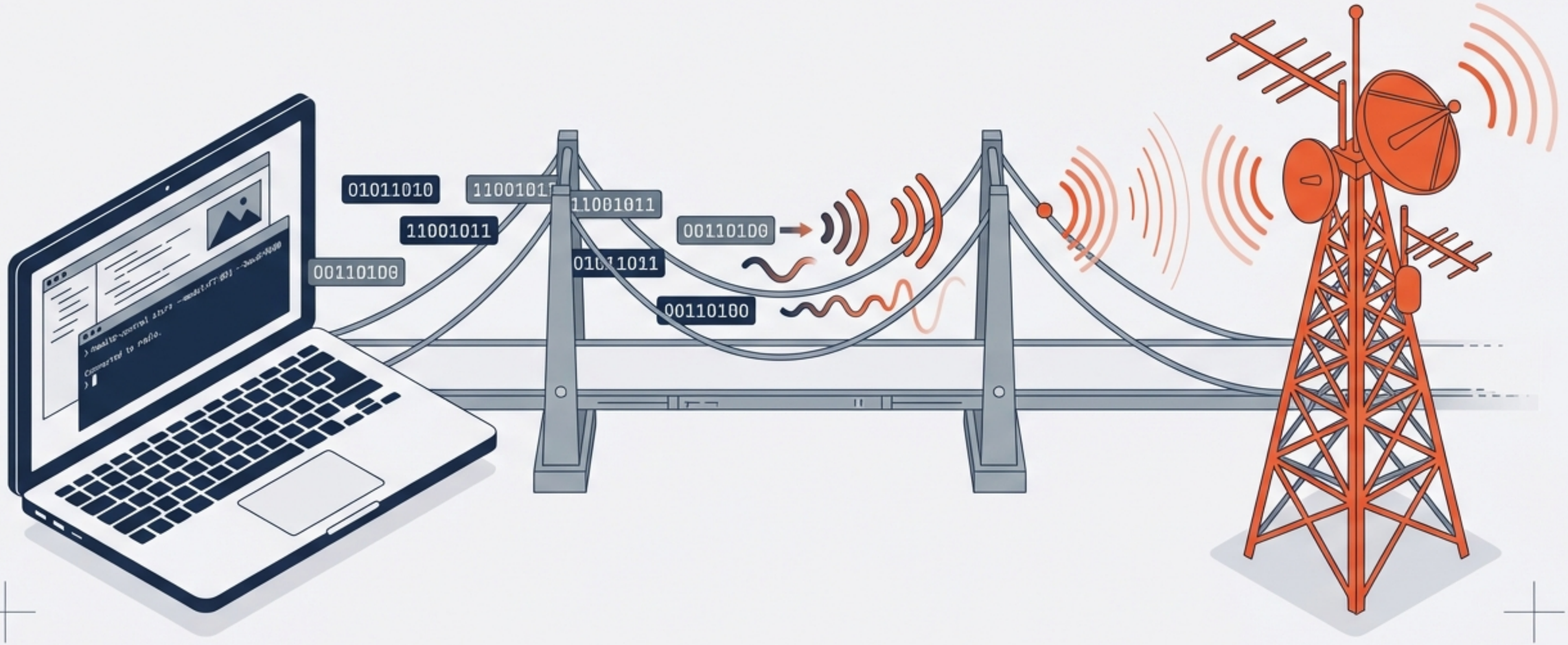
# Hamlib 入門

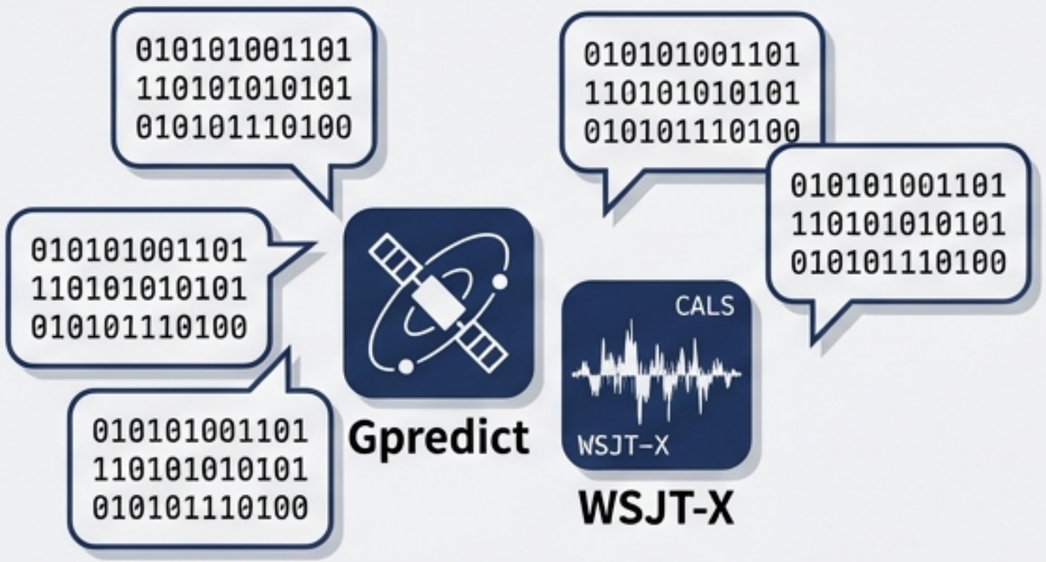
無線機とソフトウェアをつなぐ「架け橋」の仕組みと設定



