REALNODE PROTOCOL

B2B Integration Documentation

Detailed technical deployment manual.

Universal Architecture
This document details the technical integration of the RealNode protocol on your platform. The interface you are about to integrate is designed to dynamically adapt to your service tier (**RN Insight**, **RN Sentinel** or **RN Vault**) without requiring you to modify your code later on.

1 Step 0: Prerequisites and API Keys

Before you start coding, you must obtain your access credentials to the RealNode infrastructure.

1. **Account Creation:** Go to the RealNode administrator portal (`https://app.realnode.emkaylabs.tech/signup`) to create your enterprise account. The quick registration form will only ask for your **Company Name**, **Work Email**, and an **Admin Password**.
2. **Key Generation:** At the end of this registration, the portal will generate a **key pair** for you: a **Public Key** (e.g., `pk_live_XXX`) intended for your website, and a **Secret Key** (e.g., `sk_live_XXX`) strictly reserved for your backend server.
3. **Storage:** These keys identify and secure your requests. Keep your Secret Key absolutely secure in your `.env` file. Never expose it in the browser.
4. **Note on the chosen plan:** The `event_id` field in the `/consume` API depends on the subscribed plan:
   - **RN Insight** : no `event_id` required — passive analysis only.
   - **RN Sentinel** : `event_id` optional — consumption per session.
   - **RN Vault** : `event_id` **mandatory** — create the event from your admin dashboard before calling `/consume`.

2 Step 1: SDK Initialization

The first technical step consists of loading the RealNode engine on your website.

Recommended placement: This block must be inserted **on all pages** requiring RealNode security (checkout page, client dashboard), immediately before the closing `</body>` tag.

```
<script>
  // Global configuration to initialize RealNode
  window.RN_CONFIG = {
      // ONLY use your Public Key (starts with pk_)
      apiKey: "pk_live_YOUR-PUBLIC-KEY-HERE",
      clientId: "your-store-name" // Optional, useful to filter logs
  };
</script>
<!-- Asynchronous security engine loading -->
```

```
<script type="module" src="https://api.emkaylabs.tech/rn-client.js"></
    script>
```

3 Step 2: User Interface (UI) Integration

You must now integrate the visual elements necessary for user interaction. The code is structured into logical modules.

3.1 2.1. Global Styling (CSS)

Recommended placement: To be integrated into your main stylesheet (e.g., `styles.css`), or within the `<head>` tag of your pages.

```
<style>
.rn-panel { background: #111; color: #fff; padding: 25px; border-radius:
    12px; font-family: sans-serif; }
.rn-badge { background: rgba(0, 255, 136, 0.1); color: #00ff88; padding: 5
    px 10px; border-radius: 4px; }
.rn-btn { background: #00ff88; color: #000; border: none; padding: 10px 15
    px; border-radius: 6px; cursor: pointer; font-weight: bold; margin-
    right: 5px; }
.rn-btn-secondary { background: #444; color: #fff; border: none; padding:
    10px 15px; border-radius: 6px; cursor: pointer; }
.rn-btn-danger { background: rgba(239, 68, 68, 0.1); color: #ef4444;
    border: 1px solid #ef4444; }
.rn-input { background: #222; border: 1px solid #444; color: #fff; padding
    : 10px; border-radius: 6px; width: 100%; margin-bottom: 10px; }
.rn-device-row { display: flex; justify-content: space-between; align-
    items: center; border-bottom: 1px solid #222; padding: 15px 0; }
</style>
```

3.2 2.2. The Checkout Area

Recommended placement: To be integrated into the DOM of your **checkout page**, at the intended location for the main action button.

```
<div style="margin-bottom: 30px;">
    <p>Authorized tickets: <span id="rn-quota-user" class="rn-badge">--</
        span> / <span id="rn-quota-max">--</span></p>
    <input type="number" id="rn-ticket-quantity" class="rn-input" value="1
        " min="1" style="width: 80px; display: inline-block;">
    <button id="rn-btn-acheter" data-rn-protect class="rn-btn">Secure
        Purchase</button>
    <button id="rn-btn-demander-quota" style="display:none;" class="rn-btn
        ">Request more tickets</button>
</div>
```

3.3 2.3. The User Profile Area (Device Management)

