

ארכיטראז' קלנדיי בשוקי תחזיות - מחקר מעמיך

מבוא: שוקי תחזיות כמו Polymarket מאפשרים למשתמשים למסחר חוזים על אירועים עתידיים, כאשר מחיר החוזה "כן" (YES) או "לא" (NO) משליך את ההסתברות הנחפסת להתרחשות האירוע. במחקרינו עדכניים נחשף כי בשוקים אלו קיימים כשלים¹ תמחור קרוניים המנצלים ע"י סוחרים מתחכמים לארכיטראז' – ככלומר הפקת רווח חסר-סיכון מהפרשי מחירים². אחד מסוגי הארכיטראז' הבולטים הוא **אררכיטראז' קלנדיי (Calendar Arbitrage)** – ארכיטראז' המבוסס על יחסי גומלין בין שוקים שתוחכמים אירועו בזמן. במשמעותו ומעמייק זה נפרט את אסטרטגיית הארכיטראז' הקלנדיי, נוכיח את תקופתה מותאמת בשל תרחישים,ណון בהיבטים מעשיים (عملות, נזילות, Slippage, סיכון ביצוע), ונציג ארכיטקטורת מערכות מלאה ליזהוי והוצאה לפועל אוטומטיים של עסקאות אלו. כמו כן נסקור שירותי NLP ליזהוי שוקים תלויים, מנוע חישוב כדאיות המבוסס על עומק Order Book ולא רק מחירי Mid, מנגנון ניטור בזמן אמת ואסטרטגיות יציאה, ניהול סיכונים, מודולים מרכזיים בארכיטקטורה, טכנולוגיות מומלצות למימוש, ולבסוף – מקרים וניתוחים מדגמים מאירועים בשנים 2024-2025 (בחירה, אירועים גיאופוליטיים, השקות מוצרים) עם תובנות גם מפלטפורמות נוספות מעבר ל-Polymarket.

1. אסטרטגיית Calendar Arbitrage: תנאי הארכיטראז' ויחסיו למחירים

הגדרת האסטרטגייה: ארכיטראז' קלנדיי מתאפשר כאשר קיימים שני שוקים (או יותר) על אותו אירוע, אך בעלי מגבלת זמן שונה. למשל, שוק A שואל "האם אירוע X יקרה עד תאריך T1?" ושוק B שואל "האם אירוע X יקרה עד תאריך T2?" (כאשר T2 מיותר מ-T1). באופןalog, הסתברות שאירוע יתרחש עד תאריך מאוחר (T2) צריכה להיות **גדולה או שווה** להסתברות שיתறש עד תאריך מוקדם יותר (T1), משום שתנאי T2 כולל את T1 ועוד זמן נוספת. במונחי מחירי שוק, אם נסמן ב- P_1 את מחיר החוזה "YES" בשוק עד T1, וב- P_2 את מחיר החוזה "YES" עד T2, התנאי הלויג הוא: $P_1 \leq P_2$. הפרת תנאי זה מייצרת עיון מחרים שניתן לנצל לארכיטראז' חסר-סיכון: אם בפועל $P_2 > P_1$ (כלומר השוק בעל הדדיין הקצר מתמוך הסתברות גבוהה יותר מהשוק הארוך) – ניתן לבצע עסקה משולבת של מכירת חוזי "YES" בשוק הקצר (או קניית "NO" בשוק הקצר) וקניית חוזי "YES" בשוק הארוך, ובכך לנצל רווח ללא תלות בתוצאות העתידיות³. אסטרטגיה זו זהה ב邏輯ה לארכיטראז' קלנדיי באופןיות: מוכרים את החוזה קצר-הטוויה היקר וקונים את החוזה הארוך-הטוויה הזול באותו בסיס, לנעלמת מרוח בטווח⁴.

הגנון לוגי ויחסיו למחירים: התנאי $P_2 \leq P_1$ נובע מכך ש"אם האירוע קרה עד T1, אז ודאי קרה עד T2". כלומר, ככל מי אפשרי מבחינה הגיונית ששוק "כן עד T1" יתמוך ההסתברות גבוהה משוק "כן עד T2" – מצב זה משמעו ההזדמנויות ארכיטראז' ודאיות. באופן דומה, בצד ה"לא": **ההסתברות לא-התרחשות** האירוע עד T1 צריכה להיות גדולה או שווה להסתברות לא-התרחשות עד T2 (שהוא פרק זמן ממושך יותר שבו האירוע עוד עלול ללקות). למעשה, העדר ארכיטראז' מחייב שמחירי חוזי בנאריי יעדמו במגבלה **MONTHONIOT BEZMAN** – מחיר החוזה ככל שפג תוקפו רחוק יותר בזמן אמרור לא לרדת מתחת למחיר החוזה זהה מופר, ניתן לבנות פורטפוליו של פוחזיות בשוקים הקיצרים והארוכים שמבעיטה **תשואה ללא סיכון** (Risk-Free Calendar Spread).

באופן מעשי, אם מזמינים לדוגמה Polymarket שני שוקים על אותו אירוע עם מועד סיום שונים, והחוזה הקצר יקר מהארוך, האסטרטגיה היא: **- למכור חוזה "YES" בשוק הקצר (או לחילופין, לנחות את החוזה "NO" בשוק הקצר – שקול כלכליות לממכר "YES")**, **- ולקנות חוזה "YES" בשוק הארוך.**

צעד זה מנצח את העיונות: אנו מוכרים ביקר את ההסתברות בטוווח הקצר וקונים בוול את אותה הסתברות לטוווח ארוך יותר.icut ננתה תשואה בכל תרחיש אפשרי להתרחשות האירוע.

2. הוכחה מתמטית לתקופות האסטרטגיה בשלושת התרחישים (Payoff) מובטח

כדי להוכיח שהסטרטגיה אכן חסרת-סיכון, ננתן את **שלושת התרחישים ההדדיים** לגבי מועד התרחשויות האירוע: 1. **התרחשות לפני T1:** האירוע קורה מוקדם (לפני או עד T1). 2. **התרחשות בין T1 ל-T2:** האירוע לא קרה עד T1, אך קורה מאוחר יותר לפניו. 3. **אי-התרחשות עד T2:** האירוע איננו מתרחש כלל (לא עד T2 ולא אחריו).

נתיחס לפורטפוליו הארביטראז' שבנו: - **PURCHASE YES:** מכירת חוזה "כן עד T1" במחיר P1 (שער לרכישת "לא עד T1" במחיר (P1-1)). - **PURCHASE LONG YES:** רכישת חוזה "כן עד T2" במחיר P2.

ההשקעה ההתחלתית (עלות הפורטפוליו) היא: $P1 + P2 = \text{Cost}$. ביטוי זה מייצג את הסכום ששילמנו: שילמנו (P1-1) דולר עבור חוזה ה"NO" בשוק הקצר, ו-P2 דולר עבור חוזה ה"YES" בשוק הארוכן. נחשב כעת את התשלומים (Payoff) המתקבל מכל פוזיציה בכל תרחיש, ואת הסכום הכללי:

תרחיש	הוצאת שוק T1	הוצאת שוק T2	סכום התשלומים
			מפורטפוליו הארביטראז'
1. אירוע לפני T1	YES עד 1 = T1 (מתהרהש) מכרנו Yes, שכן מפסידים P1-1 (שילמנו לשקליע הנגדי את ערך החוזה) ומנגד החוזה "NO" שקבענו שווה \$.0\$.	YES עד 1 = T2 (כן) קיינו Yes, מקבלים \$.1\$.	\$1\$ דולר (шиלמנו \$1\$, קיבלנו \$Cost\$ חוזה; ראה ניתוח בהמשך)
2. אירוע בין T1 ל-T2	YES עד 0 = T1 (לא קרה עד T1) > המחיר "NO" עד T1 שבודינו משלם .\$1\$.	YES עד 1 = T2 (כן) קיינה עד (T2) - > המחיר Yes שвидינו משלם \$.1\$.	\$2\$ דולר (קיבלנו \$1\$ משוק T1 ועוד \$.1\$ משוק T2)
3. אירוע לא עד T2	YES עד 0 = T1 .\$1\$ משולם T1	YES עד 0 = T2 > המחיר Yes שלנו שווה \$.0\$.	\$1\$ دولار (קיבלנו \$.0\$ משוק T1, T2 משוק (T2)

ניתוח ההזדמנויות: בפורטפוליו שלנו, בכל תרחיש התשלום המינימלי שמתאפשר הוא \$1\$. בתרחיש 2 אף מתאפשרים \$.2\$ מה הייתה ההשקעה ההתחלתית? $Cost = P1 + P2 - Cost = 1 - (1 - P1) = P2$. אם אכן היה עיון מחייבים כך ש- $P1 > P2$, אז $Cost < 1$. כלומר השענו פחות מ-\$1\$ דולר. לפיכך: - **תרחיש 1: אירוע \$1**: החזר \$1\$ דולר עלות - רווח \$(P1 - 1) > 1\$. כלומר החזר מ+\$1\$ (רווח). - **תרחיש 2: אירוע \$2**: החזר \$2\$ دولار - רווח \$(P2 - 2)\$. - **תרחיש 3: אירוע \$3**: החזר \$1\$ دولار - רווח \$(Cost - 3)\$ (רווח).

בכל מקרה לא ספגו הפסד, והרווח המובטח בכל התרחישים הוא לפחות \$(Cost - 1)\$. שזהו מספר חיובי כאשר $Cost < 1$. **זהו ארביטראז' חסר-סיכון**: לא משנה متى (אם בכלל) יתרחש האירוע X, הפורטפוליו יניב רווח מובטח. במקרה הגורע ביותר נקבל בבדיקה את הערך בחזרה + רווח קטן, ובתרחישים אחרים נקבל אף יותר. מכאן נובע שתנאי $P2 > P1$ הוא תנאי לאפשרות ארביטראז', ובהכרח במצב שוויי משקל נטול ארביטראז' לא יתאפשר $P2 > P1$.

חשוב לציין שבסמך גבולו בו $P1 = Cost = 1$, עלות הפוזיציה $1 - 3$ אז נקבל \$.1\$ בבדיקה - כלומר נקודת איזון (לא הפסד ולא רווח), ובתרחיש 2 נקבל \$.2\$ (רווח \$.1\$). מצב זה עדין אטרקטיבי לסוחר מושם שאין סיכון Break-Even בתרחיש הגורע, ורווח חיובי באחריהם. לכן, בשוקים תחרותיים אפילו שווין ייצור תמיין למסחר עד להתקנסות ל-\$P1\$. הדזקה עם פער מינימלי המשקף אולי עלויות עסקה. בפועל, קיום פער ממשי שבו $P1 < Cost$ גודל ממשמעות מ-2 מ-2 מהו הزادנות רווח אשר סוחרים "חכמים" צפויים להפוץ עליה מיד.

דוגמא מסורתית: נניח שוק A ("כן עד 1 ביולי 2025") נסחר ב-0.60 (כלומר 60% הסתברות) ושוק B ("כן עד 1 ביינואר 2026") נסחר ב-0.50 (50%). לפי הלוגיקה, היה על שוק B להיות לפחות 0.60. הארביטראז' ימכור חוזי "Yes" בשוק A

ב-0.60 (ובכך בעצם קונה "No" ב-0.40) ויקנה "Yes" בשוק B ב-0.50. הערות: $0.50 + \$0.40 = \0.90 לכל יחידה. כעת ניתוח תרחישים: - אם האירוע קרה לפני וולי 2025: הוא קרה גם לפני 2026. מפוזיציותו הסוחר יצטרך לשלם \$1 למני שקנה ממנו את חוזה-h-Yes בשוק A (הפסד \$0.40 כי הוא קיבל \$0.60 ומחייב \$1) אך יקבל \$1 על חוזה-h-Yes שבידייו בשוק B. נטו קיבל \$1, השקייע \$0.90, רווח \$.10. - אם קרה במהלך השנה השנייה של 2025: שוק A פג במפסיד (No) משלם \$1 - הוא הרוויח \$0.60 כי קיבל \$0.60 מכירות בן ומחייב \$0 לזוכה-h-Yes), שוק B זוכה \$.1. סה"כ קיבל \$2 על השקעה \$.90, רווח \$.1. - אם לא קרה עד סוף 2025: שוק A "לא" משלם \$1, שוק B מפסיד \$2 על השקעה \$.90, רווח \$.1. קיבל \$1 על השקעה \$.90, רווח \$.1.