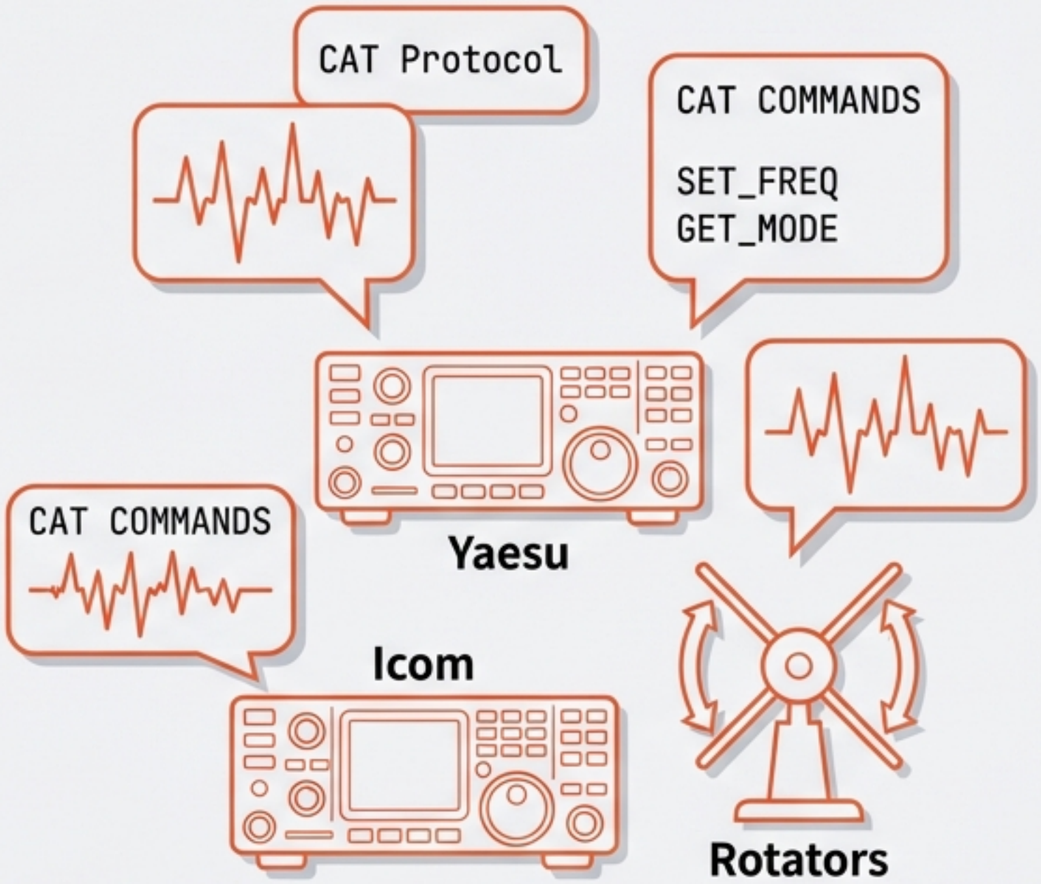

NotebookLM

# 言葉の壁

- ソフトウェアは「データ」を語る
- 無線機は「CATプロトコル」を語る
- 問題点：メーカーごとに言葉が違うため、会話が成立しない




ソフトウェア
(WSJT-X, Gpredict)

