

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

מטרת המסמך

עבור (Mini PC) עבור תהליכי התקינה מודרני, פשוט, ו"מקצועי" למחשב הלקוח (Design Spec) מסמך זה מגידר תכונת Clearpoint.

- **מטרה עסקית:** לאפשר התקינה בשטח ע"י טכני/מתקין חיצוני בצורה מהירה וברורה.
- במקומם רצף פקודות ידני (שירותים + Wizard) "מטרה טכנית": להפוך את התקינה ל"מושך".
- **הגבלה:** מסמך זה **לא משנה כלום בקוד הקיט'**. הוא מפרט את היעד ואת התכונון לארסה הבאה.

Scope

In-scope (V2 בגרסת)

- Installer (Wizard) גרפי Ubuntu 22.04.
- ביצורה אוטומטית (RAM / שירותים / תיקיות / FFmpeg / Cloudflared disk) התקינה של תלותים נדרשים.
- ביצורה מודרכת (כולל בדיקות) Cloudflared tunnel + DNS יצירת/ניהול reboot.
- "יצירת קונפיג מקומי מסודר במקום" קבצים מפוזרים.
- מסך סיכום + (health checks) בדיקות בסיום התקינה.

Out-of-scope (בשלב זה)

- - שינוי/כתביה מחדש של הלוגיקה הקיים בת scripts/utils/* .
- V3. זה יכול להיות אחד "Agent" ריפקטורי מלא ל-
- DB שינוי סכימת.

הגדרת UX Installer GUI

שפה

עם אפשרות לאנגלית בעתייד (RTL) בירית מחדל: **עברית**.

Flow-Wizard של ה

1. Welcome

- הסביר קצר מה יותקן.
- sudo דרישת הרשות.

2. פרטי לקוח

- שם לקוח / מזאה 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.
- ונצרים מראש מתוך מערכת ניהול camera-*.*.service ו/או camera-.sh USB-ומועברים ל(/admin/cameras).
- לפני המשך USB-מזאה/מאמת שנמצאו קבצי מצלמות בInstaller-ה.

5. Cloudflare Tunnel חיבור ל

- Login (cloudflared tunnel login)
- Create tunnel
- config ייצור
- cloudflared c-service התקנת

6. Install / Apply

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

7. Verification

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

- חיצוני URL בדיקת
- uploader “dry run” או TEST_MODE)

8. Finish

- סטטים, איפה הלוגים URL: סיכום
- ”כפתרו: “העתק כתובות”

(Components) רכיבים

1) Runtime (בשתח)

: ניתן להישאר עם המנגנון הנוכחי, אבל לארוז אותו בצורה מסודרת 2-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/cron/התקינה תשכפל/תמוך אותם באותו מקום אחד ומוגדר, וтвор שירותים

2) Installer GUI

של ההתקינה, ומדוחת state מנהלת sudo אפליקציה שמריצה פעולות מערכת עם Progress.

אפשרויות טכנולוגיות (לבחירה בהמשך):

- **Electron** קל ומהיר לפיתוח; דרוש bundling (GUI)
- **Tauri** קל יותר, אבל מרכיב יותר סביב (Rust)
- **Python + GTK** (גופץ בלינוקס; פחות וובי)

והתנהגות UX המسمך לא מקבע טכנולוגיה – רק

מבנה קבצים מומלץ על המחשב

להחליפ “פיזור” בסטנדרט:

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

קונפיג (Config) – config.yml

דוגמה (תכניו):

- 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 min-Admin-קבצי המצלמות מסופקים מראש מה Installer-camera-*.*.sh לא מייצר config. מהתוך המאץ מה config.

systemd services (תכנו)

מטרה: לא להיות תלויים בהרצה ידנית.

שירותים מוצעים:

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

טובי logging/logging בהמשך כדי לקבל ניטור systemd timers אבל מומלץ לעבור לcron אשר להשאר V2-V3.

Cloudflared + DNS

עשה installer מה

- cloudfared tunnel login
- בשם הלוקה tunnel יוצרת
- כתיבת config.yml/etc/cloudfared/config.yml
- cloudfared service install

DNS

או פעולה ידנית API דרש הרשות DNS-record ב-Cloudflare.

אפשרויות:

- DNS-המסר מציג למשתמש מה להוסיף ב-(**V2**) פשוט (Name/Target), copy buttons.
- אוטומטי (**V3**) שימוש ב-Cloudflare API token כדי ליצור CNAME.

השלכות מוצר – אבטחה

(Known) מצב היום

קשה token או service role (למשל) חלק מהסקרים משתמשים במפתחות צזקים.

מינימלי (בלי שינוי קוד) **V2** יעד

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

V3 יעד מומלץ (שינוי ארכיטקטורה)

- מהמחשב של הלוקה role service להוציא לchlottiin.
- מוגבל שמוונפק מהשרת device token לעבור ל-

בדיקות בסוף התקנה (Acceptance Tests)

- systemctl is-active cloudfared
- curl http://localhost:8080/ או בדיקת file listing
- לכל מצלמה:
 - stream.m3u8 קיימן
 - מתעדכן 8-מ3ענ-ה (age < 60s)
- בדיקת גישה חיצונית:
 - https://<subdomain>.clearpoint.co.il/.../stream.m3u8
- לוגים:
 - journalctl -u <service> -n 50 או לוג פיזי ב /opt/clearpoint/logs

מה צריך להכין לפני שמתחילים למש (Prerequisites)

- החלה על טכנולוגית GUI (Electron/Tauri/GTK).
- Packaging: החלה על
 - self-extracting installer.
 - deb או .deb
- הגדרה סופית של config.yaml ל- schema .

Roadmap

(בלבד, ללא שינוי קוד Installer)

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

- בדיקות סיום

(שדרוג מוצר) V3

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