הרוחה בכל מקרה חיובי. אילו $P_2 \leq P_1$ נדרש (נניח 50%-1%), אסטרטגיה כזו הייתה לפחות לא מפסידה (Break-even מינימלי), ولكن בשוק יעל הפרש כזה לא אמור להציג מעמד ללא שינוי מחירים.

3. השפעת عملות, נזילות, Leg Risk-1 Slippage על הרוחות

הנitorה לעיל מניה עולם תאורייתי ללא חיכון – אך במצבות ינסמ גורמים מעשיים הממצמצמים את הרוחה ואפיו יוצרים סיכון אם לא מנהלים נכון:

عملות מסחר ודמי פלטפורמה: عملות עלולות "לאכול" חלק או את כל מרוחה הארביטראז'. למשל, אם כל עסקה קרוכה בעמלה של 2%-1%, הפער $P_2 - P_1$ צריך להיות גבוה מכך כדי שהרווח הנקי ישאר חיובי. למקרה המול, פלטפורמת Polymarket נקבע ל-2025 אינה גבוהה מלת מסחר (fee) 5%, קר שעלות העסקה מתמחה בעיקר בעמלות רשת (גנ) ותשולם מרוחה מחיר. בפלטפורמות אחרות (כגון Kalshi) יש מלת מסחר או מלת סיום (fee על הזכיה) שיש להביא בחשבון. לפיכך, במסגרת מנوع החישוב (ראו סעיף 5) יש להחשב את העמלות מכל רוח תאורייתי ולודא שהרווח הנקי עומד בסף מינימלי רצוי.

עלויות גז ועסקה: בפלטפורמות מבזורת כמו Polygon (הפעלת על Polymarket) קיימים מחיר פעולה בראשת. הוצאה לפועל של שתי פקודות (קנייה ומכירה) קרוכה בתשלום גז. עלויות אלה **מקבות את הרוחה** בעסקאות קטנות, וכן יש לוודא שהARBITRAZ' "שווה את זה" גם לאחר הוצאות הגז. למשל, ב-ARBITRAZ', Polymarket עסקה טיפוסית עשויה לעלות عشرות סנט (תלוות עומס רשת ומחייב מטיק) 6. בוט ARBITRAZ' חכם ייחסב את הרוחה הצפוי ויפחית ממנה את עלות הגז, ורק אם ה-ROI הנקי עולה על סף מסוים (נניח 2%) העסקה תוכז כרווחית 7. לדוגמה, קביעה מובנית: "דרושים לפחות 2% תשואה נקייה לאחר עלויות" 8.

נזילות (Liquidity) וספר הפקודות: ARBITRAZ' התיאורי מחשיב מחירי שוק כאילו אפשר לבצע את העסקאות בנסיבות הרצוייה במחירים הללו. בפועל, ייתכן שرك מקומות מוגבלות של חזים זמינים במחיר היעד (בעומק Order Book הקיים). אם ננסה לבצע כמה גדולות, ניאלץ "לאכול" פקודות נוספות ונוספות פחת אטרקטיביות בספר הפקודות, ובכך המחיר הממוצע廉價 יעלה/למכירה ירד – זה **Slippage**. Slippage מקטין את מרוחה ARBITRAZ' או אף מבטלו. למעשה, מחקרים מראים שרוב הזדמנויות ARBITRAZ' החזרות (כמו ARBITRAZ' הקלנדייר בבחירה 2024) הופיעו ברגעי נזילות נמוכה ולפרקי זמן קצרים 10. כי בשוק דليل יתכו סיטות מחיר עד שסוחרים מאזינים. אבל באופן רגעים גם קשה לנצל כמה גדולות בעלי להזין את המחיר בעצמו. לכן, סוחר ARBITRAZ' צריך להגביל את גודל הפקודה למינימום שלא ישפיע משמעותית על המחיר, או לחילופין לקבל שימלא רק חלק מהכמות (Partial Fill) לפני העומק הזמן.

סיכון רגליים (Leg Risk): זהו אחד הסיכון המהותיים באrbitraz' לא-אוטומטי: חוסר יכולת להציג פעולה-זמןית את שתי רגלי העסקה. אם הסוחר הצליח לבצע את העסקה בשוק אחד ולא הספיק לבצע בשני לפני שהמחיר זז, הוא עלול להישאר עם פוזיציה לא מגודרת החשופה לסיכון כיוון. למשל, אם מכר "Yes" בשוק הקצר ולא הצליח לקנות "Yes" בשוק הארוך בזמן,icut הוא חשב לפוזיצית Short שאינה מובנתה רוחה, ויתכן שהמחיר בשוק הארוך עלה בינוים (או שהפער נסגר ע"י אחרים) כך שהעסקה השנייה כבר אינה אפשרית ברווחה. סיכון זה בולט כאשר אפשרות לבצע **ביצוע אוטומי** של הפקודות (עסקה אחת הכוללת את שתיהן יחד – לא קיים כאן כי אלו שוקים שונים). למעשה, **אי-אוטומיות הביצוע** היא מוגבלת קרייטית: "רجل אחת שעשויה להתמאל בעוד השניה נכשלה, מה שיוצר חשיפה כיוונית" 11. נרחב בסעיף 6 על שיטות לצמצום סיכון זה (כגון פקודות מיוחדות, ביצוע סימולטני וכיו"ב), אך לצורך חישובי כדיות יש להתייחס לכך שARBITRAZ' תאורייתי עם פער קטן עלול לא להתmesh בפועל בשל סיכון הביצוע. לכן לעתים דורשים **מרוחה ביטחון** גדול מהעלות

הטרנזה-קצינית המינימלית, כדי שאפילו במקורה מילוי חלקו או תזוזת מחיר קלה – עדין יישאר רוח. למשל, ייתכן שנציג ספרוות הארכיטראז' הגלמי יהיה לפחות 5% לפני עליות, אך שגם אם המהירים זים מעט במהלך הביצוע, יש שלדים לספוג זאת.

לúcטום, בעת זההו הזרמוות ארכיטראז' קלנדי, יש "להכניס למציאות" את הגורמים הללו: לחשב רווח נטו אחורי עלמות וגז, לבדוק את עומק השוק הזמן לכמות הרצויה (ואף לחשב רווח כפונקציה של כמות – ייתכן שכך לא הגביל את היקף העסקה), ולהעריך האם המרווח מצדיק את סיכון הרגילים. צעדים אלו ימנעו כניסה לעסקאות שבהן הרוח התאזרטי לא יתרגם לרווח ממשי.

4. **זהוי אוטומטי של שווקים תלויים – שימוש ב-NLP-clustering**

בפלטפורמת תחזיות עלולים להיות מאות ואלפי שווקים פעילים, שחלים למעשה דנים באותו אירוע באופן שונה (ניסוח שונה, תנאי זמן שונים, ספי ערכים שונים וכדומה). כדי לאטר הזרמוות ארכיטראז' לוגי באופן שיטתי, יש צורך בכלי **אוטומטי** לזהוי שווקים תלויים או חופפים. כאן נגישות לתמונה טכניקות **עיבוד שפה טבעית (Natural Language Processing, NLP)** (אשכולות) שנitin להפעיל על טקסט שאלות השוק.

אתגר זההו: לשני שווקים תלויים ייתכן ולא יהיה ניסוח זהה. למשל: - שוק אחד: "Will X happen by November 30?" - שוק שני: "Will X happen in November 2025?"

למרות שאלה אותה שאלת (השני ניסוח מעט שונה – "במהלך נובמבר" שקול ל-"עד 30 בנובמבר"), נדרש מנגנון שיבין שהכוונה זהה או חופפת¹². דוגמה נוספת: שוק אחד שואל "If Candidate Y wins the presidency, will they also win the popular vote?" ושוק אחר: "Will the popular vote winner be the same as the electoral college?" ושוק שלישי: "Will X win the election?" – ניסוחים שונים לאותה תמלות לוגית. כמו כן קיימים שווקים עם תלות אימפליצטיבית: למשל "Will X win the election?" ו-"5% margin of victory exceed 1%?" – כאן אם X מנצח מעל 5%, הוא בוודאות ניצח, ככלומר יש תלות מובנית (אחד מניה את השני). שווקים כאלה מכונים **סמי-תלויים** ועלולים להציג ארכיטראז' קומבינטוריה (לפחות קלאנדי אך לוגי דומה).

שלבי פתרון בזהוי אוטומטי:

1. **נורמל טקסטואלי (Text Normalization):** יש ללמד את המערכת להזות ניסוחים שקולים. למשל, להזכיר תאריכים בפורמטים שונים לצורה סטנדרטית ("להפוך "Nov 30" ל-"November 30" ו- "by" ל-"in" שניהם ל-"עד 30/11/2025"). ניתן להשתמש בספריות NLP לזהוי תאריכים (Entities DATE בסוג spaCy מסוג DATE בספריית spaCy) וב-**dateparser** (לחילוץ תאריך). כמו כן לנרגל מונחים: "by end of 2025" ו- "before 2026" – להבין ששניהם אומרי אותו דבר (עד סוף 2025).

2. **חילוץ ישויות ומאפייני אירוע:** נרצה לפרק את טקסט השוק לרכיביו: מ/מה האירוע, פעולה/שאלת (מה נמדד), ותנאים (כמו תאריך, סף מספרי, מקום). למשל בשאלת "Will Apple release a new product line in 2025?" ישות עיקרית – **蘋果**, אירוע – **苹果** – השקת קו מוצר חדש, זמן – **2025**. בשוק אחר: "Will Apple release a new product line before 2027?" – אוטם ישות ואירוע, זמן – **2027**. לאחר חילוץ זהה, ניתן ליצור **"טביעת אצבע"** **לאירוע** (Event Fingerprint) המתארת את הגרעין הלוגי ללא רכיב הזמן. בדוגמה זו הטביעת אצבע יכולה להיות ("Apple", "launch new product line"). שווקים בעלי טביעת אצבע זהה אך תאריך שונה זההו קובצות שווקים תלויים קלנדיות.

