

ניתוח טכני והערכה פיננסית: ספריית ComposeMediaPlayer

תקציר מנהלים

דוח זה מציג ניתוח מקיף והערכת שווי פיננסית עבור ספריית הקוד הפתוח ComposeMediaPlayer, הזמינה במאגר ה-GitHub תחת המזהה `kdroidFilter/ComposeMediaPlayer`. הניתוח מתמקד בשני יעדים מרכזיים: הערכת העלוות הנדרשת לפיתוח הספרייה מאפס, וניתוח מעמיק של רמת המיומנות, איכות העבודה והמומחיות הכוללת של המפתח, המכונה "kdroidFilter".

ממצאי הדוח מצביעים על כך ש-ComposeMediaPlayer אינה ספריית עזר פשוטה, אלא פתרון הנדסי מתוחכם לבעיה מורכבת ומוכרת היטב בתחום פיתוח המובייל והפלטפורמות המרובות (Multiplatform). הספרייה מספקת שכבת הפשטה (abstraction) חיונית מעל המורכבות הטבועה בנגני מדיה נייטיב בפלטפורמות שונות, ובכך חוסכת למפתחים זמן פיתוח יקר ומפחיתה באופן משמעותי את רף הכניסה הטכני.

ההערכה הפיננסית מגיעה למסקנה כי עלות הפיתוח המשוערת לשכפול נכס תוכנה זה, בהתבסס על שעות עבודה של מומחה ברמה הנדרשת ובשוק הישראלי, נעה בטווח של **\$85,000 עד \$100,000 דולר אמריקאי**. הערכה זו מבוססת על פירוק מפורט של שלבי הפיתוח, החל ממחקר וארכיטקטורה ועד לבדיקות, תיעוד ופרסום.

ניתוח פרופיל המפתח, אלי גמבש ("kdroidFilter"), חושף מומחה ברמה הגבוהה ביותר, שכישוריו חורגים מהגדרת "מפתח בכיר" טיפוסי. תיק העבודות שלו, המורכב כמעט אך ורק מכלי תשתית וספריות עבור סביבת Kotlin Multiplatform (KMP), ממצב אותו כאדריכל תוכנה ומומחה נישה בעל ערך שוק גבוה. עבודתו מדגימה באופן עקבי יכולת לזהות בעיות יסוד באקוסיסטם, לתכנן פתרונות אלגנטיים ולבצע אותם ברמה מקצועית גבוהה, כולל תיעוד מקיף ותחזוקה שוטפת.

לסיכום, הפרויקט מהווה הישג הנדסי משמעותי ונכס קניין רוחני בעל ערך. הן הספרייה עצמה והן יוצרה מייצגים נכסים אסטרטגיים עבור כל ארגון טכנולוגי השואף להאיץ את מאמצי הפיתוח שלו בסביבת Kotlin Multiplatform.

חלק 1: ניתוח טכני וארכיטקטוני של

ComposeMediaPlayer

חלק זה מבסס את הנחת היסוד של הדוח: ComposeMediaPlayer אינו כלי עזר פשוט, אלא פתרון מתוחכם לבעיה הנדסית קשה 臭명. באמצעות פירוק הארכיטקטורה והתכונות שלו, אנו מכמתים את המורכבות הפנימית שלו, המהווה קלט קריטי להערכה הפיננסית הבאה.

א. מרחב הבעיה: המורכבות הטבועה בניגון מדיה באנדרואיד מודרני וב-KMP

כדי להעריך נכונה את ערכה של ComposeMediaPlayer, חיוני להבין תחילה את האתגרים שהיא פותרת. ניגון וידאו ואודיו ביישומי מובייל מודרניים, ובמיוחד בסביבת Jetpack Compose, הוא משימה מורכבת באופן מפתיע. ספריית המדיה הרשמית של גוגל, Jetpack Media3, היא אמנם חזקה, אך היא מציגה מורכבות משמעותית, בפרט עבור מפתחים חדשים למערכות מבוססות View או אלה העובדים בפרדיגמה הדקלרטיבית של Compose.¹ הצורך לגשר בין רכיבי אנדרואיד מסורתיים (כמו

PlayerView) לתוך עולם ה-Compose באמצעות AndroidView מהווה מקור נפוץ לבאגים, בעיות ביצועים ואתגרים ארכיטקטוניים.³

