

Clearpoint Installer V2 – Design (Ubuntu 22.04 + Cloudflared + GUI)

מטרת המסמך

עבור (Mini PC) עבור תהליך התקנה מודרני, פשוט, ו"מקצועי" למחשב הלקוח (Design Spec) מסמך זה מגדיר תכנון Clearpoint.

- **מטרה עסקית:** לאפשר התקנה בשטח ע"י טכנאי/מתקין חיצוני בצורה מהירה וברורה.
- במקום רצף פקודות ידני (שירותים + Wizard) "**מטרה טכנית:** להפוך את ההתקנה ל"מוצר".
- **הגבלה:** מסמך זה **לא משנה כלום בקוד הקיים**. הוא מפרט את היעד ואת התכנון לגרסה הבאה.

Scope

In-scope (בגרות V2)

- Installer גרפי (Wizard) ל-Ubuntu 22.04.
- בצורה אוטומטית (RAM disk / שירותים / תיקיות / Cloudflared / FFmpeg) התקנה של תלותים נדרשים.
- reboot כך שהמערכת תעלה לבד אחרי (systemd) הגדרת שירותים.
- בצורה מודרכת (כולל בדיקות) Cloudflared tunnel + DNS יצירת/ניהול.
- "יצירת קונפיג מקומי מסודר במקום" קבצים מפוזרים.
- מסך סיכום + (health checks) בדיקות בסיום התקנה.

Out-of-scope (בשלב זה)

- . scripts/utils/* - שינוי/כתיבה מחדש של הלוגיקה הקיימת ב
- (V3 זה יכול להיות) אחד "Agent" ריפקטור מלא ל
- DB שינוי סכימת.

UX (Installer GUI) הגדרת יעד

שפה

- עם אפשרות לאנגלית בעתיד, (RTL) ברירת מחדל: עברית.

Flow של ה Wizard

1. Welcome

- הסבר קצר מה יותקן.
- sudo דרישת הרשאות.

2. פרטי לקוח

- שם לקוח / מזהה לקוח
- tunnel subdomain (למשל customername.clearpoint.co.il)

3. הגדרות מערכת

- (אופציונלי) Auto-login האם להגדיר
- Ubuntu 22.04 אימות שיש

4. קבצי מצלמות (מוכנים מראש)

- לא מייצר קבצי מצלמה Installer-ה.
- נוצרים מראש מתוך מערכת הניהול camera-*.service ו/או camera-*.sh הקבצים (/admin/cameras) USB-ומועברים ל.
- לפני המשך USB-מזהה/מאמת שנמצאו קבצי מצלמות ב Installer-ה.

5. Cloudflare Tunnel-חיבור ל

- Login (cloudflared tunnel login)
- Create tunnel
- config יצירת
- cloudflared כ-service התקנת

6. Install / Apply

- יצירת תיקיות
- RAM disk /mnt/ram-ts יצירת
- runtime העתקת קבצי
- USB-העתקת קבצי מצלמות מה
- systemd services יצירת
- CRON (עם cron אם נשארם עם) הגדרת

7. Verification

- localhost:8080 עולה על Live server
- stream.m3u8 קבצי נוצרים לכל מצלמה
- cloudflared service פעיל

- חיצוני URL בדיקת
- uploader “dry run” (עם TEST_MODE ללא העלאה/או עם)

8. Finish

- סטטוסים, איפה הלוגים, URL: סיכום
- “כפתור: העתק כתובת”

רכיבים (Components)

1) Runtime (בשטה)

ניתן להישאר עם המנגנון הקיים, אבל לארוז אותו בצורה מסודרת V2-ב:

- scripts/utils/live-server.js
- scripts/utils/uploadVods.ts
- scripts/utils/status-check.sh
- scripts/utils/disk-check.sh
- camera-*.sh
- camera-*.service (אופציונלי)

סביבם cron/ההתקנה תשכפל/תמקם אותם במיקום אחיד ומוגדר, ותיצור שירותים.

2) Installer GUI

Progress של ההתקנה, ומדווחת state מנהלת, sudo אפליקציה שמריצה פעולות מערכת עם.