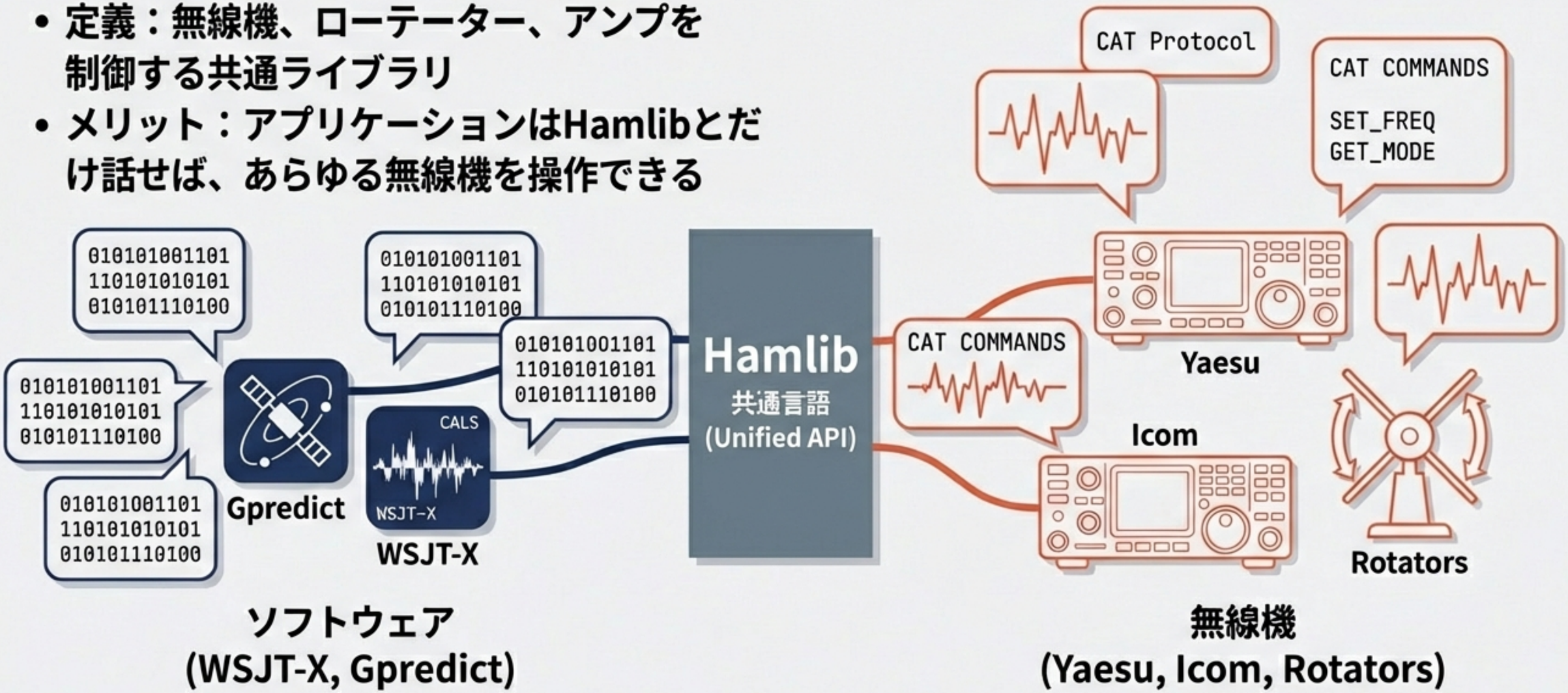



無線機
(Yaesu, Icom, Rotators)