הקושי הזה אינו תיאורטי; הוא מתועד היטב בקהילת המפתחים. דיונים מקצועיים ומערכות מעקב אחר תקלות רשמיות מדגישים את המאבקים של הקהילה. לדוגמה, במערכת מעקב התקלות של גוגל ניתן למצוא בקשות לרכיב מדיה פשוט יותר, מבוסס Compose באופן מלא, מה שמאמת את הצורך בשוק בספרייה כמו ComposeMediaPlayer.² מדריכים ומאמרים טכניים רבים מתמקדים בכמות הקוד התבניתי (boilerplate) ובניהול מחזור החיים (lifecycle) הנדרשים רק כדי להפעיל נגן ExoPlayer בסיסי ב-Compose, מה שמדגיש את מחסום הכניסה הגבוה.⁴

המורכבות גדלה באופן אקספוננציאלי בהקשר של Kotlin Multiplatform (KMP). בסביבה זו, מפתח חייב לשלוט לא רק ב-Media3/ExoPlayer של אנדרואיד, אלא גם ב-AVPlayer של אפל (עבור iOS ו-macOS), ב-Media Foundation של מיקרוסופט (עבור Windows), ב-GStreamer (עבור לינוקס), וב-API של HTML5 (עבור Web/Wasm).¹² האתגר אינו טמון רק במימוש כל פלטפורמה בנפרד, אלא ביצירת API מאוחד ומופשט שמרגיש פשוט ועקבי למפתח המשתמש בספרייה. הערך של

ComposeMediaPlayer אינו טמון רק בפונקציונליות שלה, אלא בהפשטת המורכבות. היא פועלת כ"סופג מורכבות", סופחת אליה את הנטל העצום של ניגון מדיה ספציפי לכל פלטפורמה, וחושפת API דקלרטיבי ופשוט. זוהי תכונה מובהקת של תוכנה תשתית בעלת ערך גבוה. תהליך המחשבה המוביל למסקנה זו מתחיל בהבנת הקושי הבסיסי בניגון מדיה ב-Compose, כפי שעולה מהמקורות.² לאחר מכן, זיהוי התמיכה בשש פלטפורמות שונות, כל אחת עם מסגרת מדיה נייטיב משלה,¹² מכפיל את הבעיה פי שש. ההישג הארכיטקטוני המרכזי, אם כן, אינו רק "לגרום לווידאו להתנגן", אלא ליצור API יחיד ואלגנטי שמסתיר שש מערכות מורכבות, שונות ולעיתים קרובות לא תואמות. כתוצאה מכך,

הצעת הערך העיקרית של הספרייה היא צמצום זמן הפיתוח והמומחיות הנדרשת ממשתמשיה. צוות המשתמש בספרייה זו יכול לעקוף שבועות או חודשים של מחקר ופיתוח מיוחדים, מה שמתורגם ישירות לחיסכון בעלויות וזמן יציאה מהיר יותר לשוק. זה הופך את הספרייה למאפשר אסטרטגי, ולא רק לרכיב.

ב. צלילה עמוקה לארכיטקטורה: כיתת אמן בהפשטה

הארכיטקטורה של הספרייה היא המאפיין המרשים ביותר שלה. היא משתמשת בתבנית עיצוב קלאסית מסוג "Facade", המספקת ממשק יחיד ומאוחד מעל תת-מערכות שונות ומורכבות.¹² ניתוח הפרויקט חושף החלטה ארכיטקטונית פרגמטית ומקצועית: במקום לבנות מנוע מדיה מאפס, המפתח בחר למנף את הנגנים הנייטיב של כל פלטפורמה. החלטה זו מתעדפת ביצועים ויציבות על ידי שימוש ברכיבים שנבדקו בשטח, תוך מיקוד הערך הייחודי של הספרייה בשכבת ההפשטה של KMP.

הספרייה מממשת תמיכה ייעודית לכל פלטפורמה, כפי שמפורט בקובץ ה-README של הפרויקט¹²:

- **אנדרואיד:** Media3
- **iOS:** AVPlayer-I
- **macOS:** AVPlayer-I
- **Windows:** Media Foundation
- **לינוקס:** GStreamer
- **(WebAssembly (Wasm):** נגן HTML5

הדבר מדגים רוחב יריעה נדיר ובעל ערך של ידע ספציפי לפלטפורמות שונות. עם זאת, גולת הכותרת היא ה-API המאוחד. פונקציות כמו `play()`, `pause()`, `rememberVideoPlayerState()`, `seekTo()` מספקות חווית מפתח עקבית בכל הפלטפורמות, ומסתירות את הפרטים המורכבים של הקריאות הנייטיב שמתחת לפני השטח. "קסם" זה מייצג חלק ניכר ממאמץ הפיתוח והוא ליבת הערך של הספרייה.