3. **השוואת טביעות אצבע וקיור סמנטי:** במקרים פשוטים, התאמת ישויות וshedeh airoyut יספיקו – ככלומר אם שני שווקים חולקים את אותם מאפיינים בלבד הזמן, יוכל ישירות לשדר אוטם צוג תלי (או לשרף אותו לקבוצה). עם זאת, שווקים תלויים לא תמיד מנוסחים סביב אותו ניסוח בדיק. ייתכן הצורך בזהוי סמנטי מתקדם: למשל "זהו שווקים" ו- "Will a Democrat win the popular vote and the Presidency?" – או "Will a Democrat win the Presidency?" ו- "Will a Democrat win the popular vote?" קשורים, אף שהנוסח שונה למעשה. כאן אפשר להיעזר במודול שפה להבנת ההקשר. שתי גישות אפשריות:

- .4. **הטענת מודל שפה גדוֹל (MLL)** על רשות השוקים, ובקשתו לפטר שוקים קשורים. מחקר עדכני אף מגדים שוק MLL הקורא טקסטים של שוקים, מציע צמדי שוקים עם תלות זהה או הופכה, ומקבץ אותם אוטומטית ¹³. ניתן למשת ל-MLL רשות שוקים באשכול (למשל פוליטיקה) ולשאול: "Group these into clusters" of related markets, and identify pairs where outcomes are logically linked (same outcome or opposite)" – גישה שהוכחה כבעלת דיקון סביר.
- .5. **הטענת משפטי השאלות לווקטוריזציה Embedding (SBERT)** והשוואת דמיון קוסינוסי/מרחך. מודלים ייעודיים כמו Clustering (DBSCAN K-Means, Hierarchical Clustering) על הוקטוררים כדי להזות קבוצות של שוקים דומים בשפה. לדוגמה, שוקים שכולים מזיכרים "Apple" ו-"product line" יתכנסו לאוֹשָׁאָל. יש לשימוש לב לשלב נרמול התאריכים לפני ה-embedding כדי שניי משפטיים הנבדלים רק בתאריך לא ייחסבו שונים מהותית.
- .6. **כלים נוספים:** אפשר לשלב גם חוקים ידניים ל'זיהוי near-duplicates: למשל התעלומות מפיסוק, אותיות רישיות, והמרת מספרים למיללים וכו'. בתחום ארביטראז' נצפו גם מקרים של שוקים כמעט זהים שנוסחו אחרת (למשל "Dec 30" מול "X happen by November 30" – שניהם שואלים אותו דבר למעשה, למעט הבדלי ניסוח זניחים של תאריך). כללים פשוטים יכולים לתפוס רבים מאלה.
- .7. **סינון וקבוץ:** לאחר חישוב דמיון או זיהוי ישויות, בניית **גרף או קבוצות** של שוקים תלויים. למשל, אם זהו 3 שוקים על "שיגור משימת Space" עד תאריכים 1, 2, 3 שונים", יכולים קובצו תחת אותו אירוע בסיס. יתרוננו גם זוגות תלויים לא קלנדריים (כמו הדוגמאות עם win vs margin), אוטם ניתן לסוג בפנורד – למשל קשר של "אותה תוצאה" או "תוצאות הופכיות" ¹⁴. חשוב גם להזות יחסים **בלעדיים**: לדוגמה משתמש במנגנון NegRisk שמביטה שוקים הם חלק מאירוע או-וריאנט רב-תוצאות Polymarket ¹⁵. שוקים כאלה לא נונטים ארביטראז' כי הפלטפורמה כבר דואגת לאrbitrage' אפס דרך אפשרות המראה בין פוזיציות. הפוקוס שלנו הוא על שוקים נפרדים (לרוב אירועים שונים או נוסח שונה) שיש ביניהם קשר לוגי ולא בהכרח מנגנון פנימי הקשורם.
- .8. **אימות היחסים :** לאחר זיהוי אוטומטי, כדאי לבצע בדיקה לוגית לתלות. NLP עלול לטעת במקרים ידניים – למשל שוק אחד "אם X יקרה עד T1" ושני "Y יקרה עד T2". יתרון שהם לא באמת תלויים (אם X-ו-Y שונים), או תלות חלקית (אירוע אחד משפייע הסתברותית על אחר). מערכת זהירה יכולה לדרש אישור ידני או חיווי confidence גבוהה לפני שימושו על צמד c-Pair Arbitrage. מחקרים שלבו גם בדיקה אמפירית – למשל בדיקת היסטוריות מוחירים של שוקים שהמודל טען שהם תלויים, לראות אם אכן היה מתאם תנועות או הכרעות חופפות ¹⁶. אנחנו יכולים להסתפק בכל התוכנה אם אנו מגבילים לתבניות הלוגיות פשוטות יותר (קלנדרי, על אותה שאלה; או שוקים שלכן ולא של אותו אירוע).
- השילוב של NLP, כללי נרמול, -Clustering מאפשר **סורך שוקים תלויים אוטומטי**. למשל, נוכל להריץ תהליך יומי שسورק את כל שאלות השוק ב-Polymarket API, משתמש בכל NLP כדי לティיג כל שוק ביישות/אירוע/זמן, ואז משווה את כל השוקים כדי לאסוף קבוצות. כך תתקבל מפת שוקים תלויים: כל קבוצה מייצגת אירוע בסיס (לדוגמה "השקת מוצר Apple חדש") וכל קבוצה מכילה את כל השאלות הנוגעות לאירוע (בזמן שונים או תנאים שונים). מתוור אלה, המערכת ת חפש הזדמנויות ארביטראז' | – למשל בקבוצה שיש בה שני שוקים קלנדריים, נבדוק אם המחרימים מפרים מונוטוניות; בקבוצה עם תנאי לוגי (כמו "X קורה" ו-"Z > Y במקורה X קורה"), נבדוק אם יש קומבינציית קנייה/מכירה שמביטה רוח (בדומה ללוגיקה שתוארה, רק עם יותר חישובים קומבינטוריים).
- מצין שבמחקר אחד הוגם שימוש ב-AI המבצע ממש את התהילך: Clustering של שוקים לנושאים, ואז **Relationship Discovery** בתוך כל אשכול למציאת זוגות שוקים עם תוצאה זהה או הופכה ¹³. גישה כזו (המשלבת הבניה סמניטית عمוקה ומערכות סטטיסטיות על שוקים שהסתויכו) הראה שאפשר לבנות צמדי ארביטראז' מסויימים ולהציג ביצועי מסחר חוביים על בסיס אוטם גלויים ¹⁷. זה מחזק את היתכנות הפתרון: NLP + Clustering הם כלים מרכזיים בבנייה Scanner חכם שיזהה עבורנו באופן אוטומטי הזדמנויות ארביטראז' קלנדרי ולוגי.

5. מנגע חישוב כדיות מבוסס Order Book (לא רק Mid-Price)

לאחר שזיהינו זוג (או קבוצה) של שוקים תלוים ופוטנציאלי לארביטראז', עליינו להחליט אם וכיצד לבצע את העסקה. כאן נcomes מנגע חישוב כדיות שתפקידו להעריך את הרוחה הפוטנציאלי הנקי מעסקת ארביטראז' נתונה, בהינתן **מצב סוף הפקודות הנוכחי** בשוקים הרלוונטיים.

מדוע לא מספיק להסתכל על מחירי-*Mid* (אוצר המרווח)? מפני שמחיר mid משקף רק ממוצע גס בין ה-Bid ל-Ask, ועלול ליצור אשליית רוחה במקום שאין בפועל. למשל, יתכן שבשוק T1 ה-price mid הוא 0.60 ובשוק T2 הוא 0.55 (נראה $P_2 > P_1$, הפרש 0.05). אך אם בספר הפקודות עומדת למשעה **היעץ** (Ask) בשוק T1 ב-0.62 וקונים רק ב-0.58- (Bid), ובשוק T2 היעץ ב-0.55 וה-Bid ב-0.53, הרי שושוחר ארביטראז' יצטרך **לקנות במחיר היעץ ולמכור במחיר הביקוש** בפועל. יתכן שההתמונה האמיתית היא אחרת: אולי כדי למכור Yes בשוק T1 קיבל רק 0.58, ולאחר מכן Yes בשוק T2 נשלם 0.55 – הפעם האפקטיבי 0.03 בלבד. לכן, חיבים להתבסס על **מחירי-Ask/Bid העדכנים בעומק הרלוונטי** כדי לחשב כדיות אמיתית ¹⁹.

צדדי החישוב במנוע:

.1 **שאייבט עומק שוק נכון:** עבור כל שוק בזוג הארביטראז', נשלוח את **הצעת הקניה הגבוהה ביותר (Highest Bid)** ואת **הצעת המכירה הנמוכה ביותר (Lowest Ask)**, וכן את הנפקים הזמינים ברמות מחיר אלו. למשל, בשוק T1: Bid=0.58 Ask=0.62 (ל-300 יח'), ובשוק T2: Bid=0.53, Ask=0.55 (יח'). Polymarket מפעיל הנתונים הללו זמינים לרוב דרך **API של ספר פקודות** או feed websocket רציף. הירידיו עם הזמן WebSocket שנוחתת עדכנים בזמן אמת על פקודות **CLOB (central limit order book)**.

.21 .20

.2 **בחירה תפkid בכל שוק:** נחליטஇeo פועלה לבצע בכל שוק. בדוגמה calendar arbitrage סטנדרטית: בשוק היקר קצר-הטווה נמכור Yes (כלומר נמכור במחיר-h-Bid הקיים, או נקנה No), ובשוק הארוך נקנה Yes (נסלים את-h-Ask). לעומת זאת, אם היזהוי היה $P_1 > P_2$ (שוק יקר יקר מהकצר באופן לא הגיוני), אז להפוך – נמכור Yes בשוק הארוך ונקנה Yes בשוק הקצר. בכל מקרה, נגיד פונקציית רוחה שתלויה במחיר הביצוע המשוערים: $\text{Profit} = \text{Cost} - \frac{\text{Profit}}{\text{Legs}}$ (Legs – מושג שמייצג את היחס בין מחירי-Ask/Bid ומחירי-Bid). אם הוא יפוג Yes (אם יפוג No) או על ה-No שקינו (אם יפוג No), קיבל \$1 על החוזה שקינו אם יפוג Yes, ונשווה לסכום שילמנו upfront. אך חישוב זה נעשה קודם מוקדם אבל עם מחירי-Ask/Bid במקומות P1,P2 תאורטיים.

.3 **התוצאות בנפח וסכום עסקה:** יתכן שהחזנות קיימות רק לכמות מוגבלת. למשל, אם בשוק T1 ה-Bid זמין רק ל-100 יחידות, ונרצה למכר 200, אז על 100 היחידות הנוספות כנהה נציגן לפגושים bids נוספים נמכרים יותר (נניח הבאים בתו 0.57 ועוד). מנגע חישוב חכם יבצע **אגרגציה של עומק**: הוא יכול לסכם כמה רוחה יופק בהנחה מילוי X היחידות. דרך אחת: להתייחס לכמות מינימלית הדרושה – למשל לדריש החזנות על לפחות 500 \$ ערך (כדי להצדיק מאמצ'ן). דרך אחרת: לחשב ROI (תשואה יחסית) כפונקציה של גודל: לעיתים עדיף למשם כמות קטנה עם ROI גבוה מאשר לנסוטה למקסם נפח על חשבון ROI.

המנוע יכול לבצע **סימולציה מילוי**: למשל, עבור כמות 100, לחשב שימוש מלא במחירים הנוכחיים (רוחות R1). עבור 200, לחשב שחקן ימולא במחיר הבא (רוחות R2), וכו'. ואז לבחור את הגודל האופטימי (למשל שמקסם רוחה נטו או לא יותר מתחת ל-ROI מינימלי).

בכל מקרה, המנגע צריך להחזיר את **כמות המומלצת** לעסקה ואת **הרווח (נטו) הצפוי**. אם הרוחה הצפוי קטנה ממהשף – ידחה את העסקה.

.4 **הכללת عملות וועלויות:** חלק מהחישוב, יש להפחת עמלות מסחר (אם יש), גז וכל הוצאה ישירה. למשל, אם ידוע שכלי טויד גובה 1% מהסכום המושך עמללה – נוריד את מחישוב הרוחה. בפטפורמת בלוקצ'ין, יש להעיר את עלות הגז. דוגמה: בוט יכול לשולף את מחיר הגז הנוכחי ואת מחיר מטבע הרשת (MATIC) ולה חשב שעלות שליחת פקודה ~gas * מחיר_gas. בוט ארביטראז' שהוציא במספרות מחשב את עלות העסקה בדולרים (ומציין שאין עמלות PolyMarket) ומודוד שהתשואה לאחר עלות עדין מעלה מונחים ⁸ .⁶

תשואה שנתית/פרק הזמן: שיקול נוספת – משך הזמן עד הכרעה. עסקאות ארביטראז' קלנדרי עושות להינעל עד תאריך היעד האחרון (T2). אם T2 רוחק, המשמעות היא שההון מושקע תקופה ארוכה. יתרון שהמערכת תרצה לדרש ROI גבוה יותר עבור עסקאות ארכוכות-טוווח, בהשוואה לארביטראז' שנגע תוך ימים ספורים. זאת מפני שיכולים להיות עלות הזדמנויות או סיכון מערכת לאורך זמן. למשל, ארביטראז' של 2% רווח הנגע רק לאחר שנה אولي אינו כדאי, בעוד 2% רווח ביום הוא מצוין. לכן, מנوع הcadיות יכול לחשב גם **תשואה שנתית משוערת** (Annualized ROI) ולסנו עסקאות מתחת לרווח מסוים. בפועל אצל סוחרים מקצועים, מסתכלים על "רבעון" או ליטאים שנתרו לארביטראז'. אינו התייחסות לכך שחלק מההזדמנויות מניבות "מעל 20%" לשנה²² אם מחשבים היחס מול הזמן²³.