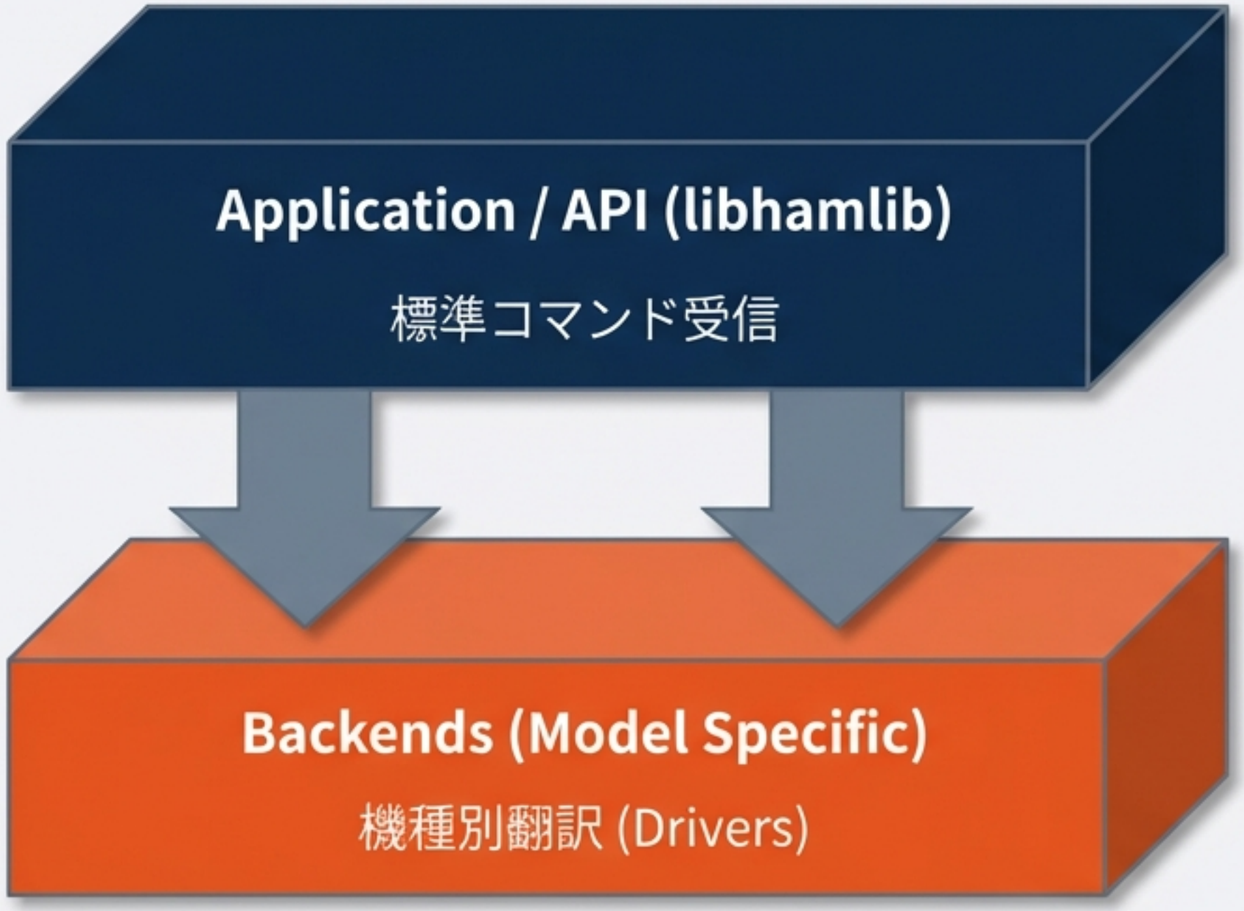
NotebookLM

# 解決策：Hamlib

- 定義：無線機、ローテーター、アンプを制御する共通ライブラリ
- メリット：アプリケーションはHamlibとだけ話せば、あらゆる無線機を操作できる




NotebookLM




# 仕組み：共通APIと機種別バックエンド

- Frontend: `libhamlib` - 標準化されたコマンドを受け取る
- Backend: 機種ごとの「方言」に翻訳して送信

NotebookLM

# 翻訳には「辞書」が必要：モデルIDの特定

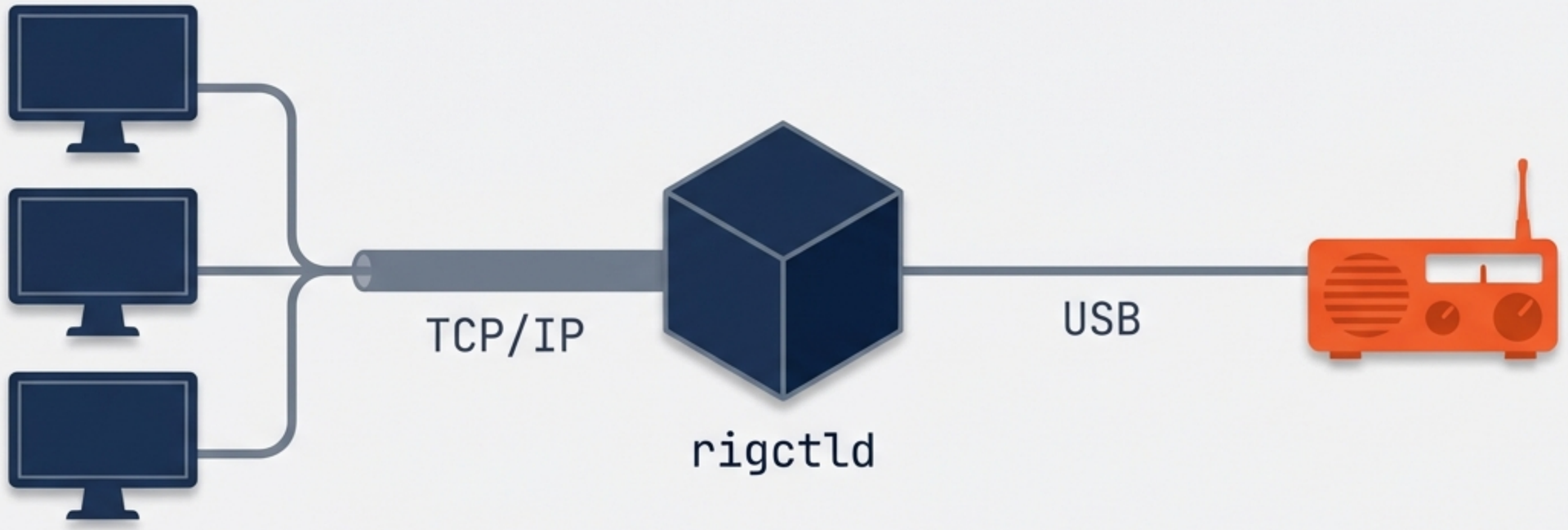
| | |
|---|---|
| **YAESU** | |
| FTX-1 | 1051 |
| FT-991A | 1035 |
| FT-817 | 1020 |
| **ICOM** | |
| IC-9700 | 3081 |
| IC-705 | 3085 |

