

REALNODE PROTOCOL

B2B Integratiegids — React & Next.js (NPM)

Technische implementatiehandleiding voor moderne JavaScript-omgevingen.

Componentarchitectuur (React)

Dit document beschrijft de integratie van het RealNode-protocol via ons officiële NPM-pakket. Specifiek gebouwd voor React en Next.js, biedt dit pakket kant-en-klare componenten (afrekenknoppen, quotumbadges, apparaatbeheerders) om uw transactietrechter te beveiligen zonder de noodzaak om complexe interfaces te ontwerpen.

1 Stap 0: Account Aanmaken en API-sleutels Ophalen

De allereerste stap is het aanmaken van een account op de RealNode-portal en het ophalen van uw toegangssleutels. Deze sleutels identificeren uw applicatie veilig bij onze infrastructuur.

1. **Registratie:** Bezoek de beheerdersportal (<https://app.realnode.emkaylabs.tech/signup>) om uw bedrijfsaccount aan te maken. U moet uw bedrijfsnaam, een professioneel e-mailadres en een beheerderswachtwoord opgeven.
2. **Sleutels ophalen:** De portal genereert een uniek sleutelpaar voor uw omgeving:
 - **Public Key** (formaat: `pk_live_...`): Te gebruiken in uw frontend React-code. Het is volkomen veilig om deze sleutel in de browser te tonen.
 - **Secret Key** (formaat: `sk_live_...`): Uitsluitend te gebruiken op uw backend-server om transacties cryptografisch te valideren. Deel deze sleutel nooit en neem deze niet op in uw frontend-code.

2 Stap 1: Het NPM-pakket Installeren

Open uw terminal in de hoofdmap van uw React- of Next.js-project en voer de volgende opdracht uit:

```
npm install @emkaylabs/realnode-sdk
```

Wat doet deze opdracht? Het downloadt en installeert het `@emkaylabs/realnode-sdk` pakket in uw project. Dit pakket bevat:

- De kern HTTP-client die communiceert met de RealNode API (`/verify`, `/consume`, `/check-session`, enz.).
- Productieklare React-componenten (`RealNodeProvider`, `RealNodeCheckoutButton`, `RealNodeDeviceManager`, `RealNodeQuotaBadge`).
- Hulpprogramma's voor sessiebeheer en veilige lokale opslag (levenscyclus van het apparaattonen).

Na installatie is het pakket beschikbaar in uw `node_modules` map en wordt ernaar verwezen in uw `package.json`.

3 Stap 2: Globale Configuratie — De Provider

De `RealNodeProvider` component fungeert als de centrale hub. Het initialiseert de beveiligingsengine en stelt de SDK-context beschikbaar aan alle geneste componenten. Het moet **exact één keer** worden gemount, zo hoog mogelijk in uw componentenboom.

Waar plaats ik deze code? In uw hoofdbestand: `App.jsx` voor Create React App, of `_app.jsx` / `layout.jsx` voor Next.js.

```
import { RealNodeProvider } from '@emkaylabs/realnode-sdk/react';

export default function App({ children }) {
  return (
    <RealNodeProvider apiKey="pk_live_UW_PUBLIC_KEY">
      {children}
    </RealNodeProvider>
  );
}
```

Onderliggende initialisatievolgorde:

1. De Provider stuurt automatisch een `GET /sdk/config?api_key=pk_live_...` verzoek naar onze edge-servers.
2. De servers retourneren de actieve configuratie die is gekoppeld aan uw abonnement: het beschermingsniveau en de ingeschakelde functies.
3. Op basis van deze gegevens mount de SDK dynamisch de juiste modules zonder dat er code-wijzigingen aan uw kant nodig zijn:
 - **Insight:** Alleen passieve gedragsanalyse. Geen zichtbare UI-interferentie.
 - **Sentinel:** De volledige kinetische gedragsmatrix wordt op de achtergrond geactiveerd.
 - **Vault:** De kinetische matrix en de biometrische modal (FaceID, TouchID, Windows Hello) worden geactiveerd. De modal wordt automatisch in de DOM geïnjecteerd.
4. De SDK probeert stilzwijgend een bestaande sessie te herstellen (als de gebruiker al eerder tijdens hetzelfde evenementvenster een verificatie heeft voltooid) om overbodige biometrische prompts te voorkomen.