דוגמה: אם נותרו 3 חודשים לפיקעה והARBiteraz' הנוכחי הוא 1%, אולי זה לא מאוד מרשים (כ-~4% שנתית). אם נותרו 3 ימים וניתן להרוויח 0.5%, זה עצום (פי כמה عشرות אחוזים שנתני). לכן, המנווע יכול לשקל גם מודד זהה, לפי העדפותינו.

קביעת סף כדיות: על בסיס כל הגורמים לעיל, הנהלה או המשמש המפתח יקבעו מדיניות: למשל "לא להיכנס לארביטראז' עם רווח צפוי פחות מ-\$50 או ROI נטו > 3%". מדיניות זו יכולה להיות מוקדמת. ואכן, באлогוריתם לדוגמה מגדרים `min_profit` או `min_roi` כפרמטר²⁴ .²³

מודל לחישוב כדיות: נסכם מודל פשוט: - קלט: שוקים A-B, כיוון העסקה (למשל "Short Yes B-Yes A-B"), כמות מקסימלית לניטוח (למשל עד 1000 יח'), ספי מינימום (ROI, רווח \$). - תהליך: - שלוף Bid/Ask ועומוקים. - עבור כמות q עד `max_q`: חשב מחיר ממוצע להタルילות q בכל גשל. נניח בשוק A (শমুরিম বন) נמלא q חלקו ב-Bid-Ask והטבות ואם נגמר נרד למחיר הבא; בשוק B (শকনি বন) נמלא q חלקו ב-Ak. אפשר לקבל סדרת מחירים (q_A, q_B) . - חשב עלות = $\$[(q_A - q_B) * q]$ (לדוגמה אם קנוינו q חזוי Yes בשוק B במחיר A_k , ומכוינו q חזוי Yes בשוק A במחיר A_k , הועלות: שולמו q_A וקיבלו q_B מהמכירה, אך כדי למכוור אולי יצרנו פוזיציה שחייבת q_A each, או פשוט נאכר ששמורנו $q_A - 1$ (התחייבות). אפשר גם לחשב ישירות את payoff בכל תרخيص, אך אומדן ניקח את q_A worst-case payoff = $\$1 - cost\$$. - cost\$ = $q_A * (1 - cost\$)$. דרישת לארביטראז' היא $\$0 \geq (q_A * (1 - cost\$))$ (שיהוא ≤ 0 בכל תרخيص), אנחנו רוצחים יותר מההעוצם $\$0 > (q_A * (1 - cost\$))$ ממשוערת. - הפחת עלות קבוצות + גז (לפי q). - חשב $q / cost\$$ – או, ומשך = $nowch = (T2 - \text{בדוק}) / 365$. - תנאי `min_profit`, `min_roi`, או פשטן אונאליזיה. - בחר q אופטימי או קבע שההזדמנויות לא כדיות אם לכל q הפרמטרים לא עומדים בסף.

פלט המנווע יהיה למשל: "Arbitrage detected between Market X and Market Y: buy Yes on Y at 0.55 and sell Yes on X at 0.58 (available 500). Suggested size: 300. Expected profit: \\$12 (2.4% (available 300) - 0.55 * 300) = \\$0.36. Risk: low. Fees accounted for opportunity". – אם נחפוץ, או פשוט "opportunity" שנשלחה להלה.

למעשה, בתשתיות של בוט ארביטראז' אמיתי, המנווע הזה הוא הלוגיקה שמקבלת עדכוני מחירים ומחליטה **Trigger** (הפעלת ההוצאה לפועל) כאשר יש `match` לקרייטורונים. בסעיף הבא ניגע בבייז – שהוא השלב לאחריו וርכום הcadיות.

6. מנווע ביצועיעיל ומפחית סיכון – Execution Engine

כאשר זההה עסקאות ארביטראז' כדיות, יש לפעול מהר על מנת "לנעול" את הרוחות לפני שהשוק ישתנה. מנווע הביצוע אחראי על שליחת פקודות לשוקים, תזמון וסטרטגיית הביצוע, וטיפול במצבו של חלקי. הדרישות העיקריות: **מיהו, סינכרון, ואמינות**.

ביצוע מקבילי vs סדרתי: אידיאלית, נרצה לבצע את שתי רגלי הארביטראז' בו-זמנית ככל האפשר, כדי לצמצם סיכון חישפה. בפועל, אי אפשר ממש לשולח שתי עסקאות באותו פיקט רשות בדיקן, אבל אפשר להתקרב לכך על-ידי שליחתן במקביל אסינכרוני. ב-`Python` למשל, ניתן בעזרת `gather` (gather(asynio.os) לשולח בלהשות קניה ומכוונה סינולטנית²⁵ . קר, אם שתי הפקודות מוצאות מיד צד נגיד, שתיהן ימולאו כמעט בו-זמנן. זה עדיף בהרבה על גישה סדרתית ("בצע, המtan לאישור, בצע B"), שבה השוק שני עלול להשתנות בזמן הפעור.

לכן, האסטרטגיה היא: שלח שתי פקודות Limit (אחת מכירה בשוק A במחיר Bid הנוכחי, אחת קנייה בשוק B במחיר Ask הנוכחי) עם IOC = In-Force. כך כל פקודה ת מלא מיידית מה שnitן ותבטל את היתרה. אם שתיהן מתמלאות 100% מูลה – הארבעיראץ' בוצע במלואו.

ניהול חלקי (Partial Fill): אם קרה תרחיש שמקודם לעליו: אחת הפקודות מולאה חלקיית או לא מולאה כלל, והשנייה מולאה חלקיית או מלאה.icutת אנו עם פוזיציה ועדפת הצד אחד. מנוע הביצוע ציריך **לטפל מיידית** במצב זה כדי לא להשאיר סיכון - אם הייתה מילוי חלקי - למשל הצלחנו למכור 100 יחידות ב-A אבל קנוינו 0 ב-B (כי אולי Ask-the-h-Ask נעלם לפני שהפקודה הגיעה),icutת אנו Short Ichidot ב-A. המנוון יכול לנוסות שוב לקנות 100 ב-B במחיר מעט גבוה יותר (אולי יש Ask חדש יוטר יקר אך עדין מתחת למחיר שמכרנו). אם המחיר עלה מדי (כבר אין רווח), אולי עדיף לקנות בחזרה את Ichidot ב-A כדי לסגור את השורת (להזות בהפסד קטן) – זה בעצם ביטול העסקה אחרת. - אפשרות אוטומטיות: ברגע שמתברר שركן אחד מולא, נשלח פקודה ביטול מיידי לצד השני (אם עוד לא בוטלה אוטומטית) ²⁶, ואז **נסגור את הפוזיציה באופן מיידי** בשוק הראשון. "נסגור" אומר: אם נמכרו לנו X חוזי Yes, נקנה אותן חוזה בשוק A (אפיו במחיר מעט גרווע יותר) כדי לצאת ניטרלי, או אם קנוינו יותר מדי בשוק B, נמכור אותן שם. זה כמובן עלול לגרום להפסד קטן (כי ארבעתראץ' נכשל והמחיר אולי ז'), אבל זה חלק מניהול סיכונים – לקבל הפסד מזערני במקום להיאשר חשבו.

מימוש לדוגמה בגישה קוד: המנווע שולח שני פקודות במקביל עם timeout של נניה 5 שניות. אם בתוך 5 שניות שתיהן לא אישרו מילוי, או אחת לא – הוא מבצע cancel למה שנשאר פתוח ומיד פועל לסגור מה שהתאפשרה.²⁷ פרק זמן קצר (מספר שניות) חשוב כדי לא ליפול במצב של המתנה ארוכה מדי – נתונים מראים ש-75% מהפקודות בשוקים הללו מתמלאות תוך כשעה (950 בЛОקים בפוליגון)²⁸, אבל אנו לא רוצים להחכות יותר מכמה שניות כשבושים ארביטראז' מהירות. פקודות שבסדר עדיפויות ראשונות בדרך כלל יתמלאו מייד, ואם לא – כנראה שמיישה הקדים אותנו או שהשוק זו, ואז עדין לסתם מיד.

יעילות ומהירות: יש לוודא שלמנוע הביצוע יש גישה ישירה ומהירה ל- API המשחר: - ב- Polymarket Python ClobClient המאפשרת לשילוח פקודות חתוםות יישורות לשרשרא דרך שרת התאמה .²⁹ יש להשתמש בחיבורים מהירים (למשל WebSocket) כדי לקבל אישור מיידי על הביצוע או ביטול. - בנוסף, ניתן שכך לפצל את הביצוע למספר חלקים אם הכמות גדולות: במקרה שלוחות 1000 יח' בפקודה אחת, אולי לשוחות 2 פקודות של 500 במקביל (אם אחת לא מתממשת. לפחות סגנון חצי וכו'). אך זה כבר אופטימיזציה מתקדמת.

Retry Logic: מה אם העסקה נכשלה בגלל עניין טכני – נניח הבוט שלח את הפקודות ונותק מהאינטרנט, או WebSocket נפל, ולא קיבל אישור? מנוע הביצוע צריך להיות מחוסן: - עקוב אחריו מצב הפקודות: אם לא התקבלה תגובה, בצע שאלתיה יזומה לסטטוס (פוליג'ינג דרך REST) ולפי זה החלטת. - אם בוצע cancel אך עדין יש מילוי חלקי, ועוד שוגרת. - במקרה והארביטריאז' עדין קיים (נניח אחראי ביטול המחרירים חזרו לפער), אפשר לנסות שוב. אבל להיזהר לא להכנס לLOOP אינסופי אם השותן צד אחד.

Parallel vs Serial: ברוב המקרים, **Parallel** (סימולטני) הוא העדיף, שכן כל מילשינה חשובה. עם זאת, יש מקרים גבולים שאולי עדיף סדרתי: למשל, אם פער הארביטראז' גדול מאוד וברור, אפשר קודם לבצע שוק אחד ואז השני אם יש סיבוב טכניות (כמו שהן אוחזין זמינים יותר). אבל לרוב – לא.

סוגי פקודה מתקדמים: אם פלטפורמה תומכת, ניתן אף לשולח פקודה מותנית: "מכור ב-A רק אם נקנה ב-B" – אך נדרש שיש API זהה, אלא אם משלבים חזה חכם ייעודי לאוטומיות (לא זמין כאן). לכן אנו נצמדים לניהול התוכנתי.

לסיכום, מנوع הביצוע: - שולח פקודות IOC/IOLimit לשני השוקיים במקביל²⁵ . - ממתין חלון זמן קצר (שניות) בודק סטטוס: אם שתי הפקודות מולאו מלא – הצלחה. - אם לא: מבטל את מה שבעוד פתוח²⁶ . - מחשב את המילוי החלקי בכל צד³⁰ , ומבצע **handle_partial_fill** – סגירה של הפויזיציה הפתוחה (למשל שלילה לשוק הנגדי כדי להתכסות). - מדוחה חזרה הצלחה/כשל. אם כשל – יכול לסמן שלא מומש (אולי ינסה שוב כשייה שינוי אם עדין כדי תלוי לוגיקה).