※ 正しいID指定が接続の第一歩



NotebookLM

# rigctld：ハードウェアとソフトウェアを切り離す「仲介役」

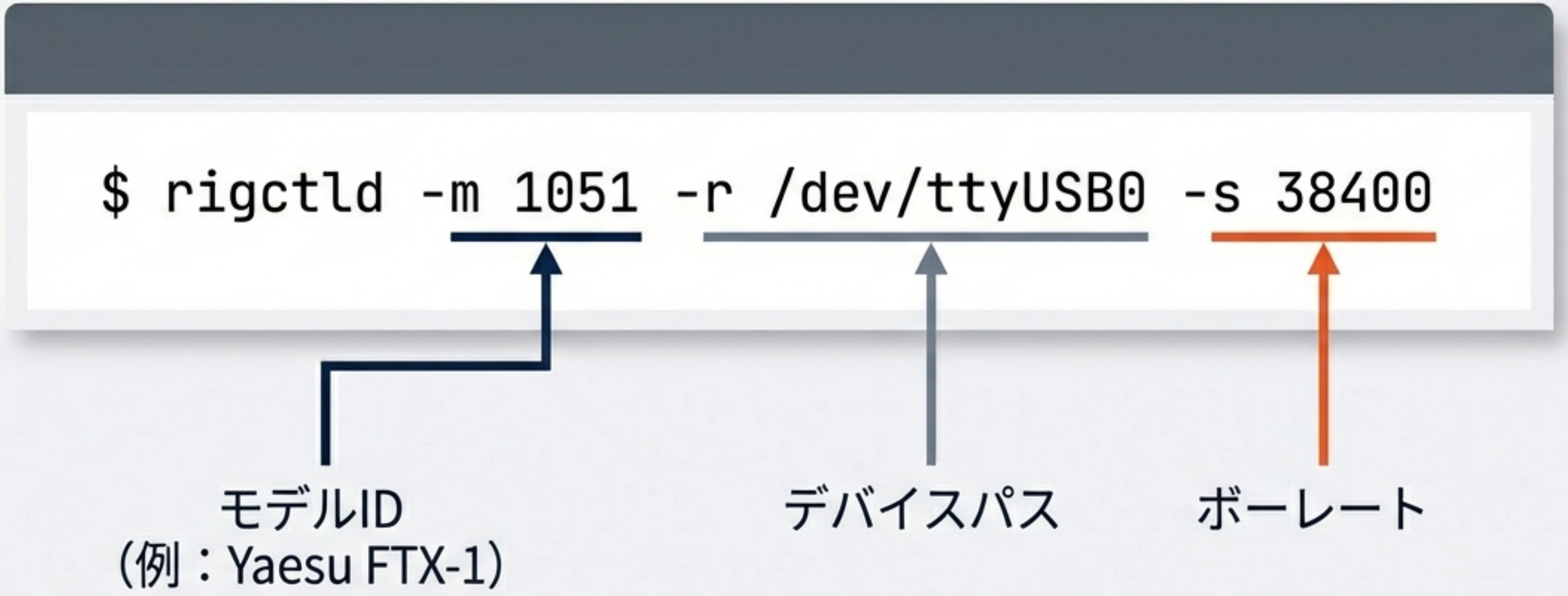



- 役割：ネットワーク・デーモン
- 機能：複数のプログラムが、ネットワークポートを介して1台の無線機を共有可能にする
- 利点：USBケーブルの物理的な制約から解放される