Recommended placement: To be integrated into the user dashboard (**"My Account"** or **"Profile"**).

```
<div class="rn-panel">
    <h3>My Secured Devices</h3>
    <button id="rn-btn-revente" class="rn-btn-secondary">Free my device (
        For resale)</button>
    <div id="rn-devices-list" style="margin-top:15px;">Loading...</div>

    <h3 style="margin-top: 30px;">Transfer my access to a third party</h3>
    <input type="text" id="rn-transfer-id" class="rn-input" placeholder="
        Recipient ID">
```

```
    <button id="rn-btn-transferer" class="rn-btn-secondary">Transfer my
       account</button>

    <h3 style="margin-top: 30px;">Emergency Recovery</h3>
    <input type="text" id="rn-recovery-code" class="rn-input" placeholder=
       "Recovery Code (Ex: RN-REC-...)">
    <button id="rn-btn-recuperer" class="rn-btn">Restore via Code</button>

    <p style="margin-top:10px; font-size: 0.85em; color: #aaa;">Lost your
       code?</p>
    <input type="email" id="rn-rescue-email" class="rn-input" placeholder=
       "your@email.com">
    <button id="rn-btn-rescue" class="rn-btn-secondary">Request a Rescue
       link</button>
</div>
```

4 Step 3: Frontend Logic (JavaScript)

This JavaScript module orchestrates the client-side integration. It serves as the bridge between the user interface and the RealNode cryptographic API.

Recommended placement: To be included **in your main JavaScript file** (e.g., `main.js`) or at the end of the document via a `<script>` tag.

```
document.addEventListener("DOMContentLoaded", async () => {

    // 0. DISPLAY REAL-TIME QUOTAS
    window.addEventListener('rn-purchase-success', (e) => {
        const result = e.detail;
        if (result.remaining !== undefined) {
            document.getElementById('rn-quota-user').innerText = result.
               remaining;
            document.getElementById('rn-quota-max').innerText = result.
               max_limit;
            if(result.remaining === 0) {
                document.getElementById('rn-btn-acheter').style.display =
                   'none';
                document.getElementById('rn-btn-demander-quota').style.
                   display = 'inline-block';
            }
        }
    });

    try {
        await window.RN_Monitor.restoreSession();
    } catch(e) { /* The user does not have an active session yet */ }

    // 1. DEVICE LOADING AND MANAGEMENT
    const listDiv = document.getElementById('rn-devices-list');
    setTimeout(() => {
        if(listDiv.innerHTML === 'Loading...') listDiv.innerHTML = '
           Untracked devices (Insight Plan)';
    }, 3000);

    window.addEventListener('rn:v2:sensor-ready', async () => {
        try {
            const devices = await window.RN_Monitor.listDevices(window.
               RN_Monitor.currentIdh);
            listDiv.innerHTML = '';
            devices.data.forEach(dev => {
                listDiv.innerHTML += `
                <div class="rn-device-row">
                    <span style="font-family: monospace;">Device: ${dev.
                       device_hash.substring(0,8)}...</span>
                    <button onclick="revocDevice('${dev.device_hash}')"
                       class="rn-btn rn-btn-danger">Block (Stolen)</button
                       >
                </div>`;
            });
        } catch (e) { /* Ignore for RN Insight level (passive) */ }
    });

    // Profile management utility functions
    window.revocDevice = async (hash) => {
        if(confirm("Ban this device from the network?")) {
            await window.RN_Monitor.revokeDevice(hash);
            alert("Device successfully banned.");
```

```
    }
};

document.getElementById('rn-btn-revente').onclick = async () => {
    if(confirm("Clean this device of your identity before reselling it
       ?")) {
        await window.RN_Monitor.requestResale(window.RN_Monitor.
           currentIdh);
        alert("Device freed.");
    }
};

document.getElementById('rn-btn-transferer').onclick = async () => {
    const newId = document.getElementById('rn-transfer-id').value;
    if(confirm("Irreversibly transfer your account to this user?")) {
        try { await window.RN_Monitor.transferDevice(newId); alert("
           Transfer successful."); }
        catch (e) { alert("Error during transfer."); }
    }
};

// 3. PURCHASE LOGIC (CORE INTEGRATION)
// NOTE: data-rn-protect makes the SDK intercept the click
// automatically. The 'rn:verified' event is emitted after biometrics.
document.getElementById('rn-btn-acheter').addEventListener('rn:
   verified', (e) => {
    const result = e.detail;
    const desiredQuantity = parseInt(document.getElementById('rn-
       ticket-quantity').value) || 1;

    if (result.remaining !== undefined) {
        document.getElementById('rn-quota-user').innerText = result.
           remaining;
        document.getElementById('rn-quota-max').innerText = result.
           max_limit;
    }

    if (result.status === 'authorized') {
        if (result.remaining === undefined || desiredQuantity <=
           result.remaining) {
            const rnPayload = {
                quantity: desiredQuantity,
                idh: window.RN_Monitor.currentIdh,
                device_hash: result.device_hash,
                device_token: localStorage.getItem("rn_device_token"),
                client_id: window.RN_CONFIG.clientId || "rn-universal-
                   client"
            };

            // -> SEND TO YOUR SERVER <-
            fetch('/api/your-checkout', {
                method: 'POST',
                headers: { 'Content-Type': 'application/json' },
                body: JSON.stringify(rnPayload)
            }).then(response => {
                alert("Security process validated. Finalizing
                   transaction...");
            });
        } else { alert("Operation denied: Insufficient quota."); }
    }
```

```
    });

    // 4. RESTORATION AND EXTENSION
    document.getElementById('rn-btn-demander-quota').onclick = async () =>
        {
        await window.RN_Monitor.requestQuotaExtension(); alert("Request
            sent.");
    };
    document.getElementById('rn-btn-recuperer').onclick = async () => {
        const code = document.getElementById('rn-recovery-code').value;
        try { await window.RN_Monitor.recover(code); alert("Identity
            restored!"); }
        catch (error) { alert("Invalid recovery code."); }
    };
    document.getElementById('rn-btn-rescue').onclick = async () => {
        const email = document.getElementById('rn-rescue-email').value;
        try { await window.RN_Monitor.requestRescue(email); alert("Link
            sent via email."); }
        catch (error) { alert("Error during request."); }
    };
});
```

