

REALNODE PROTOKOLL

B2B Integrationshandbuch — Vanilla JS

Detailliertes technisches Bereitstellungshandbuch.

Universelle Architektur

Dieses Dokument detailliert die technische Integration des RealNode-Protokolls auf Ihrer Plattform mithilfe von Vanilla JS. Die zu integrierende Schnittstelle passt sich dynamisch an Ihr Servicelevel an (**RN Insight**, **RN Sentinel** oder **RN Vault**), ohne dass Sie den Code nachträglich ändern müssen.

1 Schritt 0: Voraussetzungen und API-Schlüssel

Bevor Sie mit dem Codieren beginnen, müssen Sie Ihre Zugangsdaten zur RealNode-Infrastruktur einholen.

1. **Kontoerstellung:** Besuchen Sie das RealNode-Adminportal (<https://app.realnode.emkaylabs.tech/signup>), um Ihr Unternehmenskonto zu erstellen. Das Schnellregistrierungsformular fragt lediglich nach: **Firmenname**, **Geschäftliche E-Mail** und einem **Admin-Passwort**.
2. **Schlüsselgenerierung:** Am Ende dieser Registrierung generiert das Portal ein **Schlüssel-paar** für Sie: einen **Public Key** (z. B. `pk_live_XXX`) für Ihre Website und einen **Secret Key** (z. B. `sk_live_XXX`), der streng für Ihren Backend-Server reserviert ist.
3. **Aufbewahrung:** Diese Schlüssel identifizieren und sichern Ihre Anfragen. Bewahren Sie Ihren geheimen Schlüssel in absoluter Sicherheit in Ihrer `.env`-Datei auf. Geben Sie ihn niemals im Browser preis.
4. **Hinweis zum gewählten Plan:** Das `event_id`-Feld in der `/consume`-API hängt vom gebuchten Plan ab:
 - **RN Insight** : Keine `event_id` erforderlich — nur passive Analyse.
 - **RN Sentinel** : `event_id` optional — Verbrauch pro Sitzung.
 - **RN Vault** : `event_id` **obligatorisch** — erstellen Sie das Ereignis über Ihr Admin-Dashboard, bevor Sie `/consume` aufrufen.

2 Schritt 1: SDK-Initialisierung

Der erste technische Schritt besteht darin, die RealNode-Engine auf Ihrer Site zu laden.

Empfohlene Platzierung: Dieser Block muss **auf allen Seiten** eingefügt werden, die RealNode-Sicherheit erfordern (Checkout-Seite, Kundenbereich), unmittelbar vor dem schließenden `</body>`-Tag.

```
<script>
// Globale Konfiguration zur Initialisierung von RealNode
window.RN_CONFIG = {
  // Verwenden Sie AUSSCHLIESSLICH Ihren Public Key (beginnt mit pk_)
  apiKey: "pk_live_IHR-PUBLIC-KEY-HIER",
```

```
        clientId: "name-ihres-shops" // Optional, nuetzlich zum Filtern von
        Logs
    };
</script>
<!-- Asynchrones Laden der Sicherheits-Engine -->
<script type="module" src="https://api.emkaylabs.tech/rn-client.js"></
    script>
```

3 Schritt 2: Benutzeroberfläche (UI) Integration

Sie müssen nun die für die Benutzerinteraktion erforderlichen visuellen Elemente integrieren. Der Code ist in logische Module gegliedert.

3.1 2.1. Globale Stile (CSS)

Empfohlene Platzierung: Fügen Sie dies in Ihr Haupt-Stylesheet (z. B. `styles.css`) oder in das `<head>`-Tag Ihrer Seiten ein.

```
<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. Der Checkout-Bereich

Empfohlene Platzierung: Integrieren Sie dies in das DOM Ihrer **Bestellbestätigungsseite**, an der Stelle, die für die primäre Aktionsschaltfläche vorgesehen ist.

```
<div style="margin-bottom: 30px;">
  <p>Autorisierte 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">Sicher
    Kaufen</button>
  <button id="rn-btn-demander-quota" style="display:none;" class="rn-btn
    ">Weitere Tickets anfragen</button>
</div>
```

3.3 2.3. Das Benutzerprofil (Geräteverwaltung)

Empfohlene Platzierung: Integrieren Sie dies in das Benutzer-Dashboard (“**Mein Konto**” oder “**Profil**”).

