

# Contents

<b>AUGMENT E-SCOOTER BACKEND - PRÆCIS KLONINGS PROMPT</b>	<b>1</b>
OVERSIGT . . . . .	1
1. TEKNISK STACK . . . . .	1
2. API STRUKTUR . . . . .	2
3. DATABASE SCHEMA . . . . .	3
4. AWS COGNITO KONFIGURATION . . . . .	5
5. CHARGESEE INTEGRATION . . . . .	6
6. BLUETOOTH DEVICE PROTOCOL . . . . .	6
7. WEBSOCKET STREAM PROTOCOL . . . . .	7
8. REFERRAL SYSTEM . . . . .	8
9. FIRMWARE OTA UPDATES . . . . .	8
10. MULTI-LANGUAGE SUPPORT . . . . .	9
11. PUSH NOTIFICATIONS . . . . .	9
12. ANALYTICS & MONITORING . . . . .	9
13. ENVIRONMENT VARIABLES . . . . .	10
14. SECURITY CONSIDERATIONS . . . . .	11
15. DEPLOYMENT ARCHITECTURE . . . . .	11
16. IMPLEMENTATION CHECKLIST . . . . .	11
17. API RATE LIMITS . . . . .	12
18. SAMPLE GRAPHQL REQUESTS . . . . .	12
19. ERROR CODES . . . . .	13
20. TESTING ENDPOINTS . . . . .	14

# AUGMENT E-SCOOTER BACKEND - PRÆCIS KLONINGS PROMPT

## OVERSIGT

Denne prompt indeholder alle tekniske detaljer til at klone backend'en for Augment e-scooter sharing platformen, baseret på analyse af den Android APK.

---

## 1. TEKNISK STACK

### Backend Infrastructure

- **API Type:** GraphQL API (primary) + REST endpoints
- **Hosting:** AWS Infrastructure
  - AWS Lambda (serverless functions)
  - AWS API Gateway
  - AWS Cognito (authentication)
- **Database:** PostgreSQL via Supabase
  - PostgREST API layer
- **Real-time:** WebSocket server på port 8969 (/stream endpoint)
- **CDN/Hosting:** Netlify (staging: <https://staging--augment-escoot.netlify.app>)

### Tredjeparts Services

- **Payment Processing:** Chargebee
- **Analytics/Monitoring:**
  - Sentry (<https://o447951.ingest.sentry.io/api/4509632503087104/envelope/>)
  - Firebase Analytics

- **Customer Support:** Intercom
- **Notifications:**
  - Firebase Cloud Messaging
  - OneSignal

## Authentication & Security

- AWS Cognito Identity & User Pools
  - OAuth2 flow med refresh tokens
  - MFA support (SMS, TOTP, EMAIL\_OTP)
  - WebAuthn/Passkey support
- 

## 2. API STRUKTUR

### GraphQL API Endpoint

Main Production: <https://frbc72oc4h.execute-api.eu-west-1.amazonaws.com/prod/graphql-public>  
 Staging: <https://staging--augment-escoot.netlify.app>

### WebSocket Stream Endpoint

Development: <http://localhost:8969/stream>  
 Production: [wss://\[domain\]:8969/stream](wss://[domain]:8969/stream)

### Primære GraphQL Queries

#### User Management

```
query MeAccountDetails
query MeCountry
query MeFirstPromoter
query MeSubscriptions
query MeInvoices($country: CountryCode!, $language: Language!)
query MePaymentSources
query MePayNow
query MeManagePayment
query MeDevices
query IntercomHexCodes
```

#### Device Management

```
query CheckDeviceOwnership($input: CheckDeviceOwnershipInput!)
query DeviceQuery($input: DeviceQueryInput!)
query DeviceSubUsers($input: DeviceQueryInput!)
query DeviceSettingsOperationCode($btMac: MACHX!)
query FirmwareUpgrade($firmware: CheckFirmwareUpgradeInput!, $language: Language)
query GetCertificationStandardChangeCode($btMac: MACHX!, $certificationStandard: CertificationStandard!)
```

#### System

```
query MandatoryMobileUpdate($input: CheckMandatoryMobileAppStoreUpgrade!)
```

### Primære GraphQL Mutations

#### Device Operations

```

mutation ClaimDeviceOwnership($input: ClaimDeviceOwnershipInput!)
mutation ResetDeviceOwnership($input: ResetDeviceOwnershipInput!)
mutation RenameDevice($input: RenameDeviceInput!)
mutation ChangeDeviceIdentifiers($input: AdminChangeDeviceIdentifiersInput!)
mutation ChangeDeviceSharingSettings($input: ChangeDeviceSharingSettingsInput!)
mutation AddDeviceStatus($input: AddDeviceStatusInput!)