5 Step 4: Transaction Validation (Your Backend)

Once your server (e.g., Node.js, PHP, Python) has received the `rnPayload` from your frontend, it must transmit these proofs to the RealNode API.

Implementation: You must create an endpoint on your server (e.g., `/api/checkout`) capable of receiving the `rnPayload` from the frontend, and then executing the HTTP attestation request described below to the RealNode infrastructure.

Critical Security (Dual Key): For this server-to-server step, you must use your **Secret Key** (`sk_live_...`) obtained in your administrator space. Never put this secret key in your frontend JavaScript code.

Request to perform from your backend to RealNode:

```
POST https://api.emkaylabs.tech/consume
Headers:
    Content-Type: application/json
    X-RealNode-API-Key:    pk_live_YOUR-PUBLIC-KEY
    X-RealNode-Secret-Key: sk_live_YOUR-SECRET-KEY
Body:
{
    "quantity":    3,
    "idh":         "THE-ID-RETRIEVED-ON-FRONTEND",
    "device_hash": "RN-DEVICE-XXXX",
    "device_token": "TOKEN-XXX",
    "client_id":   "your-store-name",
    "event_id":    "EVT-XXXX"  // Mandatory for RN Vault
}
```

Note: the `event_id` field is mandatory only for the RN Vault plan. It corresponds to the identifier of the event created in your administrator dashboard.

6 Step 5: Testing your Integration

To ensure you have configured everything correctly, perform these two simple tests:

1. **Frontend Test (Quotas)**: Open your checkout page in your browser. If the integration is successful, you will see the "Authorized tickets" counter change from `-` to your actual quota (e.g., `5 / 5`) a few seconds after the page loads.
2. **Backend Test (Consumption)**: From your server's terminal, you can simulate a backend validation with this `curl` command:

```
curl -X POST https://api.emkaylabs.tech/consume \
  -H "Content-Type: application/json" \
  -H "X-RealNode-API-Key: pk_live_YOUR-PUBLIC-KEY" \
  -H "X-RealNode-Secret-Key: sk_live_YOUR-SECRET-KEY" \
  -d '{
    "quantity": 1,
    "idh": "test-id",
    "device_hash": "test-hash",
    "device_token": "test-token",
    "client_id": "test-client"
    // Add "event_id": "EVT-XXXX" if RN Vault plan
  }'
```

HTTP 200 Response: server-to-server communication operational.
HTTP 403 Response: check your API keys.
Error "Invalid event ID": add the `event_id` *field (RN Vault plan only).*

7 Step 6: Special Cases & Best Practices

7.1 6.1. Integration on React / Vue / Angular (SPA)

The examples in Step 3 use `DOMContentLoaded`. In a Single Page Application (SPA), components are mounted dynamically.

- **React:** Place the JavaScript logic in a `useEffect()` hook instead of `DOMContentLoaded`.
- **Vue.js:** Place the logic in the `mounted()` lifecycle hook.
- **Angular:** Use `ngOnInit()`.

Make sure to clean up your event listeners (`removeEventListener`) when unmounting the component to avoid memory leaks.

7.2 6.2. Content Security Policy (CSP)

If your site uses strict CSP headers, the RealNode script may be blocked. Please whitelist our domain in your server configuration:

```
Content-Security-Policy:
  script-src 'self' https://api.emkaylabs.tech https://*.hanko.io;
  connect-src 'self' https://api.emkaylabs.tech https://*.hanko.io;
```

Technical Support: For any integration questions, contact our team at **security@emkaylabs.tech**. We respond within 24 hours.