ג. מערך התכונות ורמת המימוש

הספרייה מציעה סט תכונות עשיר שכל אחת מהן מוסיפה רמת מורכבות משמעותית לפיתוח. ניתוח שיטתי של התכונות המפורסמות¹² חושף את עומק העבודה ההנדסית:

- **שכבות UI מותאמות אישית (Overlays):** תכונה קריטית ליישומי מדיה מודרניים. היכולת לממש זאת באמצעות Compose טהור, גם במצב מסך מלא, היא הישג טכני משמעותי העוקף את המגבלות של ממשקי המשתמש הנייטיב של הנגנים.
- **תמיכה בכתוביות (SRT/VTT):** מימוש תכונה זו דורש לא רק ניתוח (parsing) של פורמטי קבצים, אלא גם סנכרון מדויק של הצגת הטקסט עם תזמון ניגון הווידאו – משימה לא טריוויאלית כלל. ההחלטה לרנדר את הכתוביות באמצעות Compose כדי להבטיח מראה אחיד בכל

- הפלטפורמות היא בחירה חכמה וממוקדת איכות.
- **חילוץ מטא-דאטה (Metadata):** דורש התממשקות עם ממשקי API נמוכי-רמה של מדיה בכל פלטפורמה כדי לשלוף נתונים כמו רזולוציה, קצב סיביות (bitrate) ומשך זמן. הציון כי תכונה זו היא "ניסיונית" ומוגבלת ב-WASM הוא סימן לתיעוד כן ושקוף.
- **הצגת עוצמת שמע (Audio Levels):** ויזואליזציה בזמן אמת של ערוצי שמע דורשת גישה לצינורות עיבוד אודיו נמוכי-רמה.
- **תכונות ניסיוניות (מסך מלא, שינוי קנה מידה של תוכן):** המפתח מתייג נכונה תכונות אלה כניסיוניות, מתוך הכרה בקושי העצום להשיג התנהגות עקבית בין מערכות הפעלה ומנהלי חלונות שונים. הדבר מדגים בגרות וגישה אחראית ליציבות ה-API.

ד. בגרות הפרויקט, תיעוד וחזון

הפרויקט אינו יצירה חד-פעמית, אלא מוצר תוכנה בוגר ומתוחזק. היסטוריית הגרסאות והפרסומים במאגר המרכזי של ¹³ Maven מעידה על מאמץ תחזוקה פעיל ומתמשך. סטטיסטיקת מערכת מעקב התקלות, על אף מספר קטן של בעיות, מצביעה על תגובתיות מצד הבעלים.¹⁶

איכות התיעוד יוצאת דופן עבור פרויקט קוד פתוח. קובץ ה-¹² README.md מקיף, ברור ומובנה היטב. הוא מספק הוראות התקנה, טבלת תאימות, דוגמאות קוד ברורות להתחלת עבודה, ואזהרות לגבי תכונות ניסיוניות. רמת תיעוד זו אופיינית לספרייה מקצועית ברמה הגבוהה ביותר.

לבסוף, מפת הדרכים (Roadmap) הציבורית של הפרויקט, הכוללת תכונות כמו נגן אודיו עצמאי ותמיכה בזרמי אודיו ווידאו נפרדים¹², מדגימה חזון ברור ושאפתני לעתיד הספרייה. הדבר מצביע על כך שהמפתח חושב באופן ארכיטקטוני ומתכנן צמיחה ותועלת לטווח ארוך.

חלק ו: פרופיל יכולות של המפתח ("kdroidFilter" / אלי גמבש)

חלק זה עובר מניתוח ה"מה" לניתוח ה"מי". על ידי בחינת כלל תיקי העבודות הציבורי של המפתח, ניתן לבנות פרופיל החורג הרבה מעבר להיקף של פרויקט בודד. מבט הוליסטי זה חיוני לסיווג מדויק של רמת המיומנות שלו, אשר בתורו מצדיק שימוש במודל תגמול של מומחה בכיר בהערכה הפיננסית.

א. פורטפוליו של כלי תשתית, לא רק יישומים

זהות המפתח היא אלי גמבש ("kdroidFilter") מירושלים, ישראל.¹⁷ נתון גיאוגרפי זה הוא קריטי, מכיוון שהוא מאפשר לנו לעגן את הניתוח הפיננסי בשוק ההייטק הישראלי הספציפי והיקר.

