

Développeur systèmes senior, 7 ans en C sur Linux bare-metal, kernel drivers et AOSP. Architecte d'une plateforme cloud gaming en production (kernel, conteneurs AOSP, backend Rust temps réel, fleet GPU bare-metal) servant plusieurs centaines de joueurs simultanés. Collaboration technique directe avec Canonical (Anbox) et Nvidia. En recherche d'un CDI systèmes ou embarqué.

EXPÉRIENCE PROFESSIONNELLE

ZenQuality

Avr. 2026 - présent

Développeur indépendant • Yerres • zenquality.fr

- ▶ Mission SEO et conformité légale RGPD pour PME service (Île-de-France) — audit technique Core Web Vitals + Schema.org + NAP, refonte CGV B2B/B2C, RGPD, mentions légales, mise en conformité médiateur CM2C. Plan d'action 12 sprints.
- ▶ Site vitrine WordPress ([Gutenverse](#)) pour PME esthétique — conception, intégration, déploiement et support continu (hébergement client).
- ▶ Setup et auto-hébergement de l'infra ZenQuality sur VPS [Scaleway](#) — stack [Astro](#) / [React](#) / [PHP 8](#) / [PostgreSQL](#) conteneurisée [Docker](#), déploiement automatisé, pipeline de production complet.

CareGame

Mars 2019 - Mars 2025

Développeur logiciel — Systèmes & Backend • Paris • Full remote dès 2020

- ▶ Conception et maintenance de modules kernel Linux en [C](#) (x86 / ARM) — drivers d'interface hôte/conteneur orchestrant la communication bidirectionnelle entre Linux hôte et instances AOSP conteneurisées.
- ▶ Conception de l'isolation CPU/GPU par session sur la fleet bare-metal — adaptation de modules GPU Nvidia, partitionnement 2 cœurs/session (physiques et émulés), sérialisation des accès concurrents sur zones GPU partagées. Résultat : passage à 32 sessions AAA stables/serveur.
- ▶ Co-développement du backend [Rust](#) orchestrant le cycle de vie des conteneurs AOSP — communication WebSocket temps réel clients ↔ instances, intégration [Docker](#) et [LXC](#), scheduling des sessions sur la fleet GPU.
- ▶ Développement de virtual input devices AOSP en [Java](#) (touchscreen, gamepad) — pipeline complet inputs frontend → backend Rust → drivers hôtes → injection AOSP, optimisé pour latence temps réel gameplay AAA.
- ▶ Architecture et exploitation de fleet GPU bare-metal [AWS g4dn.metal](#) (8× GPU T4, 64 vCPU, ~20 serveurs en pic) — isolation 2 cœurs CPU/session, ramdisk I/O, 32 sessions AAA simultanées par serveur. Plusieurs centaines de joueurs servis en parallèle (Asphalt 9 : 3 sessions / GPU T4).
- ▶ Reprise et hardening d'un PoC [LXD](#) + [Docker](#) issu d'une R&D Nvidia (intégration côté Docker principalement) — adaptation à la production cloud gaming, débogage kernel/conteneur, performance validée par les équipes Nvidia comme dépassant le scope initial du PoC.
- ▶ Collaboration technique directe en anglais avec Canonical ([Anbox](#) + builds LXC/LXD non commerciaux, remontée bugs et feature requests) et Ampere Computing (intégration et benchmark de serveurs ARM pré-commerciaux pour évaluation de migration de fleet).
- ▶ Pipeline d'installation automatique de jeux AOSP et persistance des sauvegardes utilisateur — fusion Android Backup natif + scripts custom pour couvrir les cas non gérés nativement (DRM, données externes).
- ▶ Développement d'un outil d'orchestration [Bash](#) modulaire (1000+ lignes) pour la gestion du cycle de vie des conteneurs et l'intégration en production — parsing avancé des arguments Docker, préparation et lancement automatisé des conteneurs AOSP, brique appelée par la CI et le backend Rust.

Développeur C — Système embarqué · Ivry-sur-Seine
Stage 42 (6 mois) puis CDD (4 mois)

- Développement du logiciel embarqué en `C` pour boîtier connecté basé Orange Pi (`Debian ARM`) — interception du flux `ESC/POS` d'une imprimante thermique, génération PNG du ticket, transfert WiFi direct vers application mobile à la connexion.
- Intégration matérielle : `GPIO` physique (bouton + timeout), hotspot WiFi embarqué avec diffusion des credentials via `NFC` pour appairage automatique.

PROJETS & RÉALISATIONS

Code source & projets persos — serveur Git auto-hébergé en production en continu

Configuration Claude Code, dotfiles, projets bas-niveau (42, expérimentations C/Rust) — accessibles publiquement. Mirror automatique vers GitHub via push hook.
git.bchanot.fr/bchanot

HomeLab — Infrastructure personnelle en continu

Auto-hébergement Git (`Gogs`) / DNS / VPN / SMB — NAS Asustor, Freebox `WireGuard` VPN site-to-site, Pi-hole, segmentation réseau, hardening `fail2ban`, `gocryptfs` sur dossiers sensibles.

FORMATION

École 42 2015 - 2019

Programmation informatique · Clichy

Systèmes / Kernel : ft_linux & kfs-1 (Linux From Scratch — bootloader ASM, GDT, interruptions, driver char device) · drivers & interrupt · process & memory · little penguin
Bas niveau : malloc (allocateur mémoire) · nm (parsing ELF) · ft_ls · ft_select · 42sh (shell POSIX complet)
Réseau / Sécurité : ft_ssl_md5 (crypto) · snow crash (exploitation système) · lem-in · push-swap · doctor quine

Next Formation 2013 - 2015

TSRIT — Technicien Supérieur Réseaux & Télécoms, Félicitations du jury · Vincennes

COMPÉTENCES TECHNIQUES

Langages	C · Rust · Bash · Python · Java (AOSP/Android)
Embarqué	Linux kernel drivers · AOSP · ARM / x86 · GPIO · NFC · ESC/POS · Orange Pi · cross-compilation GCC
Conteneurs	Docker · LXC / LXD · QEMU · cgroups · namespaces
Backend	Rust · WebSocket · architecture GPU bare-metal · ramdisk I/O
Systèmes	Linux bare-metal · AOSP · Android Backup · systemd · SELinux
Cloud / Infra	AWS (EC2, g4dn bare-metal, IAM, S3, CloudWatch) · Scaleway VPS · OVH / Hetzner · Nginx · Apache · Let's Encrypt
DevOps	Git · GitHub Actions · GitLab · CI/CD
IA / Outils	Claude Code (agents/skills custom) · N8N · automatisation
Familier avec	C++

LANGUES

Anglais C2 Espagnol B1 Français natif

CENTRES D'INTÉRÊT

Voyage Sciences Astronomie
Nature Jeux vidéo Musique