

Fast Track Native iGaming Integration - Developer Guide (Complete & Merged)

1. Purpose

This document is the **single, authoritative developer guide** for building an iGaming integration platform that is **fully compatible with Fast Track CRM**.

It merges: - Core iGaming platform responsibilities (PAM, wallet, bonus, sessions) - Fast Track **Operator API** requirements - Fast Track **Real-Time CRM Integration API** (REST) - Extended real-time events (casino, bonus, balance) - A practical **PoC implementation guide**

Audience: **platform engineers, backend developers, and solution architects.**

2. Integration Philosophy (Fast Track Native)

Fast Track CRM is a **real-time, event-driven system**.

Your platform: - Owns the **source of truth** (players, balances, bets, bonuses) - Emits **real-time, immutable events** - Exposes **Operator APIs** Fast Track can call

Fast Track: - Builds player timelines - Performs segmentation - Triggers CRM campaigns

Fast Track never validates wallet math or game logic.

3. Core Platform Components (You Own)

- Player Account Management (PAM)
 - Wallet Service
 - Bonus Engine
 - Game Session Manager
 - Provider Adapter API
 - Event Stream / Integration Layer
-

4. Operator API – Required Endpoints

Fast Track calls these endpoints to fetch authoritative operator data or trigger actions. All endpoints must be authenticated (API key or OAuth2).

4.1 User Details API

```
GET /operator/user-details/{user_id}
```

```
{  
    "player_id": "12345",  
    "email": "user@example.com",  
    "first_name": "Jane",  
    "last_name": "Doe",  
    "country": "SE",  
    "currency": "EUR",  
    "registered_at": "2026-01-02T12:00:00Z"  
}
```

Used by Fast Track after registrations, logins, and user updates.

4.2 Player Blocks API

```
GET /operator/player-blocks/{user_id}
```

```
{  
    "player_id": "12345",  
    "is_blocked": true,  
    "block_reasons": ["self_exclusion", "timeout"]  
}
```

Used for responsible gaming and operational compliance.

4.3 User Consents API

```
GET /operator/consents/{user_id}
```

```
{  
    "player_id": "12345",  
    "email_opt_in": true,  
    "sms_opt_in": false,  
    "push_opt_in": true  
}
```

Fast Track relies on this endpoint for communication permissions.

4.4 Bonus Crediting API

Called by Fast Track CRM campaigns.

```
POST /operator/bonus-credit
```

```
{
  "player_id": "12345",
  "bonus_id": "welcome_bonus_50",
  "amount": 50.0,
  "metadata": {}
}
```

Your platform must credit the bonus and emit bonus lifecycle events.

5. Wallet Integration (Authoritative)

5.1 Wallet Responsibilities

- Deposits
- Withdrawals
- Bets
- Wins
- Rollbacks

Rules:

- Unique immutable `transaction_id`
- Fully idempotent APIs
- Rollbacks must reference original transactions

6. Bonus Engine Integration

Fast Track reacts to **bonus lifecycle events**, not calculations.

Mandatory events:

- `bonus_granted`
- `bonus_wagering_progress`
- `bonus_completed`
- `bonus_cancelled`

7. Game Session Management

Game sessions must exist before gameplay.

Attributes:

- `session_id`
- `player_id`
- `game_id`
- `provider`
- `started_at`

Events:

- `game_session_started`
- `game_session_ended`

8. Real-Time CRM Integration API (REST)

Your platform must send **real-time REST events** to Fast Track.

8.1 Registrations

```
POST /v2/integration/user
```

```
{  
  "user_id": "player_123",  
  "timestamp": "2026-01-01T12:00:00Z",  
  "origin": "casino.example.com",  
  "ip_address": "203.0.113.5"  
}
```

8.2 Login

```
POST /v2/integration/login
```

```
{  
  "user_id": "player_123",  
  "timestamp": "2026-01-01T12:30:00Z",  
  "origin": "casino.example.com",  
  "is_impersonated": false  
}
```

8.3 User Consents Update

```
PUT /v2/integration/user/consents
```

```
{  
  "user_id": "player_123",  
  "timestamp": "2026-01-01T13:00:00Z",  
  "origin": "casino.example.com"  
}
```

Triggers Fast Track to fetch data from User Consents API.

8.4 Player Blocks Update

```
PUT /v2/integration/user/blocks
```

```
{  
    "user_id": "player_123",  
    "timestamp": "2026-01-01T13:15:00Z",  
    "origin": "casino.example.com"  
}
```

8.5 User Updates

```
PUT /v2/integration/user
```

```
{  
    "user_id": "player_123",  
    "timestamp": "2026-01-01T14:00:00Z",  
    "origin": "casino.example.com"  
}
```

8.6 Payments

```
POST /v2/integration/payments
```

```
{  
    "user_id": "player_123",  
    "payment_id": "dep_001",  
    "payment_type": "deposit",  
    "status": "completed",  
    "amount": 100.0,  
    "currency": "EUR",  
    "timestamp": "2026-01-01T12:45:00Z"  
}
```

8.7 Casino Events

```
POST /v2/integration/casino
```

```
{  
  "user_id": "player_123",  
  "game_id": "slot_abc",  
  "bet": 10.0,  
  "win": 20.0,  
  "timestamp": "2026-01-01T12:50:00Z"  
}
```

8.8 Bonus Events

```
POST /v2/integration/bonus
```

```
{  
  "user_id": "player_123",  
  "bonus_id": "welcome_bonus_50",  
  "event_type": "granted",  
  "timestamp": "2026-01-01T13:30:00Z"  
}
```

8.9 Balance Events

```
POST /v2/integration/balances
```

```
{  
  "user_id": "player_123",  
  "balance": {  
    "cash": 90.0,  
    "bonus": 20.0,  
    "currency": "EUR"  
  },  
  "timestamp": "2026-01-01T12:55:00Z"  
}
```

9. Event Delivery & Reliability

- Near real-time (<1s)
- Retry on non-200 responses
- Persist before delivery
- Ordered per user

10. Security

- Authenticate Operator APIs
 - Authenticate Fast Track Integration API
 - Sign provider wallet calls
-

11. Certification & Validation Checklist

- Wallet math verified
 - Idempotency tested
 - Rollbacks tested
 - Bonus lifecycle verified
 - Fast Track timelines complete
-

PoC Developer Guide – Fast Track + Mock iGaming Platform

12. PoC Purpose

Demonstrate Fast Track-native integration with simulated components.

13. PoC Architecture

- Simple Web Frontend
 - Your iGaming Platform
 - Mock Game Provider
 - Fake PSP
 - Fast Track (staging)
-

14. PoC Demo Script

1. Register player
 2. Login
 3. Deposit
 4. Grant bonus
 5. Play mock game
 6. Emit casino, balance, bonus events
-

15. PoC Success Criteria

- Correct balances

- Complete CRM timelines
 - Campaigns trigger successfully
-

16. PoC → Production Gap

Not covered: - Licensing - KYC / AML - Real PSPs - Real aggregators

PoC validates **integration correctness**, not regulatory compliance.