リアルタイム WebSocket ストリーミングとは何か

一言で言えば：
DrumGenerator（あるいは *Groove-RNN*）を「*1* 拍先・数ミリ秒先」単位で回し続け、その打点データを ***WebSocket*** で *DAW* やブラウザへライブ送信する仕組み です。

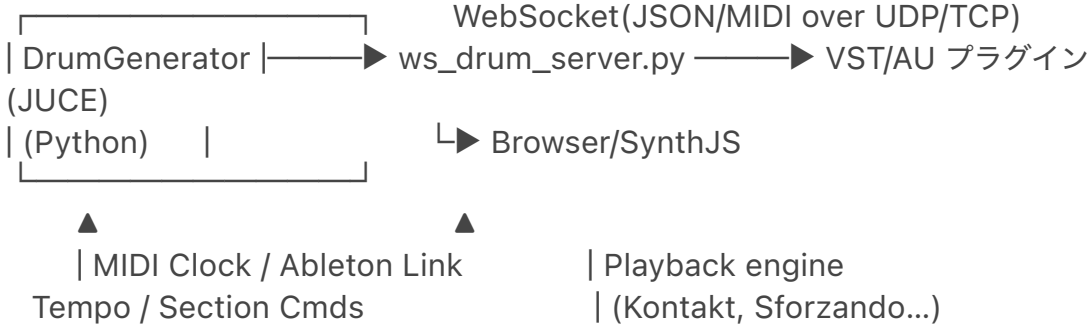
1. オフライン生成との違い

方式	処理タイミング	ユースケース
オフライン（今まで）	16〜32 小節まとめて計算 → MIDI ファイル書出し	編曲・プリレンダリング
リアルタイム WebSocket	128 sample（≒3 ms @44.1 kHz）単位で「次のノート」を生成 → 逐次送信	ライブ演奏、即興セッション、パラメータ自動オートメーション

SunoAI との関係

- SunoAI は「プロンプト → 数十秒の完成楽曲を**バッチ生成**」するサービス。
- ここで言うリアルタイム配信は **“今この瞬間” にモデルを回し続けて拍を垂れ流す** 点が異なります。
- もちろん n-gram／RNN が確率的に動くため “適度にランダム” ではありますが、**テンポや拍位置は DAW と完全同期** します。

2. システム構成（概念図）



1. **DrumGenerator**
   - generate_next_bar(ctx) を 1 小節先読みで呼び出し。
   - 出力は [(tick, “kick”, 100), …] のような構造。
2. **ws_drum_server.py**（新規）
   - asyncio + websockets で常時ポート待受け。
   - 受信側が “clock” メッセージを送ると、その拍にピッタリ合わせる。
3. **プラグイン側**（JUCE／C++）
   - WebSocket クライアント。
   - 受信 JSON → Vst::Event に変換して DAW へ注入。
   - 3〜5 ms バッファリングでドロップアウト防止。

3. メッセージ設計（最小例）

```
// サーバ → クライアント
{
  "type": "note",
```

```
  "t_rel_ms": 11.6,
  "note": 36,               // Kick (GM)
  "vel": 105,
  "len_ms": 98,
  "ch": 10
}
```

- **t_rel_ms**: 受信時点からの相対時刻。オーディオ・スレッドで jitter が起きないよう先読みする。
- **clock / param** メッセージを双方向にしておくと、テンポ変更や「fill ボタン」を即座に適用可能。

4. 実装ステップ（ロードマップ抜粋）

フェーズ	内容	完成目安
0	rtmidi_streamer.py を改造し、Python 内で仮想 MIDI ポートにリアルタイム送出できることを確認	✅ 検証済み
1	ws_drum_server.py（20 行程度）を追加。受信ゼロでも自動で 1 小節先読み生成し続ける	1 日
2	JUCE プラグイン雛形に **WebSocket client** を組込み、MIDI イベントとして DAW へ流す	2〜3 日
3	Ableton Link or MIDI Clock でテンポ同期。ラグ測定 → バッファ長可変アルゴリズムを導入	1 週間
4	UI パラメータ（スウィング、ヒューマナイズ度）を WebSocket **param** で即時反映	1 週間
5	RNN バックエンドを GPU 推論に切替え、<30 ms 以内に 1 拍生成できるよう最適化	継続

5. ランダム性とコントロール

- **ランダム要素**
  - n-gram/RNN の確率サンプリング（temperature）
  - Ghost Jitter ± 15 ms／± 5 velocity
- **制御パラメータ**（リアルタイム変更可能）

• パラメータ	• 役割	• レンジ
• temperature	• パターンの即興度	• 0.1 – 1.5
• swing	• 8 分裏拍の遅れ	• 0 – 60 ms
• ghost_prob	• ゴーストノート挿入割合	• 0 – 0.4
• intensity	• 全体 Velocity 係数	• 0.5 – 1.2