```

## Sharing

```

mutation ShareDeviceOtp($input: GenerateDeviceAccessOtpInput!)
mutation ConsumeShareDeviceOtp($input: ConsumeShareDeviceOtpInput!)
mutation RevokeAccessToOwnersDevice($input: RevokeAccessToOwnersDeviceInput!)

```

## Payment

```

mutation AssignPaymentSourceRole($input: AssignChargebeePaymentSourceRoleInput!)
mutation ChargebeeInvoicePdf($createChargebeeInvoicePdfInput: CreateChargebeeInvoicePdfInput!)

```

## GraphQL Fragments

```

fragment DeviceFragment on Device
fragment DeviceUserFragment on DeviceUser
fragment ChargebeeHostedPage on User
fragment PaymentSourceFragment on ChargebeePaymentSource
fragment PaymentSourceCardFragment on ChargebeePaymentSourceCard
fragment PaymentSourceApplePayFragment on ChargebeePaymentSourceApplePay
fragment PaymentSourceGooglePayFragment on ChargebeePaymentSourceGooglePay
fragment PaymentSourcePayPalFragment on ChargebeePaymentSourcePayPal
fragment PaymentSourceDirectDebitFragment on ChargebeePaymentSourceDirectDebit
fragment PaymentSourceGenericFragment on ChargebeePaymentSourceGeneric
fragment FirstPromoterFragment on FirstPromoterPromoter
fragment MutationResponseFragment on MutationResponse
fragment OrderItem on OrderItem

```

---

## 3. DATABASE SCHEMA

### Primære Tabeller (PostgreSQL/Supabase)

#### users

- **id** (uuid, **primary key**)
- **cognito\_username** (**varchar**)
- **email** (**varchar**, **unique**)
- **phone\_number** (**varchar**)
- **country\_code** (**varchar**)
- **language** (**varchar**)
- **created\_at** (**timestamp**)
- **updated\_at** (**timestamp**)
- **first\_promoter\_id** (**varchar**, nullable)
- **chargebee\_customer\_id** (**varchar**, nullable)
- **intercom\_hash** (**varchar**, nullable)
- **preferred\_mfa\_setting** (**varchar**, nullable)

## devices

```
- id (uuid, primary key)
- serial_number (varchar, unique, not null)
- bt_mac (varchar, unique) -- Bluetooth MAC address
- owner_id (uuid, foreign key -> users.id)
- device_name (varchar)
- firmware_version (varchar)
- hardware_version (varchar)
- certification_standard (varchar)
- created_at (timestamp)
- updated_at (timestamp)
- last_connected_at (timestamp, nullable)
- total_km (decimal)
- sharing_enabled (boolean, default false)
```

## device\_users (sharing/access control)

```
- id (uuid, primary key)
- device_id (uuid, foreign key -> devices.id)
- user_id (uuid, foreign key -> users.id)
- access_level (varchar) -- 'OWNER', 'SHARED', 'TEMPORARY'
- granted_at (timestamp)
- expires_at (timestamp, nullable)
- granted_by (uuid, foreign key -> users.id)
```

## device\_sharing\_otpss

```
- id (uuid, primary key)
- device_id (uuid, foreign key -> devices.id)
- otp_code (varchar, unique)
- created_by (uuid, foreign key -> users.id)
- created_at (timestamp)
- expires_at (timestamp)
- consumed (boolean, default false)
- consumed_at (timestamp, nullable)
- consumed_by (uuid, foreign key -> users.id, nullable)
```

## subscriptions (Chargebee integration)

```
- id (uuid, primary key)
- user_id (uuid, foreign key -> users.id)
- chargebee_subscription_id (varchar, unique)
- status (varchar) -- 'ACTIVE', 'CANCELLED', 'PAUSED'
- plan_id (varchar)
- current_term_start (timestamp)
- current_term_end (timestamp)
- next_billing_at (timestamp, nullable)
- created_at (timestamp)
- updated_at (timestamp)
```