סקירה של מאגרי הקוד הציבוריים האחרים של המפתח חושפת דפוס עקבי של יצירת כלי תשתית בעלי ערך גבוה עבור KMP.¹⁷ דוגמאות בולטות כוללות:

- StoreKit²⁰: ספריית KMP מקיפה לחילוץ נתונים (scraping) מחנויות אפליקציות אנדרואיד שונות (Google Play, F-Droid, Aptoide). זהו כלי מורכב לחילוץ נתונים ואינטראקציה עם API, לא יישום UI פשוט.
 - Platform-Tools¹⁷: ספריית KMP להפשטת כלי עזר ספציפיים לפלטפורמה, כמו ניהול ספריות מטמון (cache) ופונקציות ברמת מערכת ההפעלה. זוהי דוגמה נוספת ל"סופג מורכבות" תשתית.
 - DesktopNotifyKT¹⁷ ו-ComposeNativeTray: ספריות ליצירת אייקונים נייטיב במגש המערכת (system tray) והתראות מ-Compose, המתמודדות עם הבעיה הקשה של אינטגרציה עם שולחן העבודה.
 - כלי עזר נוספים¹⁹: אוסף של ספריות קטנות יותר אך עדיין משמעותיות לרישום לוגים, אספקת קונטקסט ועוד.
- המיקוד העקבי של המפתח ביצירת ספריות KMP המפשטות הבדלים בין פלטפורמות, במקום לבנות יישומי KMP, הוא המדד החשוב ביותר לרמת המיומנות הגבוהה שלו ולערכו בשוק. בניית יישום ב-KMP היא מאתגרת ודורשת טיפול בלוגיקה משותפת וב-UI ספציפי לפלטפורמה. עם זאת, בניית ספרייה ב-KMP קשה בסדר גודל. היא דורשת חשיבה על עיצוב API, תאימות בינארית, ניהול תלויות על פני יעדים מרחבים, ויצירת הפשטות שיהיו שימושיות למגוון רחב של צרכנים לא ידועים. תיק העבודות של המפתח¹⁷ מורכב כמעט כולו מספריות כאלה. הדבר מצביע על שינוי תפיסתי מ"בונה מוצרים" ל"בונה כלים" או "אדריכל פלטפורמה" – עבודה המאפשרת למערכות אקולוגיות שלמות להתקיים. לכן, בעת הערכת זמנו, לא ניתן להשתמש במשכורת של מפתח יישומים, אפילו בכיר. יש להשתמש בתעריף של יועץ מומחה או אדריכל הנשכר לפתור את הבעיות התשתיות הקשות ביותר, מה שמצדיק שימוש בתעריפי פרילנס מהשורה הראשונה.²⁴

ב. שליטה מוכחת בנישה מבוקשת: Kotlin Multiplatform

תיק העבודות של המפתח ממוקד כמעט אך ורק בקוטלין, ובאופן ספציפי בתחום המאתגר של Kotlin Multiplatform. זהו אינו מפתח אנדרואיד שמתנסה ב-KMP; זהו מומחה KMP שבונה כלים עבור מפתחים אחרים. התמחות זו זוכה לפרמיה בשוק. אופי הפרויקטים שלו – ספריות תשתית ולא יישומים למשתמשי קצה – ממצב אותו כבונה אקוסיסטם. הוא יוצר את הכלים המאפשרים למפתחים אחרים להיות פרודוקטיביים יותר. הדבר מדגים הבנה ארכיטקטונית עמוקה של מסגרת KMP ושל נקודות הכאב שלה.

ג. סיווג מיומנות טכנית: מעבר ל"בכיר"

ניתוח עבודתו של המפתח חושף מאפיינים ברורים של מהנדס ברמת Principal או Staff, רמות הנמצאות מעל "מהנדס בכיר" טיפוסי:

- **אוטונומיה וחזון:** המפתח מזהה בעיות מורכבות ובלתי פתורות באקוסיסטם של KMP, ומתכנן, בונה, מתעד ומתחזק פתרונות באופן עצמאי.
- **חוש ארכיטקטוני:** השימוש העקבי בהפשטה, תבניות Facade ומימושים ספציפיים לפלטפורמה על פני פרויקטים מרובים מדגים רמה גבוהה של בגרות ארכיטקטונית.
- **מכפיל השפעה (Impact Multiplier):** הספריות שהוא יוצר פועלות כמכפיל כוח, ומאפשרות לאינספור מפתחים אחרים לעבוד ביעילות. זוהי תכונה מרכזית של מהנדסים הפועלים ברמות Staff/Principal.