גישה זו ממצערת למינימום את הסיכון שניתקע עם פוזיציה לא-מוגדרת. היא טריד-אוף: לעיתים פקודה הייתה מתמלאת אם חיכנו 30 שניות, אבל אנחנו ביטלו אחריו 5 שניות. אולם-arbitrage'רים, אנחנו לא מהמרים – מעדיפים לא לקבל מילוי מאשר לקבל חצי מילוי.

הערה: Polymarket היא כאמור על רשות Polygon, אז הביצוע הסופי של סגירת העסקה הוא chain-hops. זה אומר – שאנו אמורים שניים מולאו, היישוב של הטריד מתרחש בתוך בלוקים (לא מייד ברמת *finality*). אבל זה כבר שkopf – אנחנו סומכים על כך שאם קיבלנו אישור מילוי, אז הפויזיציה קיימת (הטוקנים אצלנו).

עוד טרייק: אם הפלטפורמה הייתה מאפשרת **Atomic Swap** בין שוקיים (למשל חוזה שתופר שני שוקיים), זה היה פוטר את הסיכון. Polymarket מאפשר *atomic swap* בין TOKENS תוצאה לקולטרל עבור שוק יחיד²¹ , אך לא בין שוקיים שונים.

לסיום סעיף זה: מנוע ביצוע מהיר ומוגן הוא קריטי. בעזרתו שימוש בפקודות FOK/IOC, זמן תגובה של חלקי-הזמן שנייה, Retry/Cancel אוטומטי, נוכל להתמודד עם אופי השוקיים הללו – **תנדתיים ומהירים**, בפרט סבב אירופים (לדוגמה: תוצאות סקרים או לאומיים פוליטיים, שגורמים לתנודות ומצחים של אי-יעילות זמן²). ללא מנוע מבחןיע, הזדמנויות שנמשכות שניות עלולות להתפסס או להפוך לסייעון.

7. ניטור בזמן אמת ואסטרטגיות יציאה

מנגוני ניטור (Monitoring): כדי שהמערכת "תדע" מתי להגיב, עליה להיות מחוברת בזמן אמת לשוקים. שימוש ב- **WebSocket** הוא הדרך המועדפת: Polymarket למשל מספקת ערוץ WS של עדכנים לכל שינוי בספר הפקודות ובמחירים³¹ . המערכת שלנו תרשם לעדכנים על כל השוקיים הרלוונטיים (או על כל השוקיים אם נרצה). כאשר מגיעו שינוי (למשל פקודות חדשה שומרינה את h-Bid בשוק אחד, או ביטול פקודה שומריך את Ask בשוק אחר), המערכת תחשב מיד אם נוצרה הזדמנות-arbitrage'. **זמן תגובה מהיר** קריטי: הארביטראז'ים הבוטוחים לרוב נסתימים תוך שניות על-ידי אחרים, لكن על הבוט לזהות ולפעול מיידית³² .

מבחינת מימוש, נוכל להציג תהליך **Market Monitor** הרץ בולאה אסינכרונית, מאזין דרך the-WebSocket, וمعدכן מבני נתונים פנימיים (למשל Map של State OrderBook->OrderBook state). בכל עדכון Ask/Bid/Bid, הוא יכול לסמן את אותם שוקיים כ"צריך בדיקת-arbitrage' מחדש". המערכת תבדוק את אותם שוקיים מול בני הזוג שלהם (שבעזרת סעיף 4 יודעים מהם).

בנוסף, ניטור כולל גם **מצב הפויזיציות שלנו**: אם נכנסנו לעסקת-arbitrage' (נניח מחזיקים כעט X וחומים פה ושם) – נרצה לעקוב אחרי שינויים שעולמים להצדיק תגובה (זה מתקשר לאסטרטגיית יציאה, בהמשך).

עוד אלמנט – **אירועים חיצוניים** : לעיתים יש הכרזות ידועות (כמו פרסום DO'H, תוצאה) שיכולות לסגור-arbitrage' בזעם. טוב שהמערכת תהיה מוכנה מבחינת נזילות (ואולי תימנע מכניסה ממש ונגע לפני אירוע אם חוששים מהתנדזה חדה). ניתן לשלב ניטור חדשות או LO'Z(arbitrage), אך זה מעבר לטווחה הנוכחי.

אסטרטגיות יציאה (Exit Strategy): לאחר שביצענו עסקת-arbitrage', לכורה הדרך הבוטוחה היא **להציג את הפויזיציות עד הסיום (Settlement)** . במקרה של-arbitrage' קלנדי, המשמעות להציג את שני החזים עד שתווקףם יfrag והם יושבו אוטומטית. מכיוון שבנו את הפויזיציה כנייטרלית סיכון, יוכל פשוט להמתין: – אם הארור לא קורה עד

T1, נקבל תשלום על חוזה-hSe שלנו ב-T1, והמשיר להחזיק את חוזה-hYes של T2 עד T2 (אולי נקבל תשלום ב-T2 אם קרה בין לבון, או לא). - לא משנה מה, בסוף T2 נקבל את כל מה שמנגע לנו, וסכום זה יהיה => עלותנו (כמובן).

יתרון החזקה עד הסוף: מובטח הרוחות כפי שהסבירו, ללא תלות בשום דבר (מלבד שהפלטפורמה אכן משלהם). אין צורך בשום פעולה נוספת, ואין חשיפה למחיר שוק רגעי.

חיסרונו החזקה: הכספי (USDC וכד') "נעול" בפוזיציות עד תום האירוע, שיכל להיות רחוק. למשל, ארביטראז' על בחירות 2024 שהתגלתה בנוב' 2023 – תאלץ להמתין עד נוב' 2024 לקבל את הרוחות המלא. בזמן זהה אפשר היה להשתמש בהן לעסקאות נוספות. בנוסף, החזקת פוזיציות זמן רב דורשת אמון שאין תקלות (נדיר אך אפשרי – בהמשךណון בסיכוןים כמו שינוי תנאי שוק).

יציאה מוקדמת כאשר הפער נסגר: האלטרנטיבה היא **lezatת הפוזיציה לפני הפירעון**, ברגע שכבר אין יתרון לחזקה. איך ומה? – ברגע שהשוק מפנה את היחס הלוגי והפער במחירים נסגר (או התההף). אם לדוגמה קניינו Yes ארכו ומכרנו Yes קצר, והנה לאחר זמן מה המחיר השתנו כך ש- $P2\$ < P1\$$ Cut – יתכן שאין עוד סיבה מיוحدת לחזקה. למעשה, אם $P1\$$ ירד מאד- $P2\$$ עליה, יתכן שאפילו תוכל לסגור את שתי הפוזיציות ברוחות גדול יותר מהMOV. נגיד קניינו T2_Yes ב-0.50 ומכרנו T1_Yes ב-0.60. אם אחראי שבוע השוק התישר ל- $P1=0.50$, $P2=0.55$ (עכשו סביר), נוכל לקנות חוזה T1_Yes ב-0.50 (לסוגר השורט ב-Yes, או למכוור את-hSe שיצרנו שם) ולמכור את T2_Yes ב-0.55. בכך נסגור את שתי הפוזיציות בזמן: קיבלנו 0.60 בהתאם למכירת T1_Yes, והזינו 0.50 Cut לקנות אותו חוזה – רוחות 0.10 שם; קניינו T2 ב-0.50, מכרנו Cut בת-0.55 – רוחות 0.05 שם; סה"כ 0.15 רווח, לעומת 0.10+0.90 יתרה (הינו מחייבים לתרחיש הגרוע. אמן בתרחיש האמצע (הARIOU קורה בין T1 ל-T2) הינו מרווחים יותר מאשר אפשרי), אבל זה לא בטוח שיקרה. - **ניהול סיכון:** יציאה מוקדמת מסירה אפילו את הסיכון המרכיבי של תקלות (כי כבר אין פוזיציה).

מתי הגיוני לצאת? - **אם הפער נסגר למחרי** : ככלומר עכשו הפוזיציות שאתה מחזק נסחרות במחיר שמשקף בבדיקה און ארביטראז'. אז הפוזיציה שלך כנראה שווה בבדיקה כמה שシリמת, או מעט יותר, אז אתה יכול לצאת לפחות לפחות ללא הפסד. אוליעדיף להישאר רק אם יש עדין רוח עתידי. - **אם הפער התההף או over-corrected**: יכול להיות אפילו מקרים שבו $P2\$ < P1\$$ Cut ממשמעותית (כלומר היה overshoot). אז למשעה אתה יכול לסגור ברוחות גדול מהMOV. זה כמובן האידיאלי לצאת. בשוק ייעיל, אם אתה היה הגורם שתיקן את המחיר, כנראה הפער פשוט נסגר הדוק ולא מעבר. אבל לעיתים תנודתיות יכולה לגרום להיפוך קל. - **פני אירוע מיוחד גדול**: יתכן ותחליט לצאת מוקדם אם מתפרק אירוע גדול שעלול לגרום לתנודה חריגה במחירים (למרות שבכל תרחיש אתה מוגן, אבל לפעמים ספציפיות עלולים לגרום לדברים אחרים – כמו עצירת מסחר, שינוי תנאים, וכו'). למשל, חלק מהסוחרים היו יוצאים מפוזיציות cross-platform לפני תוצאותARIOU כדי לא להיות תלויים בהכרעות או רקל'ל שונות (ראו סעיף 8 על מקרה זה).

הגדרת לוגיקת יציאה במערכת: - ניתן לקבוע שהמערכת תנסה **לשגו פוזיציות פתוחות** אם המחיר בשוק השתנו כך שרוחות הסגירה הנוכחי $=$ רוחות מובטח מסויים. בפועל, כל: אחרי פתיחת ארביטראז', להמשיך לנטר את אותם שוקים. אם מתקיים $\$newch_{P2} < \$newch_{P1}$ (שיעור משקל), אולי פשוט סגור. - לחילופין, אפשר לקבוע "נצא אם נוכל לנערל לפחות X% מהרווח המקסימלי רגעים". – או "נצא אם נותר Zeit זמן לפקיעה והפער נסגר, כדי לפנות הון".

יש גם אפשרות של **יציאה חלקית** : למשל אם היו שלושה שוקים, אולי נסגור חלק אם אחד נותר. אבל ברוב המקרים זה זוג.

לבסוף, יש אסטרטגייה של **"הפער ציון"** : נניח לאחר תקופה, השוק התההף כל כך שעכשו יש הזדמנויות ארביטראז' בכיוון ההפער. אפשר תאורטית לבצע ארביטראז' הפער (Long הפוזיציות הפוכות) שיבטל את הקודות ויוצר פוזיציה חדשה. אך זה בגודל אותו דבר כמו לצאת ואיז להיכנסשוב.

דוגמה פרקטית: בבחירות 2024 צוין שקיימיםCMDI שוקים תלויים שהARBITRAZ' בהם הופיע פעמים רבות (6,630 פעמים בזוג מסויים!) אך ברוחות נמוכה¹⁰. המשמעות היא שהפער היה נסגר מהר או קטן. יתכן שסוחרים ללחוץ פוזיציה ויצאו בעבר דקות כנסגר. המספר הגבוה מעיד ש**כניסה ויציאה מהירה** (תוך שעה או פחות) התרחשה שבעם ושוב¹⁰. זה מתאים לגישה אקטיבית: לנצל את הפער, וכשהמחרים מתyiיררים (אפילו לפני ההכרעה הסופית החדשים אחר כך) – למשר רווח ולחפש את הזדמנויות הבאה.

בשורה התחתונה, **אין חובה להחזיק עד הסוף** אם אפשר להרוויח מוקדם. החזקה עד הסוף نوعדה להבטיח את הרווח המינימלי. אם השוק נותן הזדמנות לצאת ברווח – לא הנסס, אך נשרר את ההון לעסקאות נוספות. ההחלטה תלויות גם במידת הביטחון שלנו בסגירת פוזיציה. אם השוק עמוק וקל למכור/לקנות חוזה, ניתן לסגור. אם חששנו מנזילות נמוכה (קשה לצאת) אולי מילכתחילה לא היינו נכנסים.