## invoices

```
- id (uuid, primary key)
- user_id (uuid, foreign key -> users.id)
- chargebee_invoice_id (varchar, unique)
```

```
- amount (decimal)
- currency (varchar)
- status (varchar) -- 'PAID', 'PENDING', 'FAILED'
- invoice_date (timestamp)
- due_date (timestamp)
- pdf_url (varchar, nullable)
- created_at (timestamp)
```

#### device\_status\_logs

```
- id (uuid, primary key)
- device_id (uuid, foreign key -> devices.id)
- battery_level (integer) -- 0-100
- speed_limit (integer) -- km/h
- total_voltage_mv (integer)
- cell_voltages (jsonb) -- array of cell voltages
- temperature (integer, nullable)
- logged_at (timestamp)
- trip_start_km (decimal, nullable)
```

#### firmware\_versions

```
- id (uuid, primary key)
- version (varchar, unique)
- hardware_compatibility (varchar[] )
- release_notes (jsonb)
- binary_url (varchar)
- checksum (varchar)
- released_at (timestamp)
- mandatory (boolean, default false)
```

---

## 4. AWS COGNITO KONFIGURATION

### User Pool Attributes

- email (required, verifiable)
- phone\_number (verifiable)
- preferred\_username
- custom:country\_code
- custom:language
- custom:chargebee\_customer\_id
- custom:first\_promoter\_id

### MFA Configuration

- Optional MFA
- Supported methods:
  - SMS
  - TOTP (Time-based One-Time Password)
  - EMAIL OTP

### Password Policy

- Minimum length: 8 characters

- Requires: lowercase, uppercase, numbers
- Special characters optional

## OAuth2 Flows

- Authorization Code flow
  - Refresh token rotation enabled
  - Token expiry: Access (1 hour), Refresh (30 days)
- 

## 5. CHARGEBEE INTEGRATION

### Webhook Events

```
[  
  'subscription_created',  
  'subscription_cancelled',  
  'subscription_renewed',  
  'subscription_changed',  
  'invoice_generated',  
  'payment_succeeded',  
  'payment_failed',  
  'customer_created'  
]
```

### Payment Sources

- Credit/Debit Card
- Apple Pay
- Google Pay
- PayPal
- SEPA Direct Debit

### Product/Plans

- Subscription-based model
  - Monthly billing cycles
  - Multi-country pricing (EUR, DKK, etc.)
- 

## 6. BLUETOOTH DEVICE PROTOCOL

### BLE Services & Characteristics

#### Main Service UUID

00006880-0000-1000-8000-00805f9b34fb

#### Characteristics

```
// OTA Firmware Update  
OTA_WRITE_NO_RESPONSE_UUID: "00006881-0000-1000-8000-00805f9b34fb"  
OTA_INDICATE_UUID: "00006882-0000-1000-8000-00805f9b34fb"
```

```
// Device Control  
DEVICE_CONTROL_UUID: "00006883-0000-1000-8000-00805f9b34fb"
```

```
// Device Status (notifications)
DEVICE_STATUS_UUID: "00006884-0000-1000-8000-00805f9b34fb"
```

## Device Commands

- Lock/Unlock
- Get Status (battery, speed, voltage, etc.)
- Set Speed Limit
- Get/Set Certification Standard
- Firmware Upgrade
- Factory Reset (REDI mode)

## Status Data Format

```
{
  battery_level: number, // 0-100
  speed_limit: number, // km/h
  total_voltage_mv: number,
  cell_voltages: number[], // Individual cell voltages
  trip_start_km: number,
  total_km: number,
  motor_resistance_ohm: number,
  temperature: number // optional
}
```

---

## 7. WEBSOCKET STREAM PROTOCOL

### Connection

```
// Development
ws://localhost:8969/stream
```

```
// Production
wss://[domain]:8969/stream
```

### Message Types

```
{
  // Device location updates
  type: 'LOCATION_UPDATE',
  payload: {
    device_id: string,
    latitude: number,
    longitude: number,
    timestamp: string
  }
}

{
  // Device status updates
  type: 'STATUS_UPDATE',
  payload: {
    device_id: string,
```

```

    battery_level: number,
    is_locked: boolean,
    timestamp: string
  }
}

{
  // Firmware upgrade progress
  type: 'FIRMWARE_PROGRESS',
  payload: {
    device_id: string,
    progress: number, // 0-100
    status: 'IN_PROGRESS' | 'COMPLETED' | 'FAILED'
  }
}