NotebookLM

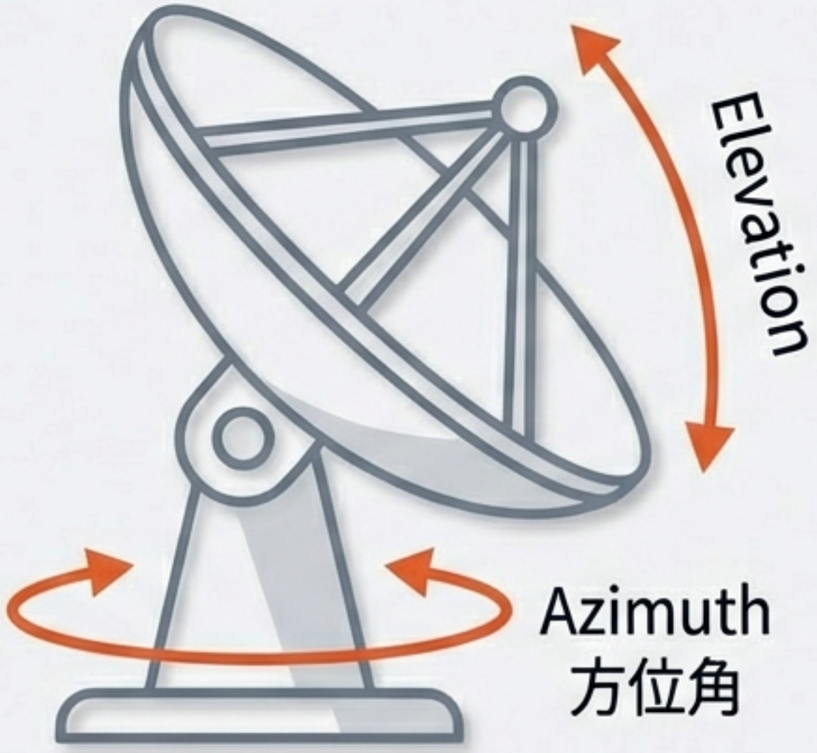
# 架け橋を架ける：コマンドラインでの接続テスト

このコマンドでサーバーが起動し、接続待機状態になる




NotebookLM

# アンテナも自由に：rotctldによるローテーター制御




ツール：rotctld（方位角と仰角を制御）

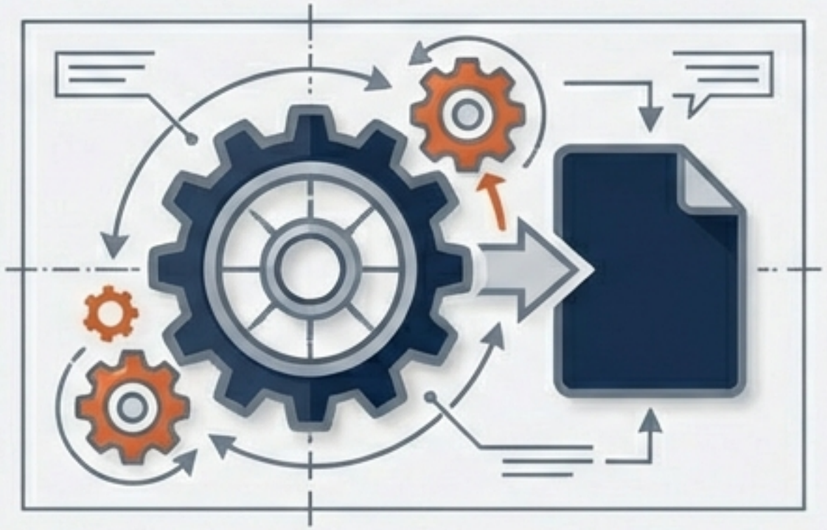
YAESU G-5500 ..... Model: 601

SkyWatcher ..... Model: 603

無線機と同じ「共通言語」で物理的な動きを制御

NotebookLM

# 手動操作からの解放：Systemdによる自動化



File: /etc/systemd/system/rigctld.service

systemctl enable rigctld