```
<div class="rn-panel">
  <h3>Meine Sicheren Ger te</h3>
  <button id="rn-btn-revente" class="rn-btn-secondary">Ger t freigeben
    (Vor Wiederverkauf)</button>
  <div id="rn-devices-list" style="margin-top:15px;">Wird geladen...</
    div>

  <h3 style="margin-top: 30px;">Zugang an Dritte bertragen </h3>
  <input type="text" id="rn-transfer-id" class="rn-input" placeholder="
    ID des Empf ngers">
```

```

<button id="rn-btn-transferer" class="rn-btn-secondary">Konto
    bertragen </button>

<h3 style="margin-top: 30px;">Notfall-Wiederherstellung</h3>
<input type="text" id="rn-recovery-code" class="rn-input" placeholder=
    "Recovery Code (Z.B.: RN-REC-...)">
<button id="rn-btn-recuperer" class="rn-btn"> ber Code
    wiederherstellen</button>

<p style="margin-top:10px; font-size: 0.85em; color: #aaa;">Code
    verloren?</p>
<input type="email" id="rn-rescue-email" class="rn-input" placeholder=
    "ihre@email.com">
<button id="rn-btn-rescue" class="rn-btn-secondary">Rescue-Link
    anfordern</button>
</div>

```

4 Schritt 3: Frontend-Logik (JavaScript)

Dieses JavaScript-Modul orchestriert die clientseitige Integration. Es stellt die Verbindung zwischen der Benutzeroberfläche und der kryptografischen RealNode-API her.

Empfohlene Platzierung: In Ihre Haupt-JavaScript-Datei (z. B. main.js) oder am Ende des Dokuments über ein `<script>`-Tag einfügen.

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

    // 0. QUOTEN IN ECHTZEIT ANZEIGEN
    // Das Event 'rn-purchase-success' wird nach jeder erfolgreichen
    // Verifizierung ausgelöst
    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';
            }
        }
    });

    // Sitzungswiederherstellung (wenn der Benutzer zur Seite zurückkehrt)
    try {
        await window.RN_Monitor.restoreSession();
    } catch(e) { /* Der Benutzer hat noch keine aktive Sitzung */ }

    // 1. LADEN UND VERWALTUNG DER GERAETE (Plaene V2/V3)
    const listDiv = document.getElementById('rn-devices-list');
    setTimeout(() => {
        if (listDiv.innerHTML === 'Wird geladen...') listDiv.innerHTML = '
            Nicht verfolgte Geraete (Insight Plan)';
    }, 3000);

    // Das Event 'rn:v2:sensor-ready' wird ausgelöst, wenn das
    // Benutzerprofil bereit ist
    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;">Geraet: ${dev.
                        device_hash.substring(0,8)}...</span>
                    <button onclick="revocDevice('${dev.device_hash}')"
                        class="rn-btn rn-btn-danger">Sperren (Diebstahl)</
                        button>
                </div>';
            });
        } catch (e) { /* Wird fuer die Stufe RN Insight (passiv) ignoriert
            */ }
    });
});
```

```

});

// Hilfsfunktionen fuer die Profilverwaltung
window.revocDevice = async (hash) => {
  if(confirm("Dieses Geraet aus dem Netzwerk verbannen?")) {
    await window.RN_Monitor.revokeDevice(hash);
    alert("Geraet erfolgreich verbannt.");
  }
};

document.getElementById('rn-btn-revente').onclick = async () => {
  if(confirm("Geraet vor dem Weiterverkauf von Ihrer Identitaet trennen?")) {
    await window.RN_Monitor.requestResale(window.RN_Monitor.currentIdh);
    alert("Geraet freigegeben.");
  }
};

document.getElementById('rn-btn-transferer').onclick = async () => {
  const newId = document.getElementById('rn-transfer-id').value;
  if(confirm("Ihr Konto unwiderruflich auf diesen Benutzer uebertragen?")) {
    try { await window.RN_Monitor.transferDevice(newId); alert("Uebertragung erfolgreich."); }
    catch (e) { alert("Fehler waehrend der Uebertragung."); }
  }
};

// 3. DIE KAUFLOGIK (HERZSTUECK DER INTEGRATION)
// HINWEIS: data-rn-protect sorgt dafuer, dass das SDK den Klick
// automatisch abfaengt. Das Event 'rn:verified' wird nach der
// Biometrie ausgeloeset.
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"
      };

      // -> AN IHREN SERVER SENDEN <-
      fetch('/api/ihr-checkout', {

