

REALNODE PROTOCOL

Documentation d'Intégration B2B

Manuel technique détaillé de déploiement.

Architecture Universelle

Ce document détaille l'intégration technique du protocole RealNode sur votre plateforme. L'interface que vous allez intégrer est conçue pour s'adapter dynamiquement à votre niveau de service (**RN Insight**, **RN Sentinel** ou **RN Vault**) sans que vous n'ayez besoin de modifier votre code par la suite.

1 Étape 0 : Prérequis et Clés d'API

Avant de commencer à coder, vous devez obtenir vos accès à l'infrastructure RealNode.

1. **Création de compte** : Rendez-vous sur le portail administrateur RealNode (<https://app.realnode.emkaylabs.tech/signup>) pour créer votre compte entreprise. Le formulaire d'inscription rapide vous demandera uniquement : **Company Name**, **Work Email**, et un **Admin Password**.
2. **Génération des clés** : À la fin de cette inscription, le portail vous générera une **paire de clés** : une **Clé Publique** (ex : `pk_live_XXX`) destinée à votre site web, et une **Clé Secrète** (ex : `sk_live_XXX`) strictement réservée à votre serveur backend.
3. **Conservation** : Ces clés identifient et sécurisent vos requêtes. Gardez votre Clé Secrète en sécurité absolue dans votre fichier `.env`. Ne l'exposez jamais dans le navigateur.
4. **Note sur le plan choisi** : Le champ `event_id` dans l'API `/consume` dépend du plan souscrit :
 - **RN Insight** : aucun `event_id` requis — analyse passive uniquement.
 - **RN Sentinel** : `event_id` optionnel — consommation par session.
 - **RN Vault** : `event_id` **obligatoire** — créez l'événement depuis votre dashboard admin avant d'appeler `/consume`.

2 Étape 1 : Initialisation du SDK

La première étape technique consiste à charger le moteur RealNode sur votre site.

Emplacement recommandé : Ce bloc doit être inséré **sur toutes les pages** nécessitant la sécurité RealNode (page de paiement, espace client), immédiatement avant la balise fermante `</body>`.

```
<script>
  // Configuration globale pour initialiser RealNode
  window.RN_CONFIG = {
    // Utilisez UNIQUEMENT votre Cle Publique (commence par pk_)
    apiKey: "pk_live_VOTRE-CLE-PUBLIQUE-ICI",
    clientId: "nom-de-votre-boutique" // Optionnel, utile pour filtrer
      les logs
  };
</script>
<!-- Chargement du moteur de securite asynchrone -->
```

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

3 Étape 2 : Intégration de l'Interface Utilisateur (UI)

Vous devez à présent intégrer les éléments visuels nécessaires à l'interaction utilisateur. Le code est structuré en modules logiques.

3.1 2.1. Le style global (CSS)

Emplacement recommandé : À intégrer dans votre feuille de style principale (ex : `styles.css`), ou dans la balise `<head>` de vos 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. La Zone de Paiement

Emplacement recommandé : À intégrer dans le DOM de votre **page de validation de commande**, à l'emplacement prévu pour le bouton d'action principal.

```
<div style="margin-bottom: 30px;">
  <p>Billets autorisés : <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">Acheter de
    mani\`ere s\`ecuris\`ee</button>
  <button id="rn-btn-demander-quota" style="display:none;" class="rn-btn
    ">Demander plus de billets</button>
</div>
```

3.3 2.3. La Zone de Profil Utilisateur (Gestion des appareils)

Emplacement recommandé : À intégrer dans le tableau de bord utilisateur (“**Mon Compte**” ou “**Profil**”).

```
<div class="rn-panel">
  <h3>Mes Appareils S\`ecuris\`es</h3>
  <button id="rn-btn-revente" class="rn-btn-secondary">Lib\`erer mon
    appareil (Pour revente)</button>
  <div id="rn-devices-list" style="margin-top:15px;">Chargement...</div>

  <h3 style="margin-top: 30px;">Transf\`erer mes acc\`es \`a un tiers</
    h3>
```

```

<input type="text" id="rn-transfer-id" class="rn-input" placeholder="
  ID du destinataire">
<button id="rn-btn-transférer" class="rn-btn-secondary">Transférer
  mon compte</button>

<h3 style="margin-top: 30px;">Restauration d'Urgence</h3>
<input type="text" id="rn-recovery-code" class="rn-input" placeholder="
  Recovery Code (Ex: RN-REC-...)">
<button id="rn-btn-recupérer" class="rn-btn">Restaurer via Code</
  button>

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

```

4 Étape 3 : La Logique Frontend (JavaScript)

Ce module JavaScript orchestre l'intégration côté client. Il assure la liaison entre l'interface utilisateur et l'API cryptographique RealNode.