自動起動有効化

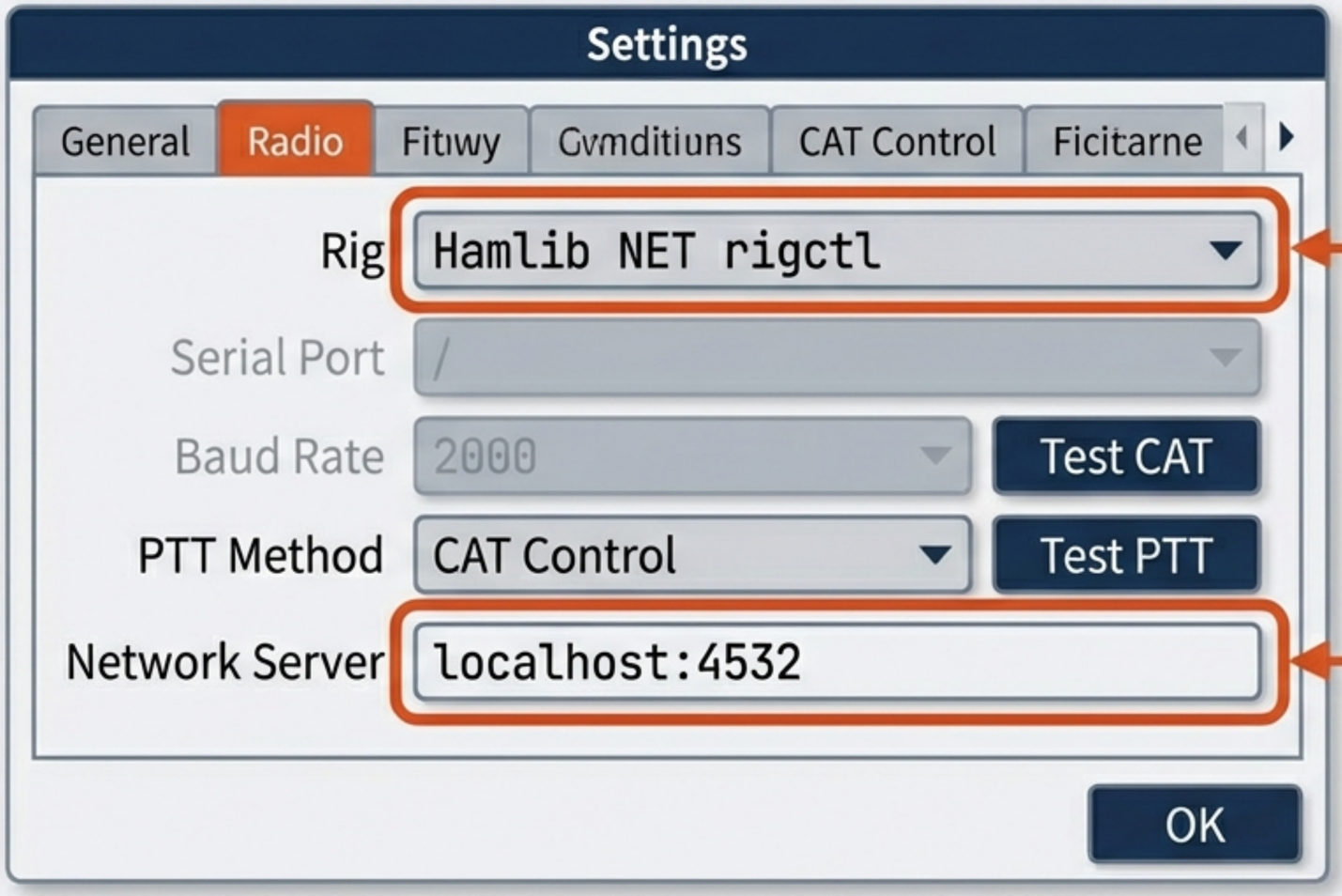
↓

systemctl start rigctld

サービス開始

NotebookLM

# 実践１：WSJT-Xの設定（ネットワーク経由）

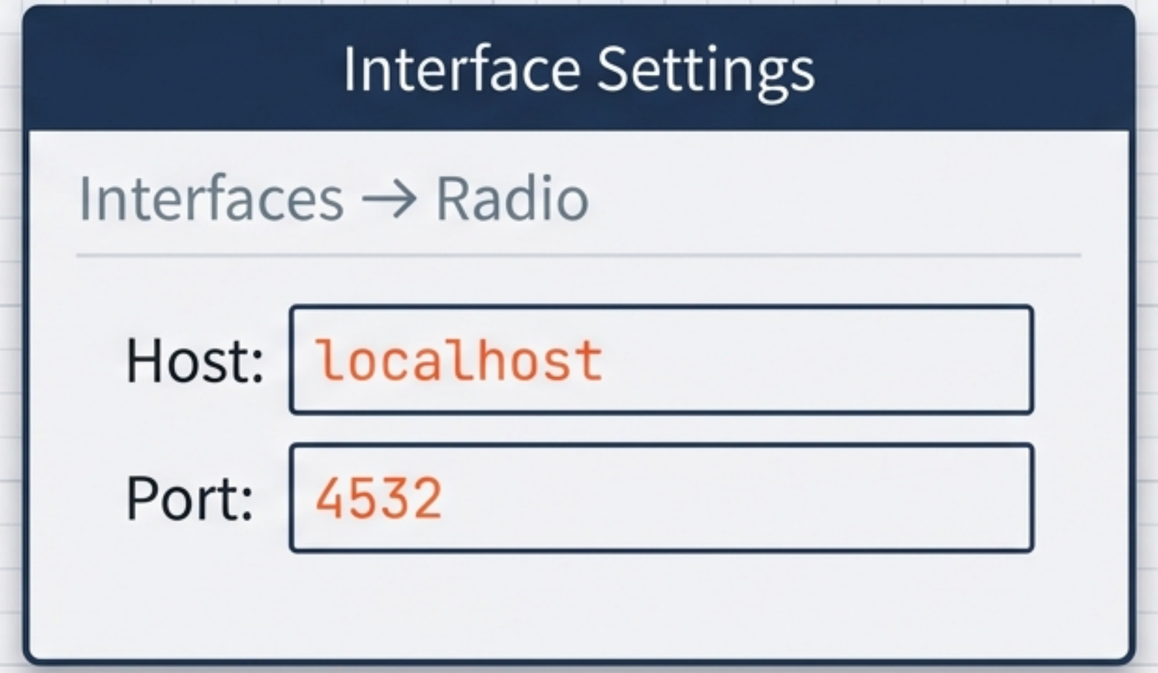



※個別の機種名ではなく、
NET rigctlを選択

デーモン(rigctld)に
接続するため

NotebookLM

# 実践 2：Gpredict による衛星追尾




リアルタイムでドップラー効果を補正し、周波数を自動追従

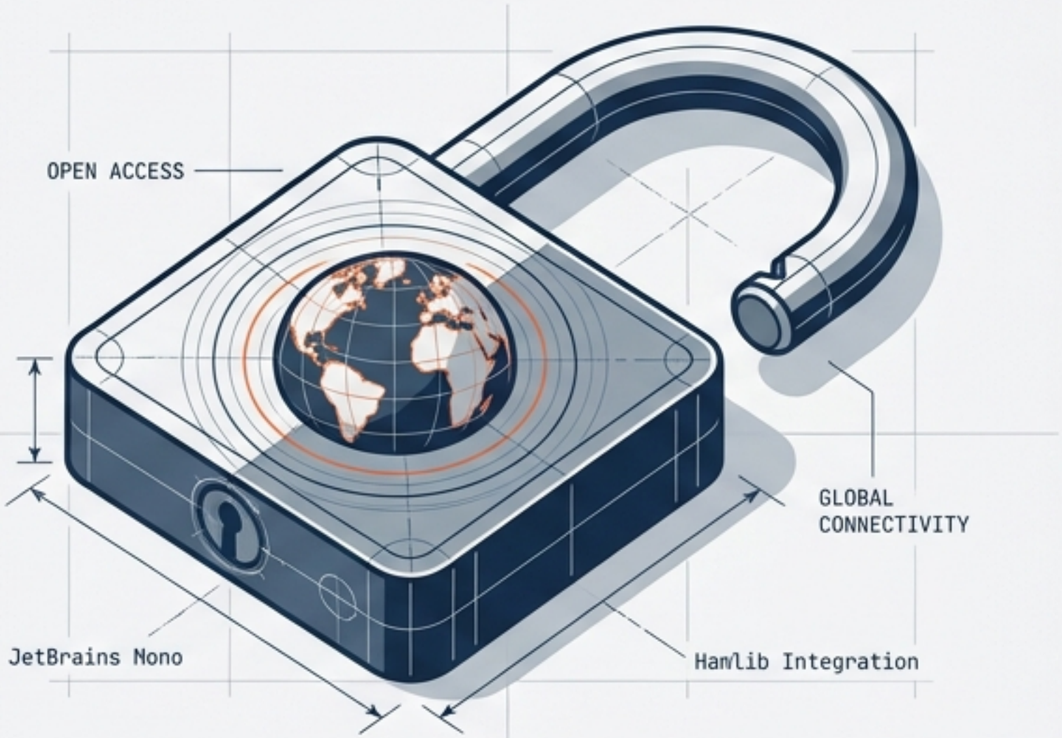
NotebookLM

# バージョンの壁：安定版 4.7 vs 次世代 5.0


最新の無線機対応にはアップデートが必要だが、ソフト側の対応状況を確認すること

NotebookLM

# オープンソースがもたらす無線の自由




**License:** GPL / LGPL による自由な利用と開発

**Summary:** ひとつのライブラリで、無限の可能性

**Goal:** 完全に統合されたシャック（無線室）の実現

NotebookLM

# 本日のまとめ：セットアップの4ステップ

- [x] 1. ID特定: 機種に対応するModel IDを調べる
- [x] 2. テスト: `rigctld` コマンドで接続確認
- [x] 3. 自動化: `systemd` でサービス化する
- [x] 4. 接続: ソフトの設定で`localhost:4532` を指定

NotebookLM

# Enjoy your controlled Radio Life.

ご清聴ありがとうございました

NotebookLM