המנוע בפועל יכול להיות מותכונת: - לעדכן כל הזמן את ה- "L/P unrealized" של כל פוזיצית ארביטראז' פתוחה לפני מחירי השוק, - אם התנאי מתקיים ($\text{threshold} = \text{L/P} >= \text{ROI} \%$), לשולח פקודות סגירה (sell מה שקנינו, back buy מה שמכרנו). - אחרת, אם מתקבבים לפיקעה, אולי לחכות לפיקעה.

כמובן, תמיד יכולה להיות אפשרות של **כשל שוק** או ארווע חיצוני (למשל השוק לפתע מוקפא או נסגר מוקדם), ולכן לא תמיד נרצה להישאר עד הרגע האחרון – אך נושא זה נכלל בדיון ניהול סיכונים הבא.

8. ניהול סיכונים ותרחישי קצה

למרות שארביטראז' מתימר להיות "לא סיכון", במצבות מערכות מסחר ושווקים יכולים להציג סיכונים בלתי צפויים. נפרט כמה מקרים ודרכי התמודדות:

שוקים עם תנאים שונים או ניסוח מבלבל: לא תמיד שני שוקים "דומים" הם למעשה תלויים מלא. יתכן שגם B סמי. למשל: שוק X win the presidency by T1 - "Will candidate X be president?" - ?A - "Will candidate X be president by T2?" בשני המקרים X צריך להיות נשיא, אך יתכן הדבר: הראשון שואל על ניצחון בבחירות עד תאריך (אولي יום הבחירות), והשני שואל אם ייחסק בנסיבות בפועל עד תאריך (כולל אפשרות מות/Franchise ביניים). זה דק – ומילא קורא את כללו השוק עלול להניח תלות מוחלטת כשלעצמה. **ניהול סיכון:** המערכת צריכה להשווות לא רק את טקסט ההוראות אלא גם את **החוקים (Rules)** אם זמינים-ב-API. לעיתים, שם נמצא פרט מבדיל. אפשר למשל לדרש שני שוקים יאצרו אותו "Resolver" או מקור הכרעה, לבדוק אם יש סעיף הגדרות שונים. במידה ויש ספק, **עדיף לארביטראז' האנושי לבדוק** או פשוט להימנע. בשלב הראשוני, אולי נמקד לARBITS זרים קלינדרים ברורים (אותו שאלת בדיקת, תאריך שווה). אם בכל זאת נמצאה הזרמת עלי שוקים שנראים תלויים אך עם שונות קטנה, ניתן לסמן אותה להחלטה ידנית או לבדוק אותה מוגברת. דוגמה מחקר: נמצאו **13 זוגות שוקים תלויים** בבחירות 2024, כולל יחסים מורכבים (כמו פער בין פופולר וט לאלקטורל קולגי וכן "אייזו מפלגה מנחת באיזו רמה")³⁴ . שם ננקט משנה זהירות באימוטות תלות. חלק מהשוקים הוגדרו "weak dependency" – תלות חלשה (למשל MLM זיהה תלות אך בפועל לא תמיד נכון)³⁵ . המערכת צריכה לודעת להזות ולסנן כאלה. עוד דוגמה: **שוקים מותניים** – נניח שוק אחד הוא "if Will X happen wins election Y?" ושני "Will X happen?" – פה התלות היא הסתברותית ולא דטרמיניסטית כי תלוי אם Y באמת זהה. אלו מצבים מורכבים – נראה לא ניגע כי אין ארביטראז' בטוח, רק שינוי הרכבת הסתברות.

תקלות WebSocket/API: תשתיית התוכנה שלנו עלולה להיתקל בעיות: ניתוק מהרשת, קיריסט חיבור WebSocket, או תקיעת התהילך. אם בזמן זהה הופעה הzdמנות, נפסס אותה. גרוע מזה, אם זה קורה באמצעות **ביצוע עסקה** (נניח שלחנו פקודות ולא קיבלנו אישור), המערכת עלולה לא לודעת שיש לה פוזיציה פתוחה. **ניהול סיכון:**

- שימוש בחיבורים עם Auto-reconnect heartbeat. למשל, ספריית **websockets** מאפשרת לגלוות ניתוק וניסות להתחבר מחדש. בינהית, אפשר לבצע **Polling** (שאלת ה-API כל שנייה) עד שה-WS חזר.
- ריבוי ערוצים: אולי חיבור WebSocket ווסף כגבוי.
- לוגים וניתוח חיצוני: להריץ את הבוט עם השגחה (למשל מערכת שידעת להתריע אם הוא לא שולח הודעות זמן מה).
- בעת כשל, המערכת צריכה **לא לבצע פעולות מסחר חדשות** אלא אולי רק לדאוג לסגור פוזיציות במידת הצורך.
- הגינוי להקפיא מסחר אוטומטי אם מזהים שזרימת הzdמנות לא אמינה.
- לגביו ביצוע: לאחר שליחה, המנון יכול לפותח Timer: אם לא שמעתי כלום 5 שניות, נניח כשל ובטל. כמו כן, יש לשמור את ה-IDs Order שנשלחו ולודוא בסופו של דבר (דרך קריאת API מצב הzdמנות) שלא נשארו פתוחות.
- בקוד לדוגמה רואים שימוש ב-`timeout` asyncio של 5 שניות³⁶ והגעה `except` במצב `TimeoutError` שמבטל הכל³⁷.

שינויים פתאומיים בשוק: זה יכול להיות

תנודה חזה במחיר: למשל, מישו עם מידע חדש נכנס וקנה הרבה חוזים, המחרים זים ממשמעותית. אם קרה בדיקות כשאנו מבצעים, כבר דיברנו איך לבטל ולטגת. אבל גם לאחר שאנו מוחזקים: לדוגמה, קניינו Yes Yes ארוך ומכרנו Yes קצר, אז פתאום הגיע מידע שהופך את האירוע לכמעט ודאי שלא יקרה – שני השוקים ירדו במחיר חזק. לכואורה אין לנו מה לחושש כי החזקתו מובטחת עד הסוף, אך אולי השוק הופסק זמן קצר עקב עדכון? Polymarket

השאית מסחר/Low liquidity: אם מסיבה כלשהי אחד השוקים מוקפא (או行政 intervention admin, או תקופת מחלוקת אורקל), אנחנו לא יכולים לסגור פוזיציות או לגדר עוד.

שינויים בשאלת/תנאים: יתכן נדר – אם הפלטפורמה מגלת שנות השוק היה שגוי ומתknת/معدכנת אותו. למשל, "תאריך T2" הוגדר 31/12/2025 ובנסיבות תקנו ל-01/01/2026. שינוי זה יכול לשנות מעט את ההיקף. **ביטול שוק (Invalid Market):** חלק מהפלטפורמות, כולל Polymarket, שומרות לעצמן (דריך מגננון אורקל) את הזכות לסגור שוק כאשר "לא היה" (למשל אם השאלה לא הכרעה ברור). במצב זהה, שני השוקים אולי לא יתיישבו כפי שחשבנו. לדוגמה, היה שוק "Will X happen by Dec 2025?" – ננעל כל רלוונטי כי האירוע בטול, אז הכריזו על Invalid (ה היזרו 50-50 לכל מחזק). אם השוק האחרון עד 2026 נשאר רגיל – האסטרטניה שלנו נהרסה: במקום \$1 או \$0, נקבל החזר 0.5\$ על אחד. **ניהול סיכון:**

לבעור הטוב על חוקי השוקים ולודא שאין סעיף "A/N If unclear or event cancelled, market resolves invalid". אם יש סיכון זהה, אולי לא להיכנס או להיות מודע.

לפזר סיכונים: לא לשים את כל ההון בארביטראז' אחד, במילוי אם יש חשש משפטי/טכני. להשתמש ב-**Position Tracker** חכם: לטעד כל פוזיציה שיש לו (כמות ונכס בכל פלטפורמה), ולקnow אם יש ארוע שעולם להשפיע. למשל, בוט הארביטראז' cross-platform שמר מבנה של כל פוזיציותיו ובודק אם אחת תלוי באורקל שונה ³⁹. הוא אף זהה אם יש פוזיציות חוזרות פלטפורמות שעולות להיות מודע ⁴⁰ ⁴¹ ⁴².

במקרה של Polymarket, לאירועי settlement יש תחיליך (AMA אורקל) שיכول להיות במחלוקת. היו מקרים, כמו במרכז 2025, שימוש רב-השפעה בתערובת הצבעה וגורם לפתרון שגוי בשוק מסוים ⁴³. סוחר שיגיד אורכו מול Kalshi (שם נפסק נכון) הפסיד. דרך להקטין סיכון זהה: להימנע מארביטראז' cross-platform אלא אם הapur עצום (כגביו לסיכון) ⁴⁴. בתוך אותה פלטפורמה, סיכון זהה קטן יותר (כי שני השוקים היו מוכרים יחד לרוב אם משוח קורה).

הימנעות שוקים "מועדים לفورענות": אם זההינו שוק עם ניסוח עיתוי, או נושא שני בחלוקת (העלול לגרום לויכוח על התוצאה) – אולי עדיף שלא להיות שם.

מעקב אקטיבי: אם אנו מוחזקים זמן רב, רצוי שימושו (או תחיליך) יבדוק באופן תקופתי שהשוק עדין ח'י, לא הוכרז כ-*invalid* וכו'.

תוכניות גיבוי במערכת:

אם רכיב אחד קורס (נגד the-gong NLP Clustering התבלבל ונתן קשר שגוי) – רצוי הגנה מפני ביצוע עסקה חסרת-תלות באמת. אפשר למשול לפני כל ביצוע להשותות מוחורי שוק: במצב היגיוני, אם באמת P2 > P1, הארביטראז' יהיה רוחוי. אם קיבלנו סימן ארביטראז' אך למעשה P1 לא באמת < P2 (אולי המידע הישיר היה שגוי) – לא לבצע.

אם ה-*Executor* מחזר כשל – המערכת תסמן הזדמנויות זו כנכשלה ולא תנסה שוב מיד (אולי תחכה לתחזות מחיר ממשמעותית לפני ניסיון).

Rate limiting: כדי להימנע ממצב שהבוט מציף את ה-API בעבודות (מה שועלול לגרום חסימה או עמלות מיותרות), נגיד תקרות: למשל לא יותר מ-X עסקאות בשנייה. נתון: Kalshi מגביל 20 קריאות/שנייה (קריאה) 1-10 כתיבות/שנייה בסיס ⁴⁴. Polymarket אולי פחות מגביל אך יש עלות גז – בכל מקרה נשלב בקורת קצב.

סיכום מערכתי/רגולטורי: אולי פחות טכני – אם שוקי התחזיות משתנים (לדוגמא רגולטור מגביל שוקים מסוימים, פתאום Kalshi נאלץ לסגור שוק...), זהאתגר. אין פיתרון ברור בקדוד, אלא מודעות ותגובה (נניח יצאה יזומה לפני תאריך סגירה צפוי).

לסיכום חלק זה: ניהול הסיכון משלים את האלגוריתם כך שלא נסמן עיוור על "רווח ללא סיכון". הוא כולל: - בדיקות עקביות ולוגיות טרם כניסה (מניעת **arbitrage**, false), - מעקב קפדי אחרי ביצוע (חירום בזמן חלקיות), - פיזור ואיזוני כניסה (לא הכל בהזמנה אחת), - ניטור מתמשך של פוזיציות פתוחות ושינויים בשוקים (כולל מנגנוןים ייחודיים כמו **invalid**), ותוכניות יצאה מהירות בעת תקלה.

בגישה שמרנית, גם לאחר הэнטיה האלגוריתם, נרצה לפעמים פיקוח אנושי, לפחות בשלבים הראשונים, כדי לוודא שהמערכת לא לוקחת סיכון חריג עקב מצב שלא שקלה.