לפיכך, בהתבסס על הראיות, "kdroidFilter" פועל ברמה הגבוהה משמעותית מזו של מפתח אנדרואיד בכיר טיפוסי. כישוריו תואמים יותר ל**מהנדס Principal או לאדריכל טכני מומחה**. סיווג זה הוא קריטי, מכיוון שהוא מכתیب את אמת המידה המתאימה לתגמול שיש להשתמש בה בהערכת השווי שלנו. שימוש במשכורת של "מפתח בכיר" סטנדרטי ימעיט באופן גס בערך המומחיות המופגנת.

חלק וו: הערכה פיננסית של מאמץ הפיתוח

חלק זה מסנתז את המורכבות הטכנית (חלק ו) ואת מומחיות המפתח (חלק וו) לכדי הערכה פיננסית כמותית. נעשה שימוש במתודולוגיית Bottom-Up, על ידי פירוק הפרויקט לשלבים והחלת תעריף שעותי שחושב בקפידה, ולאחר מכן אימות התוצאה מול נתוני שוק כלליים (Top-Down).

א. מתודולוגיית הערכה: ניתוח מבוסס שלבים (Bottom-Up)

הגישה שנבחרה היא להעריך את סך שעות האדם הנדרשות לשכפול הפרויקט על ידי מהנדס יחיד ברמה שזוהתה בחלק וו. הפרויקט יפורק לשלבים דיסקרטיים והגיוניים, החל ממחקר ראשוני ועד לתיעוד סופי. גישה מפורטת זו מספקת שקיפות ומאפשרת הערכה סופית מדויקת וניתנת להגנה יותר מאשר הערכה המבוססת על "תחושת בטן".

ב. גזירת תעריף שעתית משוקלל לאדריכל KMP מומחה בישראל

כדי לבסס את התעריף השעתי, נותחו מספר מקורות נתונים על תגמול בישראל:

- **עובדים שכירים:** משכורת ממוצעת של מפתח אנדרואיד בכיר עומדת על כ-308,000 ₪ לשנה.²⁷ בהנחה של כ-2,000 שעות עבודה בשנה, הדבר מתורגם לכ-154 ₪ לשעה. נתון זה מייצג את העלות למעסיק ואינו כולל תקורה של פריילנסר.
- **פלטפורמות פריילנס:** ב-Upwork ניתן למצוא מפתחי Full-Stack ישראלים מובילים הגובים תעריפים כמו \$40 לשעה²⁴, אם כי נתון זה יכול להיות מוטה כלפי מטה עקב תחרות גלובלית.
- **תעריפי סוכנויות ויועצים:** נתונים גלובליים מצביעים על כך שמפתחים פריילנסרים מומחים יכולים לגבות תעריפים של \$100-\$150 לשעה ויותר.²⁶ מקור ספציפי לישראל מציין טווח של \$31-\$75 לשעה, עם ממוצע של \$45 לשעה.²⁵ סוכנויות פיתוח בישראל מפרסמות תעריפים שעתיים בטווחי \$25-\$49 ו-\$50-\$99.³⁰

ישנן מספר סיבות להצדקת תעריף פרימיום במקרה זה:

1. כפי שנקבע בחלק ו, המפתח הוא אדריכל מומחה, לא מפתח יישומים סטנדרטי. כישוריו הם נישתיים ומבוקשים מאוד.
2. תעריף פריילנס חייב לכסות תקורות שאינן כלולות במשכורת: מיסים, ביטוח, ציוד, שעות לא מחויבות (מכירות, אדמיניסטרציה) ומרווח רווח. כלל אצבע נפוץ הוא להכפיל את התעריף השעתי של שכיר כדי לקבל תעריף פריילנס בסיסי.
3. שוק ההייטק הישראלי תחרותי ויקר. התעריפים הנמוכים יותר בשוק הפריילנס מייצגים ככל הנראה כישרונות צעירים יותר או כלליים יותר.

בהתבסס על שילוב המקורות, נבחר תעריף משוקלל. התחלה מהשכר הבכיר (154 ₪ לשעה $\approx$ \$42 לשעה), והכפלתו עבור תקורת פריילנס, מביאה אותנו לסביבות \$84 לשעה. תעריפי הסוכנויות מגיעים עד \$99 לשעה³⁰, והתעריפים למומחים גלובליים מגיעים ל-\$150 לשעה.²⁶ בהתחשב בכישורי הארכיטקטורה וההתמחות ב-KMP, תעריף בקצה העליון של ספקטרום היועצים/סוכנויות בישראל הוא המתאים ביותר. לכן, לצורך החישוב, יעשה שימוש בתעריף שעתית משוקלל, שמרני אך מומחה, של

\$90 דולר אמריקאי.