Veiligheid van de Public Key

Uw openbare sleutel (`pk_live_...`) kan veilig in uw openbare React-bundle verblijven. Het verleent geen toegang tot administratieve acties en dient uitsluitend om uw eigendom te identificeren bij onze verificatie-API. Uw geheime sleutel (`sk_live_...`) daarentegen moet volledig geïsoleerd blijven op uw backend-infrastructuur.

4 Stap 3: De Afrekenknop Beveiligen

Dit is de enige noodzakelijke wijziging binnen uw afrekenstroom. Vervang eenvoudig uw standaard HTML-aankoopknop door de `RealNodeCheckoutButton` component.

```
import { RealNodeCheckoutButton } from '@emkaylabs/realnode-sdk/react';

export default function CheckoutPage() {

  const handleSuccess = (proof) => {
    // proof bevat: { idh, device_hash, device_token, quantity, status }
    // Stuur deze payload naar uw backend om de bestelling af te ronden.
  }
}
```

```

    fetch('/api/uw-checkout-eindpunt', {
      method: 'POST',
      headers: { 'Content-Type': 'application/json' },
      body: JSON.stringify(proof)
    });
  };

  const handleDenied = () => {
    alert("Toegang Geweigerd: Verdachte automatisering gedetecteerd.");
  };

  return (
    <RealNodeCheckoutButton
      quantity={2}
      onSuccess={handleSuccess}
      onDenied={handleDenied}
      className="uw-bestaande-css-klasse"
    >
      Bestelling Bevestigen
    </RealNodeCheckoutButton>
  );
}

```

Uitvoeringsstroom na gebruikersklik:

1. De klik-gebeurtenis wordt onderschept. De transactie wordt tijdelijk gepauzeerd.
2. Afhankelijk van het actieve abonnement gedetecteerd door de Provider:
 - **Insight:** Een stille achtergrondverificatie wordt uitgevoerd. De gebruiker merkt niets.
 - **Sentinel:** De gedragsscore wordt berekend en geverifieerd. De gebruiker merkt niets.
 - **Vault:** De biometrische modal komt naar de voorgrond. De gebruiker verifieert zijn identiteit via FaceID, TouchID of Passkey. De hardware-operatie is in minder dan 1 seconde voltooid.
3. Bij succesvolle attestatie wordt **onSuccess** geactiveerd en wordt het cryptografische bewijs (**proof**) geïnjecteerd.
4. Als de attestatie mislukt of een botsignatuur wordt gedetecteerd, wordt **onDenied** geactiveerd en wordt de transactiereeks afgebroken.

Belangrijk: Biometrie wordt slechts eenmaal per sessie aangevraagd.

De SDK bewaart het attestatiebewijs in het geheugen. Als een gebruiker met succes een biometrische uitdaging heeft doorstaan (bijv. tijdens een eerdere aankoop), wordt er niet opnieuw om gevraagd voor latere transacties op dezelfde pagina, op voorwaarde dat de sessie geldig blijft.

5 Stap 4: Gebruikersinterface Componenten (Aanbevolen)

RealNode biedt kant-en-klare, aanpasbare componenten die naadloos opgaan in uw bestaande gebruikersinterface, zonder dat er wireframing of CSS-boilerplate nodig is.

5.1 4.1. Realtime Quotumbadge

Beschikbaar voor alle abonnementen. Deze badge toont de resterende ticketvoorraad die beschikbaar is voor de huidige gebruikerssessie. De numerieke waarde wordt automatisch gesynchroniseerd met onze edge-servers via een persistente Server-Sent Events (SSE) verbinding.

```
import { RealNodeQuotaBadge } from '@emkaylabs/realnode-sdk/react';

// Plaats deze component naast de afrekenknop.
// Rendered automatisch: "Resterende tickets: 3 / 5"
<RealNodeQuotaBadge label="Resterende tickets" />
```

5.2 4.2. Gebruikersdashboard — Apparaatbeheer

Deze component injecteert een compleet, interactief dashboard in het gebied "Mijn account" van uw gebruikers, waarmee ze het volgende kunnen doen:

- Controleren van de lijst met geautoriseerde hardwareapparaten (smartphones, laptops) die aan hun identiteit zijn gekoppeld.
- Toegang op afstand intrekken voor een verloren of gestolen apparaat.
- Hardwarebindingen ontkoppelen voorafgaand aan de doorverkoop van apparaten.
- Accounttoegang veilig overdragen aan een andere gebruiker.
- Toegang herstellen met behulp van een Master Code voor noodgevallen, verstrekt door uw beheerteam.

```
import { RealNodeDeviceManager } from '@emkaylabs/realnode-sdk/react';

export default function MyAccount() {
  return (
    <div>
      <h2>Beveiligingsinstellingen</h2>
      { /* Genereert automatisch de volledige beheerinterface. */ }
      { /* Geen extra HTML- of CSS-logica vereist. */ }
      <RealNodeDeviceManager apiKey="pk_live_UW_PUBLIC_KEY" />
    </div>
  );
}
```

6 Stap 5: Laatste Server-side Validatie (Backend)

Het cryptografische bewijs (proof) dat door de browser wordt uitgegeven, moet door uw backend-infrastructuur worden gevalideerd voordat de bestelling wordt uitgevoerd. Dit is een verplicht controlepunt: uw server voert een geverifieerde oproep uit naar de RealNode API met behulp van uw **Secret Key** om te garanderen dat er niet met het bewijs is geknoeid en dat de inventarisquota geldig blijven.

```
// Voorbeeld: Node.js / Express backend
app.post('/api/uw-checkout-eindpunt', async (req, res) => {
  const proof = req.body;
  // proof = { idh, device_hash, device_token, quantity, status }

  const response = await fetch('https://api.emkaylabs.tech/consume', {
    method: 'POST',
    headers: {
      'Content-Type': 'application/json',
      'X-RealNode-API-Key': 'pk_live_UW_PUBLIC_KEY',
      'X-RealNode-Secret-Key': 'sk_live_UW_SECRET_KEY'
    },
    body: JSON.stringify({
      quantity: proof.quantity,
      idh: proof.idh,
      device_hash: proof.device_hash,
      device_token: proof.device_token,
      client_id: 'uw-website-id',
      // event_id: 'CONCERT-2024-AMSTERDAM' // Optioneel: voor
      // evenementenspecifieke quota
    })
  });

  if (response.ok) {
    // RealNode heeft de attestatie succesvol gevalideerd. Ga door met de
    // betaling.
    await processPayment();
    res.status(200).json({ success: true });
  } else {
    // Quotum overschreden, of ongeldige cryptografische handtekening.
    res.status(403).json({ error: "Transactie geweigerd door RealNode
    Engine." });
  }
});
```

Standaard /consume API Reacties:

HTTP-code	Betekenis	Aanbevolen actie
200	Transactie gevalideerd, quotum afgetrokken	Rond de bestelling af
403	Ongeldige handtekening of bot gedetecteerd	Weiger de transactie
429	Quotum uitgeput voor deze gebruiker IDH	Toon een afkoelbericht
401	Ongeldige API-sleutel inloggegevens	Controleer uw sleutels in het dashboard

Toegewijde B2B Ondersteuning

Als zakelijke klant van RealNode ontvangt u prioritaire integratieondersteuning. Voor architectonische vragen kunt u contact opnemen met ons engineeringteam via security@emkaylabs.tech. We garanderen een technisch antwoord binnen 24 kantooruren.

7 Bijlage: Geavanceerde Configuraties

7.1 Evenement-specifiek Quotumbeheer (Vault Plan)

Voor sterk gecontroleerde drops (limited edition tickets, exclusieve sneaker releases, enz.), kunt u een interne evenement-identificatie toevoegen aan de `/consume` payload (zie Stap 5).

Hoe werkt het?

- Er is geen voorafgaande configuratie nodig in het RealNode-dashboard.
- Voeg eenvoudig uw eigen willekeurige string toe (bijv. `CONCERT-2024-AMSTERDAM`).
- De RealNode Engine zal automatisch hardwarequota voor die specifieke gebruiker isoleren en volgen, strikt binnen de context van dat evenement.