Emplacement recommandé : À inclure dans votre fichier JavaScript principal (ex : main.js) ou en fin de document via une balise <script>.

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

    // 0. AFFICHER LES QUOTAS EN TEMPS REEL
    // L'événement 'rn-purchase-success' est émis après chaque
    // vérification réussie
    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';
            }
        }
    });

    // Restauration de session (si l'utilisateur est revenu sur la page)
    try {
        await window.RN_Monitor.restoreSession();
    } catch(e) { /* L'utilisateur n'a pas encore de session active */ }

    // 1. CHARGEMENT ET GESTION DES APPAREILS (Plans V2/V3)
    const listDiv = document.getElementById('rn-devices-list');
    setTimeout(() => {
        if (listDiv.innerHTML === 'Chargement...') listDiv.innerHTML = '
            Appareils non traces (Plan Insight)';
    }, 3000);

    // L'événement 'rn:v2:sensor-ready' est émis quand le profil
    // utilisateur est prêt
    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;">Appareil : ${dev
                            .device_hash.substring(0,8)}...</span>
                        <button onclick="revocDevice('${dev.device_hash}')"
                            class="rn-btn rn-btn-danger">Bloquer (Vol)</button>
                    </div>';
            });
        } catch (e) { /* Ignore pour le niveau RN Insight (passif) */ }
    });

    // Fonctions utilitaires de gestion de profil
```

```

window.revocDevice = async (hash) => {
  if(confirm("Bannir cet appareil du reseau ?")) {
    await window.RN_Monitor.revokeDevice(hash);
    alert("Appareil banni avec succes.");
  }
};

document.getElementById('rn-btn-revente').onclick = async () => {
  if(confirm("Nettoyer cet appareil de votre identite avant de le revendre ?")) {
    await window.RN_Monitor.requestResale(window.RN_Monitor.currentIdh);
    alert("Appareil libere.");
  }
};

document.getElementById('rn-btn-transferer').onclick = async () => {
  const newId = document.getElementById('rn-transfer-id').value;
  if(confirm("Transferer votre compte de facon irreversible a cet utilisateur ?")) {
    try { await window.RN_Monitor.transferDevice(newId); alert("Transfert reussi."); }
    catch (e) { alert("Erreur lors du transfert."); }
  }
};

// 3. LA LOGIQUE D'ACHAT (COEUR DE L'INTEGRATION)
// NOTE: data-rn-protect fait en sorte que le SDK intercepte le clic
// automatiquement. L'evenement 'rn:verified' est emis apres la biometrie.
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"
      };

      // -> ENVOI VERS VOTRE SERVEUR <-
      fetch('/api/votre-checkout', {
        method: 'POST',
        headers: { 'Content-Type': 'application/json' },
        body: JSON.stringify(rnPayload)
      })
    }
  }
});

```

```

        }).then(response => {
            alert("Processus de securite valide. Finalisation de
                la transaction...");
        });
    } else { alert("Operation refusee : Quota insuffisant."); }
}
});