ג. כימות מאמץ ופירוט עלויות

חלק זה מציג את הטבלה המרכזית של הדוח, המפרטת את מאמץ הפיתוח. הצגת הפירוט בטבלה מספקת שקיפות מרבית ומאפשרת לבחון כיצד חושב המספר הסופי. היא הופכת הערכה סובייקטיבית למודל פיננסי מובנה וניתן לניתוח, ובכך עונה על הדרישה המרכזית לדוח ברמת מומחה.

טבלה 1: הערכת מאמץ פיתוח ופירוט עלויות עבור ComposeMediaPlayer

שלב / משימה	תיאור המורכבות	שעות אדם מוערכות	תעריף שעתי משוקלל (USD)	עלות מוערכת (USD)
שלב 1: מחקר וארכיטקטורה	מחקר ופיתוח ראשוני של שש מסגרות המדיה, תכנון ה-API המאוחד של KMP, תכנון מבנה המודולים.	120	\$90	\$10,800
שלב 2: מימוש מודול הליבה	בניית מודול ה-KMP המשותף, הגדרת ממשקים, לוגיקת ניהול מצב (state management).	80	\$90	\$7,200
שלב 3: מימוש צד אנדרואיד (Media3)	מימוש לוגיקת הנגן באמצעות Media3, טיפול במחזור חיים, גישור ל-Compose.	100	\$90	\$9,000
שלב 4: מימוש צד iOS/macOS (AVPlayer)	מימוש לוגיקת הנגן באמצעות AVPlayer, טיפול בתאימות בין-פלטפורמית (interop), התמודדות עם אילוצי אפל.	140	\$90	\$12,600
שלב 5: מימוש צד דסקטופ (Windows, Linux)	מימוש עבור Media Foundation, GStreamer, משימה מורכבת	160	\$90	\$14,400

			הדורשת ידע עמוק ברמת מערכת ההפעלה.	
\$8,100	\$90	90	מימוש נגן HTML5 וטיפול בתאימות דפדפנים.	שלב 6: מימוש צד ווב (Wasm/JS)
\$5,400	\$90	60	בניית מנתח (parser) ל-SRT/VTT ולוגיקת הרינדור והסנכרון מבוססת Compose.	שלב 7: תכונה - מנוע כתוביות
\$4,500	\$90	50	פיתוח ה-API עבור פקדי UI מותאמים אישית ב-Compose.	שלב 8: תכונה UI ושכבות מותאמות
\$9,000	\$90	100	כתיבת בדיקות יחידה/אינטגרציה (בהנחה), הגדרת GitHub Actions לבנייה ופרסום.	שלב 9: בדיקות CI/CD
\$3,600	\$90	40	כתיבת קובץ ה-README המקיף, פרסום החבילות ל-Maven Central.	שלב 10: תיעוד ופרסום
\$93,600		1,040		סה"כ

ד. חישוב עלות סופי ואימות שוק

החישוב המפורט בטבלה מגיע לסך של 1,040 שעות עבודה, שבהכפלתן בתעריף השעתי של \$90 מביאות לעלות פיתוח מוערכת של \$93,600. כדי לשקף את אי-הוודאות הטבועה בהערכות פרויקטים מורכבים, מוצג טווח סביר של +/- 15%, מה שמוביל להערכה סופית בטווח של \$79,560 עד \$107,640.

כדי לאמת את סבירותה של הערכה זו, ניתן להשוות אותה לעלויות פרויקטים כלליות בשוק הישראלי. עלות פיתוח אפליקציית מובייל מלאה בישראל יכולה לנוע בין \$50,000 ל-\$250,000 ויותר, תלוי במורכבות.³¹ בעוד ש-

ComposeMediaPlayer היא ספרייה ולא אפליקציה מלאה, המורכבות שלה דומה לפיתוח תכונת ליבה ביישום מדיה מתוחכם. העובדה שההערכה שלנו נופלת באופן הגיוני בתוך הקשר שוק רחב זה, מאששת את סבירותה.