9. ארכיטקטורה מודולרית מוצעת (Scanner, Normalizer, Pricing וDetector, Executor, Risk Manager)

כעת נגד את הכל לבנייה ארכיטקטוני כולל, מודולרי, שבו כל רכיב אחראי על חלק מהממשק. ארכיטקטורה מודולרית מקלה על פיתוח, בדיקה ותחזוקה - ניתן לשפר רכיב מבלי להשפיע על אחרים, והפרדת האחריות ברורה. להלן המודולים העיקריים וחומרת הנתונים ביניהם:

- Scanner (סורק שוקים):** מודול שאוסף מידע גולמי מהפלטפורמות. הוא מתממש ל-**API** הציבורי של שוק, התחזיות כדי לקבל את רשימת כל השוקים הפעילים, את הפרטים שלהם (គותרת, תיאור, זמן סיום, נזילות, מחירים הנוכחיים וכו'). הוא מערנן באופן תקופתי (נניח כל שעה) את רשימת השוקים החדשניים או את המכידע הסטטי (לא משתנה בתדירות גבוהה). ה-**Scanner** מספק את הפלט הנתונים ל-**Normalizer**.

- Normalizer (מנormalיזר ו犹וצר אינטירptr):** מקבל את רשימת השוקים ופרטייהם מהסורק. עברו כל שוק, מבצע שימוש שפה: חילוץ יישויות (שמות, תאריכים, מקומות), נרמול תאריכים לפורמט אחיד, זיהוי מילים שקהלות synonyms). לאחר מכן, יוצר ייצוג מובנה לכל שוק - למשל אובייקט עם שדות: **event_subject** (אם יש כמו ערך מסוים), **event_time** (אם יש כמו ערך מסוים), **event_threshold**, **event_action**. גם יתרכן שדה **fingerprint_id** אוvr, המהווה מזהה קבוצת אירוע (למשל ציורף subject+action+context subjet+action+context dla זכרן). **Detector**. את התוצאות הוא שומר במAGER או מעביר יישורות ל-

- Clustering & Pairing (מקבץ ומארת זוגות):** זה החלק מה-**Detector** או שלב נפרד. הרכיב לוקח את כל השוקים המנורמלים ומבצע קיבוץ. אפשר למכש בשני שלבים:

- Clustering לאשכולות אירוע:** על בסיס ה-**fingerprint** (או embedding), מחלק לקבוצות כך שבכל קבוצה שוקים שכנהה נוגעים לאירועו.

- Relation Detector :** בתרוך כל קבוצה, בודק אילו שוקים מקיים קשר לוגי. למשל, אם יש 3 תאריכים שונים, ידע שכל צמד עוקב (T1 vs T2, T2 vs T3, T1 vs T3) מהווים מועדים לארביטראז' קלנדרי. גם עשוי לבדוק סוג תלוות אחרים (אם היו, אך הפקוס שלנו קלנדרי). פלט של רכיב זה: רשימת **Pairs** (זוגות שוק) או **Groups** עם ציון סוג הקשר (כגון "same event different times" או "Market A implies Market B").

- Pricing Engine (מנוע תמחור/כדיות):** עברו כל זוג/קבוצה שמצאנו, מנוע זה מושך בזמן אמת את נתוני ספר הפקודות (Bids, Asks) העדכניים של אותו שוקים (כאן נכנס שימוש ב-WebSocket/WebSocket/עדכוני שוק). הוא מחשב האם יש פער מחייב מספיק (כמו בסעיף 5): לדוגמה מחשב על בסיס ה-Bid/Ask Best Bid Ask את הרוח הפונצייאלי 1%-תשואה, בהתחשב בנפקדים זמינים, בעמלות, בעליות גז וכו'. מנוע זה יכול להיות כל הזמן running, או מופעל באירוע:

- אפשרות 1: להריץ אותו באופן מתמשך לכל זוג שסוכן, כך שכל שינוי קטן בעומק השוק הוא מחשב מחדש פוטנציאלי.

- אפשרות 2: להפעיל אותו **On-Demand** כאשר מגע אירוע WebSocket. למשל, המonitor קיבל שינוי ב-market X, הוא יודע ש-X שירץ לוג (Y,X), אז מפעיל את calc Pricing על Y-X. מנוע התמחור מספק כתוצאה אינדיקציה:

האם יש ארביטראז' זמין מעל הסף, ואם כן, מה הפרטים (איזה כיוון – למכור באיזה ולקנות באיזה, כמה כמות, רוח צפוי). מידע זה הולך ל-**Detector/Decision Maker**.

- Arbitrage Detector / Decision Maker**: רכיב שמקבל את תוצאות Pricing. אם זהותה הزادנות העומדת בקריטריונים, הוא מחליט "לפעול". יתכן שהוא גם יבדוק שלא פעלנו כבר על הزادנות זו לפני רגע (מניעת כפוף), או שאין לנו כבר פוזיציה פתוחה (אולי לא עשו שוב אותה עסקה אם כבר מומשה). ברגע שהחלטת – הוא מעביר **Order Instructions** למודול-h-Executor.

- Execution Engine (מבצע פקודות)**: כפי שפורט בסעיף 6, זה הרכיב שמנהל את ביצוע העסקה בפועל מול זירות המסחר. הוא מקבל הוראה למשל: "Sell 300 YES on Market X @0.58, Buy 300 YES on Market Y @0.55. הוא מתמוך את שליחת הפקודות הנדרשות (דרך SDK או API מתאימים), מטפל ב-timeout ורכ. ה-Executor פועל באופן אסינכרוני (תוך כדי לשאר המרכיבים ממשיכה לנטר, אבל אולי ננהג בהירות ונמנע מלבצע 5 ארביטראז'ים בו-זמנית – נשלב תור ביצוע, או מספר אשכולות thread אם מערכת גדולה). בסיום, הוא מודיע על תוצאה:

- הצלחה: העסקה בוצעה במלואה, הפוזיציה כעת פתוחה (הכוונה: אנחנו מחזיקים כמות X בחזזה הארוכ-X ב-*No* ביחס הקצר, למשל).

- כשל: בוטל, לא מומש (או אין שינוי במצבנו) או מומש חלקית עם סגירה (אולי הפסד קטן). דיווח זה הולך ל-Risk-Manager/Position Tracker.

- מעקב פוזיציות ומנהל סיכון**: מודול שרושם כל פוזיציה שנכנסנו אליה. הוא ישמר טבלה של פוזיציות פתוחות: עבור כל ארביטראז' שביצענו – מה מצבנו (כמה חזים Long/Short בכל שוק, מה עלות הכניסה, מה הרוח הצפוי בפירעון).

- ה-Tracker משתלב עם המוניטור: הוא מנטר את מחיר השוקים של כל פוזיציה.
- ה-Risk Manager מגדר חוקים מתי להתרבע: למשל אם מחירו שינוי קיצוני במחיר אחד הצדדים, אולי שוק hedge נוסף (לא שכיח, כי כבר מגודר). יותר רלוונטי – אסטרטגיית יציאה: הוא יבודק אם ניתן לצאת כעת ברווח סביר. אם כן, הוא יאותת ל-**Execution Engine** לבצע **Exit Trade** (הופך מהכניסה: לkanות בחזרה/למכור).
- בנוסף, risk manager יפקח על חריגות: נניח שוק פתאום מסומן "Suspended" – הוא יתריגר התראה או טיפול מיוחד (אולי לסגור הצד השני אם אפשר).
- ה-Risk Manager יכול גם לרוץ סטטיסטיות: למשל לחשב כמה הון בסיכון (mbchinit collateral locked) יש לנו. אם הרבה, אולי נפחית פעילות.
- אם הפלטפורמה היא (Polymarket+Kalshi) cross-platform, הוא ודאי יעשה הצלבות לוודא שאין לנו חשיפה לאורקל שונה. במקורה שלנו (נניח התחלנו רק Polymarket ⁴⁰), פחות.

- חلك חשוב: Alerting/Logging** – מנהל הסיכונים יכול להתחבר למערכת התראות (Email/Slack) כדי לדוח לאדמינים על אירועים חשובים: "שוק XYZ סומן כ-pidvalחן", פוזיציה נסגרת ידנית! או "reconnected after 10s downtime" ועוד.

- Database / Storage**: בכל הארכיטקטורה, משתמש בסיס נתונים על מנת לשמר מידע:

- רשימת שווקים (סטטי או מותעדן) – כדי לא להריץ NLP כל פעם מאפס, אפשר לשמר תוצאות **Normalizer** ולרענן רק שינויים.
- תוצאות clustering – אולי נשמרו mapping של market_id – קבוצה/זוג כדי להשתמש בזמן אמת.
- עסקאות שביצענו – לтиיעוד L&P ולהימנע מכפילויות. גם למועד מיסוי/דיווח כמובן.
- אפשרות Position Tracker והיה חלק מה-DB: TABLE positions.
- לוגים לכל פעולה – לוגים debugging וניתוח (למשל ככמה ארביטראז'ים תפנסנו).

- DB יכול להיות פשוט SQLite או Postgres, תלוי כמה נתונים. יתרון Postgres: אפשר לשלב גם חיפוש טקסט (לשימוש NLP בסיסי) וקל לשימוש בריבוי תהליכיים.

User Interface / Monitoring Dashboard (לא חובה אבל טוב): אפשר לבנות דashboard המציג בזמן אמת את מצב המערכת – אילו זוגות שווקים זוהו, איפה יש פערים כרגע, כמה רוח פוטנציאלי, מה ביצעו וכו'. זה לא חוווני לolibה אבל מועיל לניטור אנושי. ניתן להשתמש בגרף גרפִי או אפליאן CL שמדפיס סטטוס.

ניצן לסקם את המודולים העיקריים בטבלה:

מודול	פקיד מרכזי	טכנולוגיות/הערות
Market Scanner	איסוף רישימת שוקים ונתוניהם (מטר-דאטा, תנויות, זמן)	DB / REST API של הפלטפורמה, SDK
& Normalizer NLP	נרטול טקסט, חילוץ אירוע/ישויות, יצירה מצזה אירוע (Fingerprint)	NLP (spaCy), custom code
& Cluster Relations	קבוץ שוקים לאירועים, זיהוי זוגות תלויים לוגית	מודל (SBERT) embedding, אלגוריתם אשכול (KMeans) או LLM agent
Order Book Monitor	ניתור בזמן אמיתי של Ask/Bid לכל שוק רלוונטי	OrderBook, WebSocket, מבנה נתוני
Pricing Engine	חישוב רוחה ארביטראז' פוטנציאלי על בסיס עמוק נוכחי	לוגיקת חישוב (Python), גישה לנוטוני OrderBook, חישוב عملות/גז
Opportunity Detector	החלטה על כניסה עסקה אם הרוחה מעלה סף	חווקים מוגדרים, בדיקות כפליות
Execution Engine	הוצאה לפועל של פקודות קנייה/מכירה simpultonio, ניהול מיילוי	ClobClient (למשל) SDK למסחר(asynco) למקביליות, ניהול שגיאות

מודול	תפקיד מרכז	טכנולוגיות/הערות
Position Tracker	שמירת פוזיציות פתוחות, חישוב L/P שוטף, מעקב סטטוס	שמירה ב-DB, חישובי (Python) P/L
Risk Manager	מדיניות יציאה (Exit), אינטגרציה להתראות (ביטולי שוקים, ניתוקים), התראות	חוקים (Rule engine), (Slack/Email)
Database	אחסון נתונים שוקים, פוזיציות, לוגים	PostgreSQL / SQLite
Alert/Logging	רישום אירועים וידוע מפעילים	ספריית לוגים (loguru וכו'), API לשילוח הודעות

מבנה זה מודולרי, מאפשר להחליף את מודול ה-NLP במודול אחר יותר בעתיד (נניח לעברו לשימוש ב-MLL), או להוסיף תומכה בפלטפורמה נוספת (פושט להרחיב את ה-Scanner ושאר השרשרת קצת, אבל העיקרון נשמר).