// 4. RESTAURATION ET EXTENSION
document.getElementById('rn-btn-demander-quota').onclick = async () =>
{
    await window.RN_Monitor.requestQuotaExtension(); alert("Demande
        envoyee.");
};
document.getElementById('rn-btn-recuperer').onclick = async () => {
    const code = document.getElementById('rn-recovery-code').value;
    try { await window.RN_Monitor.recover(code); alert("Identite
        restauree !"); }
    catch (error) { alert("Code de secours invalide."); }
};
document.getElementById('rn-btn-rescue').onclick = async () => {
    const email = document.getElementById('rn-rescue-email').value;
    try { await window.RN_Monitor.requestRescue(email); alert("Lien
        envoye par email."); }
    catch (error) { alert("Erreur lors de la demande."); }
};
});

```

5 Étape 4 : Validation de la Transaction (Votre Backend)

Une fois que votre serveur (ex : Node.js, PHP, Python) a reçu le `rnPayload` depuis votre frontend, il doit transmettre ces preuves à l'API RealNode.

Implémentation : Vous devez créer un point de terminaison sur votre serveur (ex : `/api/checkout`) capable de réceptionner le `rnPayload` du frontend, puis d'exécuter la requête HTTP d'attestation décrite ci-dessous vers l'infrastructure RealNode.

Sécurité Critique (Double Clé) : Pour cette étape serveur-à-serveur, vous devez utiliser votre **Clé Secrète** (`sk_live_...`) obtenue dans votre espace administrateur. Ne mettez jamais cette clé secrète dans votre code JavaScript frontend.

Requête à effectuer depuis votre backend vers RealNode :

```
POST https://api.emkaylabs.tech/consume
Headers:
  Content-Type: application/json
  X-RealNode-API-Key: pk_live_VOTRE-CLE-PUBLIQUE
  X-RealNode-Secret-Key: sk_live_VOTRE-CLE-SECRETE
Body:
{
  "quantity": 3,
  "idh": "L-ID-RECUPERE-SUR-LE-FRONTEND",
  "device_hash": "RN-DEVICE-XXXX",
  "device_token": "TOKEN-XXX",
  "client_id": "nom-de-votre-boutique",
  "event_id": "EVT-XXXX" // Obligatoire pour RN Vault
}
```

Note : le champ `event_id` est obligatoire uniquement pour le plan RN Vault. Il correspond à l'identifiant de l'événement créé dans votre dashboard administrateur.

6 Étape 5 : Tester votre Intégration

Pour s'assurer que vous avez tout configuré correctement, effectuez ces deux tests simples :

1. **Test Frontend (Quotas) :** Ouvrez votre page de paiement dans votre navigateur. Si l'intégration est réussie, vous verrez le compteur "Billets autorisés" passer de - à votre quota réel (ex : 5 / 5) quelques secondes après le chargement de la page.
2. **Test Backend (Consommation) :** Depuis le terminal de votre serveur, vous pouvez simuler une validation backend avec cette commande `curl` :

```
curl -X POST https://api.emkaylabs.tech/consume \
-H "Content-Type: application/json" \
-H "X-RealNode-API-Key: pk_live_VOTRE-CLE-PUBLIQUE" \
-H "X-RealNode-Secret-Key: sk_live_VOTRE-CLE-SECRETE" \
-d '{
  "quantity": 1,
  "idh": "test-id",
  "device_hash": "test-hash",
  "device_token": "test-token",
  "client_id": "test-client"
  // Ajouter "event_id": "EVT-XXXX" si plan RN Vault
}'
```

Réponse HTTP 200 : communication serveur-à-serveur opérationnelle.
Réponse HTTP 403 : vérifier vos clés API.
Erreur “Invalid event ID” : ajouter le champ `event_id` (plan RN Vault uniquement).

7 Étape 6 : Cas Particuliers & Bonnes Pratiques

7.1 6.1. Intégration sur React / Vue / Angular (SPA)

Les exemples de l'Étape 3 utilisent `DOMContentLoaded`. Dans une application Single Page (SPA), les composants sont montés dynamiquement.

- **React** : Placez la logique JavaScript dans un hook `useEffect()` au lieu de `DOMContentLoaded`.
- **Vue.js** : Placez la logique dans le cycle de vie `mounted()`.
- **Angular** : Utilisez `ngOnInit()`.

Assurez-vous de nettoyer vos écouteurs d'événements (`removeEventListener`) lors du démontage du composant pour éviter les fuites mémoire.

7.2 6.2. Content Security Policy (CSP)

Si votre site utilise des en-têtes CSP stricts, le script `RealNode` peut être bloqué. Veuillez autoriser notre domaine dans votre configuration serveur :

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

Support Technique : Pour toute question d'intégration, contactez notre équipe à security@emkaylabs.tech. Nous répondons sous 24h.