מסקנות והמלצות אסטרטגיות

א. סינתזת ממצאים

הניתוח המעמיק מוביל למספר מסקנות מרכזיות:

1. **נכס קניין רוחני בעל ערך:** ComposeMediaPlayer היא נכס אינטלקטואלי בעל ערך גבוה, הפותר בעיה מורכבת ונפוצה באקוסיסטם של Kotlin Multiplatform. היא מהווה פתרון "מדף" איכותי המאיץ פיתוח ומפחית סיכונים טכניים.
2. **כישרון ברמה עולמית:** המפתח, "kdroidFilter" (אלי גמבש), הוא אדריכל KMP ברמה הגבוהה ביותר, ששישוריו זוכים לפרמיית שוק משמעותית. הוא מייצג פרופיל של מהנדס בעל השפעה גבוהה, המסוגל להוביל פרויקטים תשתיתיים מורכבים באופן עצמאי.
3. **הערכת שווי מבוססת:** הערכת העלות הסופית, הנעה סביב \$94,000, משקפת נאמנה הן את המורכבות הטכנית של העבודה והן את רמת הכישרון הגבוהה הנדרשת לביצועה.

ב. תובנות ניתנות לפעולה

הניתוח מספק תובנות אסטרטגיות עבור קהלים שונים:

- **עבור מנהל טכנולוגיות ראשי (CTO) או מייסד טכני:** ניתוח זה יכול לשמש כאמת מידה לתקצוב פרויקטים פנימיים בעלי מורכבות דומה. הוא מדגיש את ההשקעה המשמעותית הנדרשת לבניית כלי תשתית בתוך הארגון ויכול להנחות החלטות של "לבנות או לקנות" (buy vs. build).
- **עבור משקיע הון סיכון (VC) או אנליסט:** דוח זה מספק מסגרת להערכת נכסים טכניים וכישרונות במהלך בדיקת נאותות. המפתח "kdroidFilter" מייצג פרופיל של מהנדס בעל השפעה גבוהה שיכול להוות גיוס אסטרטגי יקר ערך או אבן פינה במיזם טכני חדש. הספרייה עצמה יכולה להיות נכס בעל ערך עבור חברה המעוניינת להאיץ את כניסתה לתחום ה-KMP.

Sources des citations

1. Introduction to Jetpack Media3 | Android media, consulté le septembre 18, 2025, <https://developer.android.com/media/media3>
2. Official Media compose component or Tutorial to implement Jetpack media 3 in Jetpack compose. [287771883] - Issue Tracker, consulté le septembre 18, 2025, <https://issuetracker.google.com/issues/287771883>
3. A few video in a list overlap each other · Issue #1107 · androidx/media - GitHub, consulté le septembre 18, 2025, <https://github.com/androidx/media/issues/1107>
4. Create ExoPlayer VideoView in Android Jetpack Compose - GeeksforGeeks, consulté le septembre 18, 2025, <https://www.geeksforgeeks.org/kotlin/create-exoplayer-videoview-in-android-jetpack-compose/>
5. Mastering ExoPlayer in Jetpack Compose: Full Guide for Building a Video Player in Android | by Yodgorbek Komilov | Medium, consulté le septembre 18, 2025, <https://medium.com/@YodgorbekKomilo/mastering-exoplayer-in-jetpack-compose-full-guide-for-building-a-video-player-in-android-a1dc08d68cad>
6. Custom ExoPlayer Controls in Jetpack Compose | by Ruben Quadros - Better Programming, consulté le septembre 18, 2025, <https://betterprogramming.pub/custom-exoplayer-controls-in-jetpack-compose-c4089def0106>
7. How To Build a Music Player with Jetpack Compose & ExoPlayer | by MathRoda - Medium, consulté le septembre 18, 2025, <https://medium.com/@mathroda/how-to-build-a-music-player-with-jetpack-compose-exoplayer-916b7021ae50>
8. Getting Started With EXO Player in Jetpack Compose | by Siva Ganesh Kantamani | ProAndroidDev, consulté le septembre 18, 2025, <https://proandroiddev.com/getting-started-with-exo-player-in-jetpack-compose-72173ed1a86b>
9. Create a basic media player app using Media3 ExoPlayer - Android Developers, consulté le septembre 18, 2025,