```

---

## 8. REFERRAL SYSTEM

### URL Format

[https://augment.eco?ref=\[REFERRAL\\_CODE\]](https://augment.eco?ref=[REFERRAL_CODE])

### Share Message (Multi-language)

Danish: "Je krijgt 5% korting als je lid wordt van augment.eco met mijn verwijzingscode"

Italian: "Riceverai uno sconto del 5% quando ti iscrivi a augment.eco con il mio codice di riferimento"

German: "Du erhältst 5% Rabatt, wenn du dich mit meinem Empfehlungscode bei augment.eco anmeldest"

### First Promoter Integration

- Track referrals
  - Discount application
  - Commission tracking
- 

## 9. FIRMWARE OTA UPDATES

### Update Flow

1. Check for available firmware via GraphQL query
2. Download firmware binary from S3
3. Verify checksum
4. Connect to device via BLE
5. Write firmware chunks to OTA characteristic
6. Monitor progress via indications
7. Device reboots with new firmware

### Binary Format

- Encrypted firmware binary
  - Chunk size: 244 bytes
  - Protocol: Nordic DFU or custom protocol
-

## 10. MULTI-LANGUAGE SUPPORT

### Supported Languages

- English (en)
- Danish (da)
- German (de)
- Italian (it)
- Dutch (nl)
- Spanish (es)

### i18n Keys Structure

```
{  
  "onboarding": { ... },  
  "device": { ... },  
  "payment": { ... },  
  "settings": { ... },  
  "errors": { ... },  
  "notifications": { ... }  
}
```

---

## 11. PUSH NOTIFICATIONS

### Providers

- Firebase Cloud Messaging (Android/iOS)
- OneSignal (fallback/additional)

### Notification Types

- Device unlocked/locked
- Battery low warning
- Payment successful/failed
- Firmware update available
- Sharing request
- Mandatory app update

---

## 12. ANALYTICS & MONITORING

### Sentry Configuration

```
{  
  dsn: "https://c1dfb07d783ad5325c245c1fd3725390@o447951.ingest.sentry.io/4509632503087104",  
  environment: "production",  
  tracesSampleRate: 1.0,  
  profilesSampleRate: 1.0,  
  enableNative: true,  
  enableAutoPerformanceTracing: true  
}
```

### Tracked Events

- API errors

- BLE connection failures
  - Payment errors
  - App crashes
  - Performance metrics (FCP, LCP, CLS)
- 

## 13. ENVIRONMENT VARIABLES

```
# AWS
AWS_REGION=eu-west-1
AWS_COGNITO_USER_POOL_ID=[pool_id]
AWS_COGNITO_CLIENT_ID=[client_id]
AWS_COGNITO_IDENTITY_POOL_ID=[identity_pool_id]

# API
GRAPHQL_API_ENDPOINT=https://frbc72oc4h.execute-api.eu-west-1.amazonaws.com/prod/graphql-public
WEBSOCKET_ENDPOINT=wss://[domain]:8969/stream

# Supabase
SUPABASE_URL=[your_supabase_url]
SUPABASE_ANON_KEY=[your_anon_key]
SUPABASE_SERVICE_ROLE_KEY=[your_service_role_key]

# Chargebee
CHARGEBEE_SITE=[site_name]
CHARGEBEE_API_KEY=[api_key]
CHARGEBEE_WEBHOOK_SECRET=[webhook_secret]

# Firebase
FIREBASE_API_KEY=[api_key]
FIREBASE_PROJECT_ID=[project_id]
FIREBASE_MESSAGING_SENDER_ID=[sender_id]

# Sentry
SENTRY_DSN=https://c1dfb07d783ad5325c245c1fd3725390@o447951.ingest.sentry.io/4509632503087104

# Intercom
INTERCOM_APP_ID=[app_id]
INTERCOM_API_KEY=[api_key]

# OneSignal
ONESIGNAL_APP_ID=[app_id]

# First Promoter
FIRST_PROMOTER_API_KEY=[api_key]
FIRST_PROMOTER_TID=[tracking_id]

# S3 Storage
AWS_S3_BUCKET_FIRMWARE=[bucket_name]
AWS_S3_BUCKET_ASSETS=[bucket_name]
```