אפשרויות טכנולוגיות (לבחירה בהמשך):

- **Electron** (GUI דורש; לפיתוח; קל ומהיר לפיתוח; bundling קל)
- **Tauri** (Rust קל יותר, אבל מורכב יותר סביב)
- **Python + GTK** ("נפוץ בלינוקס; פחות "יובי")

והתנהגות UX המסמך לא מקבע טכנולוגיה –רק.

מבנה קבצים מומלץ על המחשב

להחליף “פיזור” בסטנדרט:

- /opt/clearpoint/
 - bin/
 - scripts/ (קבצים שלך)
 - logs/
- /etc/clearpoint/config.yml
- /var/lib/clearpoint/ (state מקומי אם צריך)
- /mnt/ram-ts (tmpfs)

קונפיג (Config) — config.yml

דוגמה (תכנון):

- customer :
 - name
 - tunnel_subdomain
- cameras :
 - id (camera uuid)
 - name
 - rtsp_url
 - username / password (אם נדרש)
- paths :
 - recordings_root
 - ram_ts_root
- cloudflared :
 - tunnel_name
 - credentials_path

ומועתקים Admin-קבצי המצלמות מסופקים מראש מה config-מתוך ה camera-*.sh לא מייצר Installer ה-V2-ב-USB.

systemd services (תכנון)

"מטרה: לא להיות תלויים ב"הרצה ידנית

שירותים מוצעים:

- clearpoint-cameras.target (אופציונלי)
- clearpoint-live-server.service (מריץ node live-server.js)
- clearpoint-status-check.service + timer (cron במקום) עם cron או להישאר עם cron ב-V2
- clearpoint-uploader.service + timer (cron במקום) או cron
- cloudflared.service (כפי שמסמך ההתקנה כבר מציע)

טובים logging/בהמשך כדי לקבל ניטור systemd timers - אבל מומלץ לעבור ל cron אפשר להשאיר ב-V2.

Cloudflared + DNS

עושה installer מה

- cloudflared tunnel login
- בשם הלקוח tunnel יצירת
- /etc/cloudflared/config.yml כתיבת
- cloudflared service install

DNS

או פעולה ידנית API דורש הרשאות Cloudflare ב-DNS record ה-

אפשרויות:

- **V2 (פשוט)**: copy buttons עם, (Name/Target)-DNS המסך מציג למתקין מה להוסיף ב:
- **V3 (אוטומטי)**: CNAME כדי ליצור Cloudflare API token שימוש ב:

החלטות מוצר – (Security) אבטחה

(Known) מצב היום

קשיח token או (service role למשל) חלק מהסקריפטים משתמשים במפתחות חזקים.

מינימלי (בלי שינוי קוד) V2 יעד

- איפה נשמרים מפתחות ולמי יש גישה "Credentials" צריך להבהיר במסך Installer ה-
- config/credentials על קבצי (chmod 600) לשמור הרשאות קבצים מחמירות.

מומלץ (שינוי ארכיטקטורה) V3 יעד

- מהמחשב של הלקוח service role להוציא לחלוטין.
- מוגבל שמונפק מהשרת device token-לעבור ל

(Acceptance Tests) בדיקות בסוף התקנה

- `systemctl is-active cloudflared`
- `curl -I http://localhost:8080/` (file listing או בדיקת)
- לכל מצלמה:
 - `stream.m3u8` קיים
 - מתעדכן `age < 60s` m3u8-ה
- בדיקת גישה חיצונית:
 - `https://<subdomain>.clearpoint.co.il/.../stream.m3u8`
- לוגים:
 - `journalctl -u <service> -n 50` או לוג פיזי ב `/opt/clearpoint/logs`

(Prerequisites) מה צריך להכין לפני שמתחילים לממש

- GUI (Electron/Tauri/GTK) החלטה על טכנולוגיית
- Packaging: החלטה על
 - `self-extracting installer` או `.deb`
- `config.yml` ל-`schema` הגדרה סופית של

מוצע Roadmap

(בלבד, ללא שינוי קוד Installer) V2

- Wizard GUI
- יצירת קונפיג
- העתקת קבצים קיימים למיקום סטנדרטי
- `services/cron` יצירת

- בדיקות סיום

V3 (שדרוג מוצר)

- אחד במקום 4 רכיבים Edge agent
- systemd timers במקום cron
- מוגבל security token
- DNS אוטומטי (Cloudflare API)