- <https://developer.android.com/media/implement/playback-app>
10. Android Compose Videos with ExoPlayer | ProAndroidDev, consulté le septembre 18, 2025,
<https://proandroiddev.com/android-compose-videos-with-exoplayer-8112627ac06c>
 11. Building a Media Player App with Jetpack Media3 in Jetpack compose - Medium, consulté le septembre 18, 2025,
<https://medium.com/@myofficework000/building-a-media-player-app-with-jetpack-media3-in-jetpack-compose-16ae8c35f955>
 12. kdroidFilter/ComposeMediaPlayer: Compose Media Player ... - GitHub, consulté le septembre 18, 2025, <https://github.com/kdroidFilter/Compose-Media-Player>
 13. io.github.kdroidfilter:composemediaplayer - Maven Central - Sonatype, consulté le septembre 18, 2025,
<https://central.sonatype.com/artifact/io.github.kdroidfilter/composemediaplayer>
 14. Compose Media Player - io.github.kdroidfilter - Maven Repository, consulté le septembre 18, 2025,
<https://mvnrepository.com/artifact/io.github.kdroidfilter/composemediaplayer>
 15. io.github.kdroidfilter:composemediaplayer-android - Maven Pa... - Socket.dev, consulté le septembre 18, 2025,
<https://socket.dev/maven/package/io.github.kdroidfilter:composemediaplayer-android/dependencies/0.8.2?classifier=javadoc&ext=jar>
 16. kdroidfilter/compose-media-player issue stats | GitHub - Ecosyste.ms, consulté le septembre 18, 2025,
<https://issues.ecosyste.ms/hosts/GitHub/repositories/kdroidfilter%2Fcompose-media-player>
 17. Elie Gambache @kdroidFilter - GitHub, consulté le septembre 18, 2025,
<https://github.com/kdroidFilter>
 18. io.github.kdroidfilter » platformtools.core-macosarm64 » 0.2.0 - Maven Repository, consulté le septembre 18, 2025,
<https://mvnrepository.com/artifact/io.github.kdroidfilter/platformtools.core-macosarm64/0.2.0>
 19. io.github.kdroidfilter - Maven Repository, consulté le septembre 18, 2025,
<https://mvnrepository.com/artifact/io.github.kdroidfilter>
 20. kdroidFilter/StoreKit: A comprehensive cross-platform Kotlin library for extracting detailed app information from Android stores. - GitHub, consulté le septembre 18, 2025, <https://github.com/kdroidFilter/StoreKit/>
 21. io.github.kdroidfilter:androidcontextprovider - Maven Central - Sonatype, consulté le septembre 18, 2025,
<https://central.sonatype.com/artifact/io.github.kdroidfilter/androidcontextprovider>
 22. io.github.kdroidfilter:storekit-apkcombo-scraper-android 0.8.2 on Maven - Libraries.io, consulté le septembre 18, 2025,
<https://libraries.io/maven/io.github.kdroidfilter:storekit-apkcombo-scraper-android>
 - 23.
 24. Hire the best Android App Developers in Israel - Upwork, consulté le septembre

- 18, 2025, <https://www.upwork.com/hire/android-app-developers/il/>
25. How Much Does It Cost to Hire an App Developer in 2024 - JayDevs, consulté le septembre 18, 2025, <https://jaydevs.com/how-much-does-it-cost-to-hire-app-developer/>
26. What is a good freelance salary ? : r/androiddev - Reddit, consulté le septembre 18, 2025, https://www.reddit.com/r/androiddev/comments/u4bcal/what_is_a_good_freelance_salary/
27. Android Developer Salary in Israel (2025) - SalaryExpert, consulté le septembre 18, 2025, <https://www.salaryexpert.com/salary/job/android-developer/israel>
28. Developer Android Salary in Israel (2025) - SalaryExpert, consulté le septembre 18, 2025, <https://www.salaryexpert.com/salary/job/developer-android/israel>
29. Estabelishing an hourly rate as freelance Android Developer : r/androiddev - Reddit, consulté le septembre 18, 2025, https://www.reddit.com/r/androiddev/comments/6rk182/estabelishing_an_hourly_rate_as_freelance_android/
30. Top App Developers in Israel - Sep 2025 Rankings | Clutch.co, consulté le septembre 18, 2025, <https://clutch.co/il/app-developers>
31. Top Israel Mobile App Development Companies for 2025 - Smartexe, consulté le septembre 18, 2025, <https://smartexe.com/blog/best-mobile-app-development-israel>
32. Mobile App Development Cost in 2025: Full Guide with Cost - APPWRK, consulté le septembre 18, 2025, <https://appwrk.com/insights/mobile-app-development-cost-breakdown/>
33. Mobile App Development Cost in Israel (2025) - Next Olive Technologies, consulté le septembre 18, 2025, <https://nextolive.com/blogs/mobile-app-development-cost-in-israel/>