---

## 14. SECURITY CONSIDERATIONS

### API Security

- JWT tokens via Cognito
- Rate limiting on API Gateway
- CORS configuration for allowed origins
- Request signing for sensitive operations

### Data Encryption

- TLS 1.2+ for all connections
- Encrypted firmware binaries
- PII encryption in database
- Secure BLE pairing

### Compliance

- GDPR compliant
  - Data retention policies
  - Right to deletion implementation
  - Privacy policy enforcement
- 

## 15. DEPLOYMENT ARCHITECTURE

### AWS Lambda Functions

- graphql-api (main GraphQL resolver)
- auth-triggers (Cognito pre/post hooks)
- chargebee-webhooks
- firmware-processor
- device-status-processor
- notification-sender

### Database Migrations

- Use Supabase migrations
- Version controlled SQL files
- Rollback procedures

### CI/CD Pipeline

- GitHub Actions / GitLab CI
  - Automated testing
  - Staging → Production promotion
  - Database migration automation
- 

## 16. IMPLEMENTATION CHECKLIST

### Phase 1: Infrastructure Setup

- Set up AWS account and services
- Create Cognito User Pool
- Set up Supabase PostgreSQL database

- Configure S3 buckets
- Set up API Gateway

#### Phase 2: Core Backend

- Implement GraphQL schema
- Create database tables and migrations
- Implement authentication flow
- Build device management APIs
- Create sharing/OTP system

#### Phase 3: Integrations

- Integrate Chargebee payments
- Set up Firebase/OneSignal notifications
- Configure Sentry monitoring
- Implement Intercom support
- Set up First Promoter referrals

#### Phase 4: Real-time & BLE

- Build WebSocket server
- Implement device protocol handlers
- Create firmware OTA system
- Build status streaming

#### Phase 5: Testing & Deployment

- Unit tests
  - Integration tests
  - Load testing
  - Security audit
  - Production deployment
- 

## 17. API RATE LIMITS

Authenticated requests: 1000/hour per user

Anonymous requests: 100/hour per IP

WebSocket connections: 5 concurrent per user

Firmware downloads: 10/day per device

---

## 18. SAMPLE GRAPHQL REQUESTS

### Claim Device Ownership

```
mutation ClaimDevice {
  ClaimDeviceOwnership(input: {
    serialNumber: "ABC123456"
    btMac: "AA:BB:CC:DD:EE:FF"
  }) {
    success
    message
    device {

```

```

        id
        deviceName
        firmwareVersion
    }
}
}

```

### Generate Sharing OTP

```

mutation ShareDevice {
  ShareDeviceOtp(input: {
    deviceId: "uuid-here"
    expiresInHours: 24
  }) {
    success
    otpCode
    shareUrl
    expiresAt
  }
}

```

### Get Device Status

```

query GetDevice {
  DeviceQuery(input: { deviceId: "uuid-here" }) {
    id
    deviceName
    serialNumber
    firmwareVersion
    lastStatus {
      batteryLevel
      speedLimit
      totalKm
      lastConnectedAt
    }
  }
}

```

---

## 19. ERROR CODES

```

{
  // Authentication
  'AUTH_001': 'Invalid credentials',
  'AUTH_002': 'MFA required',
  'AUTH_003': 'Token expired',
  'AUTH_004': 'User not verified',

  // Device
  'DEV_001': 'Device not found',
  'DEV_002': 'Device already claimed',
  'DEV_003': 'Invalid serial number',
  'DEV_004': 'Bluetooth connection failed',
  'DEV_005': 'Firmware upgrade failed',
}

```

```
// Payment
'PAY_001': 'Payment failed',
'PAY_002': 'Invalid payment method',
'PAY_003': 'Subscription inactive',

// Sharing
'SHR_001': 'Invalid OTP',
'SHR_002': 'OTP expired',
'SHR_003': 'Sharing not enabled'
}
```

---

## 20. TESTING ENDPOINTS

### Health Check

GET /health  
Response: { "status": "ok", "timestamp": "2024-01-01T00:00:00Z" }

### API Version

GET /version  
Response: { "version": "1.0.0", "graphql": "16.6.0" }

---

Dette dokument indeholder alle nødvendige detaljer til at klone Augment e-scooter platformens backend. For yderligere detaljer om specifik implementering, referer til den originale APK eller kontakt udviklerne.

**Generated:** 2024-12-25 **Based on:** Augment - Kopi.apk (Android app version detected in analysis)