```

```

        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(rnPayload)
    }).then(response => {
        alert("Sicherheitsprozess validiert. Transaktion wird
            abgeschlossen...");
    });
    } else { alert("Vorgang abgelehnt: Unzureichende Quote."); }
}
});

// 4. WIEDERHERSTELLUNG UND ERWEITERUNG
document.getElementById('rn-btn-demander-quota').onclick = async () => {
    {
        await window.RN_Monitor.requestQuotaExtension(); alert("Anfrage
            gesendet.");
    }
};
document.getElementById('rn-btn-recuperer').onclick = async () => {
    const code = document.getElementById('rn-recovery-code').value;
    try { await window.RN_Monitor.recover(code); alert("Identitaet
        wiederhergestellt!"); }
    catch (error) { alert("Ungueltiger Wiederherstellungscode."); }
};
document.getElementById('rn-btn-rescue').onclick = async () => {
    const email = document.getElementById('rn-rescue-email').value;
    try { await window.RN_Monitor.requestRescue(email); alert("Link
        per E-Mail gesendet."); }
    catch (error) { alert("Fehler bei der Anfrage."); }
};
});

```

5 Schritt 4: Transaktionsvalidierung (Ihr Backend)

Sobald Ihr Server (z. B. Node.js, PHP, Python) die `rnPayload` von Ihrem Frontend empfangen hat, muss er diese Nachweise an die RealNode-API übertragen.

Implementierung: Sie müssen einen Endpunkt auf Ihrem Server erstellen (z. B. `/api/checkout`), der die `rnPayload` vom Frontend entgegennehmen kann und anschließend die unten beschriebene HTTP-Attestierungsanfrage an die RealNode-Infrastruktur ausführt.

Kritische Sicherheit (Doppelschlüssel) : Für diesen Server-zu-Server-Schritt müssen Sie Ihren **Secret Key** (`sk_live_...`) verwenden, den Sie in Ihrem Administrationsbereich erhalten haben. Fügen Sie diesen geheimen Schlüssel niemals in Ihren Frontend-JavaScript-Code ein.

Anfrage von Ihrem Backend an RealNode:

```
POST https://api.emkaylabs.tech/consume
Headers:
  Content-Type: application/json
  X-RealNode-API-Key: pk_live_IHR-PUBLIC-KEY
  X-RealNode-Secret-Key: sk_live_IHR-SECRET-KEY
Body:
{
  "quantity": 3,
  "idh": "DIE-VOM-FRONTEND-ABGERUFENE-ID",
  "device_hash": "RN-DEVICE-XXXX",
  "device_token": "TOKEN-XXX",
  "client_id": "name-ihres-shops",
  "event_id": "EVT-XXXX" // Obligatorisch fuer RN Vault
}
```

Hinweis: Das Feld `event_id` ist nur für den Plan RN Vault zwingend erforderlich. Es entspricht der Kennung des Ereignisses, das in Ihrem Administrations-Dashboard erstellt wurde.

6 Schritt 5: Testen Sie Ihre Integration

Um sicherzustellen, dass Sie alles richtig konfiguriert haben, führen Sie diese beiden einfachen Tests durch:

1. **Frontend-Test (Quoten) :** Öffnen Sie Ihre Zahlungsseite in Ihrem Browser. Wenn die Integration erfolgreich war, sehen Sie, wie sich der Zähler „Autorisierte Tickets“ einige Sekunden nach dem Laden der Seite von – auf Ihre tatsächliche Quote (z. B. 5 / 5) ändert.
2. **Backend-Test (Verbrauch) :** Vom Terminal Ihres Servers aus können Sie eine Backend-Validierung mit diesem `curl`-Befehl simulieren:

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


*HTTP-Antwort 200: Server-zu-Server-Kommunikation betriebsbereit.
HTTP-Antwort 403: Überprüfen Sie Ihre API-Schlüssel.
Fehler „Invalid event ID“: Fügen Sie das Feld `event_id` hinzu (nur RN Vault Plan).*

7 Schritt 6: Sonderfälle & Best Practices

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

Die Beispiele in Schritt 3 verwenden `DOMContentLoaded`. In einer Single Page Application (SPA) werden Komponenten dynamisch geladen.

- **React:** Platzieren Sie die JavaScript-Logik in einem `useEffect()`-Hook anstelle von `DOMContentLoaded`.
- **Vue.js:** Platzieren Sie die Logik im `mounted()`-Lebenszyklus.
- **Angular:** Verwenden Sie `ngOnInit()`.

Stellen Sie sicher, dass Sie Ihre Event-Listener (`removeEventListener`) beim Entladen der Komponente bereinigen, um Speicherlecks zu vermeiden.

7.2 6.2. Content Security Policy (CSP)

Wenn Ihre Site strikte CSP-Header verwendet, wird das RealNode-Skript möglicherweise blockiert. Bitte autorisieren Sie unsere Domain in Ihrer Serverkonfiguration:

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

Technischer Support : Bei Fragen zur Integration kontaktieren Sie unser Team unter **security@emkaylabs.tech**. Wir antworten innerhalb von 24 Stunden.