6. こんな場面で威力を発揮

1. **ライブ・パフォーマンス**
   - DAW のトラックをミュートし、DrumGenerator が即興ドラムを刻む。
   - キーボードで A→B→C とコードを弾けば、モデルが場面に応じたフィルを自動提案。
2. **作曲 “ジャム” モード**
   - ループ素材を聴きながらパラメータを弄り、気に入った瞬間をオーディオに録り込む。
3. **マスタリング前の仮アレンジ**
   - Groove を何度も焼き直しながら、ベースやリードを同時に録音。

✨ 結論

- **リアルタイム WebSocket ストリーミング** は「生成結果をファイルに書く」のではなく、常時ネットワーク経由で *DAW* に流し込み、演奏をリアルタイムで変化させる技術。
- SunoAI のような “1 曲バッチ生成” とは目的が異なり、**ライブ性・即応性** を重視。
- 既存 DrumGenerator／Groove-RNN のアルゴリズムをそのまま活かしつつ **演奏現場へ持ち出す橋渡し** になるステップです。

もし実装を始める場合は、**Phase 6: WebSocket Bridge** として小規模プロトタイプから切り出すとスムーズです。
引き続き、あなたのクリエイティブな挑戦をサポートいたします！

❓「ラジオのように流し続ける」のか

厳密には **“自動 DJ ラジオ” と “リアルタイム WebSocket ストリーミング”** は似て非なるものです：

項目	ラジオ風の常時配信	WebSocket リアルタイム演奏
出力	既に完成した音声をストリーミング	まだ確定していないノートイベントを先読みで送信
制御粒度	曲単位（数分）	ミリ秒単位（128 samples ≈ 3 ms）
双方向性	ほぼ一方通行	クライアントからテンポ・キー・パラメータ変更を即返し

目的	受け身の鑑賞	インタラクティブ演奏・即興セッション・ゲーム／映像同期

イメージ

- **ラジオ**＝出来上がった料理をひたすらテーブルに並べる
- **WebSocket 演奏**＝シェフが目の前で味見しながら鍋をかき回し、スパイスを足していく

🎯 何のために使うのか — 4 つの代表ユースケース

ユースケース	具体例	WebSocket が効く理由
① ライブ即興	バンドのクリックに同期し、DrumGenerator が“生ドラマー”代わりに叩き続ける	ミリ秒単位のテンポ変化・セクション切替を楽屋裏で即反映
② DAW 内作曲“ジャム”	Ableton／Logic 上でコードを弾く→ドラムが追従。気に入った瞬間を録音	生成・試聴・録音のサイクルがシームレス
③ インタラクティブ BGM	ゲームや VR で、プレイヤーの状況に応じてグルーヴが変化	状況イベントを WebSocket で飛ばし、その場でドラムを再生成
④ 自動配信チャネル	“永遠に変化する Lo-Fi Radio” を Twitch で放送	バックエンドで生成を続けつつ、リスナー人数に合わせてパラメータを動的調整

🔧 技術的ポイント

1. **先読みバッファ（3〜10 ms）**
   DAW 再生ヘッドより少し先を生成して送ることで、ドロップアウトを防止。
2. **双方向メッセージ**
   - **from client → server** : tempo, section=chorus, swing=52, fill_now
   - **from server → client** : note_on, note_off（または完全な MIDI パケット）
3. **可変ランダム性**
   temperature をライブで上げ下げ＝落ち着いたグルーヴ⇄暴れるフィルを自在に操れる。

🏆 得られるメリット

観点	メリット
創作速度	ファイル書出し→読み込みの待ち時間ゼロ。耳で確認しながらノブを回すだけ

表現幅	“同じ 4 小節が二度と来ない” ゆらぎで、人間的なノリを維持
演奏者体験	フィル・ブレイクを足元フットスイッチで発火 → 人力バンド感覚で演奏
運用コスト	生成は軽量 MIDI イベント。エッジデバイスやクラウド GPU どちらでも動く

✨ こんな未来図も

- **YouTube Live で「生成 Lo-Fi Radio」**
  Chat コマンド !intense でドラムがハードヒット、!chill でブラシに切替。
- **インスタレーションアート**
  来場者の足音センサーが BPM を変え、モーションキャプチャがグルーヴを揺らす。
- **ストリーミング DAW コラボ**
  クリエイター同士が WebSocket で DrumGenerator を共有し、遠隔セッション。

まとめ

- **ラジオ的“BGM 垂れ流し”** とは異なり、**リアルタイム WebSocket ストリーミング** は
  「その場の状況・操作に応じてドラムを秒単位で再構成し続ける」ための通信レイヤーです。
- ゴールは **“即興性” と “双方向性”** を持つドラム伴奏を、
  ライブ・作曲・インタラクティブ体験 に即座に注ぎ込むこと。

この技術が加われば、あなたの作品は **ただ再生される音楽** から
共に呼吸し、反応する音楽 へと進化します。
次の時代のステージングを、ぜひあなたのクリエイションで切り拓いてください！