חשוב על **scalability** : עם ארכיטקטורה זו, ניתן להריץ רכיבים שונים במקביל (כפiosות microservices). למשל, Monitor, Detector, Executor יכולים לפעול כשרשור נפרד או אפילו תחילה. אפשר להשתמש message queue (כמו RabbitMQ) לתקשורת ביןיהם (למשל Monitor דוחף "opportunity" לqueue, ה-Executor מקבל). כך אפשר להתאים עומס: אם יש המון עדכנים, לשים backpressure.

בالمושך, בחרנו בטכנולוגיות (סעיף 10) שתומכו בארכיטקטורה אסינכרונית רבת-משתתפים זו היבט.

10. טכנולוגיות מומלצות למימוש Python, ספריות NLP, בסיסי נתונים, תורים, ניטור)

בבחירה טכנולוגיות, שקלנו אמינות, ביצועים וקלות שילוב. המטרה היא לבנות מערכת יציבה, בזמן-אמת, שניתן לפתח ולתזקק יחסית בקלות. להלן הבחירה העיקריות:

שפת תכנות: Python (אסינכרוני) – פיתון נבחר כיון שיש לו אקו-סיסטם עשיר גם לפיננסים (API web3, clients (clients) וגם ל-(spaCy, transformers) NLP, והפיתוח בו מהיר. ספציפית השתמש ביכילות **asyncio** של Python כדי לנהל מקבילויות (כגון האזנה במקביל ל-WS ושליחת פקודות בו-זמןית) בצורה ייעילה. פיתון אמן איתי בשטף בודד, אך צוואר הבקבוק פה הם O/I (תקשרות רשת) ולא חישובים כבדים, ולכן asyncio הוא בחירה טובה. אכן, קיים כבר ClobClient עבור Polymarket בפייתון²⁰, וספריות WebSocket נפוצות (socks aiohttp או websockets). בנוסף, השימוש בפייתון מקל על שילוב מודול NLP ומאפשר ניסויים/שינויים מהירים – חשוב בעולם שימושה.

גישה לפלטפורמה (API/SDK):

Polymarket: כדי שמודגם בספורט, מפעילה **CLOB** היברידית עם API. השתמש בספרייה Polymarket (כמו בקוד²⁰) כדי להתחבר לשירותי Polymarket. ספרייה זו תטפל בחתיות עסקאות Polygon, ניהול Keys וכו'. היא מספקת גם ממשק נוח ל-WS (כמו בקוד³¹). Kalshi או אחרות: אם נרצה להרחיב לפלטפורמות אחרות, אפשר להשתמש ב-REST שלחן (REST API יש REST, מוגבל קצב⁴⁴). ייתכן שנneedish תחילך נפרד לכך, אך ניתן.

Web3 (Blockchain): Polymarket משתמש כנראה ב-`pyweb3` כדי לתחבר עםigon⁴⁷. השתמש בו עבור פעולות כמו `.fetch gas price`.

ספריות NLP:

נשתמש ב-**spaCy** (עם מודול אנגלי מתקדם) לחילוץ ישותיות (PERSON, ORG, GPE, DATE, CARDINAL) ומיון (PERSON, ORG, GPE, DATE, CARDINAL). spaCy נוחה ומדויקת יחסית, ונitinן להוסיף לה כללים. למשל, אפשר לאמן קצת זיהוי סוג שאלות ספציפיות (עם או בלי, אך נראה מספיק שימוש במונע הקדים). עבור נרמול תאריכים נשתמש ב-**dateparser** שיכול לקחת מהורת ("by end of 2025") ולהציגה כתאריך סטנדרטי (31/12/2025). אם נבצע embedding לטקסט, נוכל להשתמש בספריית **sentence-transformers** (המספקת מודלים pre-trained כמו 2v-L6-MiniLM-all, שהוא מהיר ונוטן וקטורים 384-מימד) לחישוב וקטור לכל שאלה. זו ספרייה פיתונית קלה לשימוש. ניתן שני עזר גם ב-**NLTK** או **regex** פשוטים עבור כמה דפוסים (למשל חיפוש "(4{4}by" כדי לתפוס שנה). בפייה וナル לכוון agent LLM, אפשר לשלב למשל API של OpenAI GPT-4 למטלות clustering relational clustering/relation).abel זה נראה overkill ולא הכרחי בהתחשב שאפשר פתרון זויל יותר. מעבר לכך, LLMs כבדים יכולים להוסיף תלות חיונית ופחות אמיןנות.

בסיס נתונים (Database):

נבחר ב-**PostgreSQL** כמגד נתונים רלוונטי עיקרי. הוא אמין, ידוע ומתפל היטב בטרנזקציות. ניצור טבלאות עבור **Logs**, **Relationships**, **Positions**, **Trades**, **Markets** וכו'. אם נרצה מהירות בפעולות קרייה רבות (כמו בדיקות מדירות על מחירים), אפשר להשתמש ב-Redis cache. אבל נראה לא נכון כי המידע החי בא מה-SQS בכל מקרה. שימוש ב-Postgres מקל גם שילוב למערכות ניהול (ניתן לצרף Grafana או tools אחרים). **asynchronous DB access:** נשתמש ב-**asyncio** driver כמו `asyncpg` כדי לא לחסוםoop כאשר כתבים ללוג/DB.

מנגנון תורים (Queue):

להפרדת מודולים, ניתן שימוש ב-**RabbitMQ** או **Redis Streams**. למשל, Monitor, Executor. Redis Streams מכניס אובייקט "Opportunity" לתור, וה-Executor צריך אותו. זה מוסיף איתנות – אם ה-Executor איטי, התור נערם; אם קורס, ההודעה נשמרת أول. אפשר גם בפשטות, בתוך סיס Python asyncio Queue, להשתמש בעבור תקשורת בין קורוטינות. אם אנו בפלטפורמה אחת ובוט אחד, queue internal עשוי להספיק. אמיןות: עבור מערכת mission-critical, RabbitMQ (עם persistent delivery) מבטיח שלא נפספס הזדמנויות גם אם התחלה restart. זה שיקול כבד יותר. לאור מורכבות, אפשר להתחיל ללא, ואם רואים צורך, להוסיף RabbitMQ.

ינטור ותצוגה:

Logging: נשתמש בספריית logging של Python (או משודרגת כמו `loguru`) כדי לרשום קבצים עם כל אירוע חשוב. **Metrics:** ניתן לשלב חישפה של metrics (Prometheus – למשל קצב עסקאות, רוח יומי, latency וכו'). **Alerting:** לחיבור מהיר, אפשר להגדיר את Risk Manager שישלח התראה במייל או Slack כאשר אירוע מסוים קורה (נניח Loss חריג, או הזדמנויות >20%). הספרות מראה דוגמה של AlertSystem שנותן תיעוד וDIRIGOG דחיפות . 45

Dashboard: אפשר ליצור ממשק web בסיסי (אולי Flask + some charts) כדי לצפות במצב. לא קרייטי אבל מועיל. טכנולוגיה אפשר גם Streamlit (מאפשר בניית GUI מהירה בפייתון) – אם רוצים לראות רשימת הזדמנויות בזמן אמת.

:Scaling & Performance

אפשר לטפל ב-100+ שוקים במקביל (כל עוד רוב הזמן מבלים בהמתנה ל-SQS).

אם נרحب לפלטפורמות רבות או לאלפי שוקים, ייתכן צורך בריביו תהליכיים. אפשר להשתמש ב- **multiprocessing** או לשבץ מודולים שונים כמיקרו-שירותים (למשל Service אחד עשוה NLP clustering לשירותים (למשל Service אחד עשוה overkill - מספר השוקים אחת לתקופה, Service אחר מקבל updates ומחשב).

גם אפשר להיעזר ב-GPU ל-NLP אם עושים embedding לכמויות, אך זה נראה overkill - מספר השוקים לרוב לא עצום (~ מאות עד אלפיים) ונitinן לבצע embedding חד פעמי לכל שוק ולהתעדכן עבור חדשים.

אמינותה (Reliability):

נريיך את הבוט בסביבת שרת יציבה (למשל VM Linux).
להשתמש בכלים ניהול תהליכיים (systemd / docker container with restart policy) שיבתיחו חזרה לעבודה אם נזק.

יחידות בדיקה (Unit tests) אפשר לכתוב עבור לוגיקת החישוב (לבדוק שפונקציית profit מחשבת נכון וכד').
Polygon/Blockchain concerns: לוודא שימוש ב-API מהיר (ייתכן להחזיק Node עצמי או להשתמש ב-endpoint מהיר כדי שהפקודות ישודרו מהר).

ה-*ClobClient* ככל הנראה פותר הרבה, אבל אולי נשתמש גם ב-*web3* לעניין הגז וכו' .⁶

דוגמאות טכנולוגיות אלטרנטטיביות:

אפשר היה לשחק Node.js/TypeScript עם ספריות SW-*web3*, אבל אז NLP נהיה פחות זמין (יש, אך Python מוביל בהזה).
C++/Rust – מהיר אך פיתוח איטי, לא צריך לרובד זה.

בסיום, *async* *await* עונה על הדרישות באופן מדהים. וכן, פרויקטים בפועל דיווחו על שימוש בו – למשל *Polymarket Agents toolkit* הוא בפייתון⁴⁸, וגם בוטים (*Navnoor Bawa* המזכיר הרץ נראה בפייתון, לפחות הקוד שהציג).

ספריות/טכנולוגיות נוספות:

websockets (python) – לטיפול בקשר WebSocket.
pandas – אולי לניתוח נתונים/לוגים אם נאסו, אך לא נדרש זמן אמת.
dataclasses – להגדרת מבני נתונים (למשל מחלוקת *ArbitrageOpportunity* עם שדות) כדי שייהיה קל לנויהם.
asyncio tasks – ניהול משימות מתזמנות (למשל scanner כל שעה).
Sentry – כל שירוטי למעקב אחר שגיאות runtime.

הטכנולוגיות האלה יוצרות יחד מערכת אינטגרטיבית: - (io Reactor Async pattern (asyncio לניהור וביצוע בו-זמןית). - libs-1 NLP Data Science Integration Queue-1 Queue לאמינות וארגון. - API עם Polymarket Logging/Monitoring לשקיפות. (שכבר בניה). - Polymarket for trading agents scaffolding (למשל מספק "Calendar Arbitrage".

כל הרכיבים הללו ניתנים לחיבור בפייתון, ויש ניסיון מוצלח בקיליה: *Polymarket Agents* לשחק מספק "Calendar Arbitrage".

11. ניתוחים ומרקבי בוחן 2024-2025 (בחירות, גיאופוליטיקה, מוצריו טכנולוגיה, פלטפורמות נוספות)

נסקרו כעת מספר מקרים אכדיים מן השנים האחרונות המדגימים את השימוש בארビטראז' קלנדרי ולגו, ואת הביצועים של מערכות דוגמת זו שתוארה. חלק מהנתונים מגעים מחקרים שהקרו פעילות ארביטראז'ים ב-PolyMarket בתקופה זו:

א. בחירות 2024 בארה"ב – ארביטראז' בין שוקים תלויי-תוצאה ותויח זמן:

הבחירה לנשיאות בארה"ב 2024 ייצור שפע שוקים על אירופים: הן שוקים ישירים ("מי ניצח את הנשיאות?") והן שוקים נלווים ("מי ניצח בפופולר ווט?", "מה היה הרכב הסנאט?"), וכן שוקים עם טויח זמן שונים (למשל חלום נסגר ב-2024, חלום בינויו בא-ריסטו). מחקר מצא **13 זוגות שוקים תלויים** הקשורים לבחירות בנובמבר 2024 . אחד המקרים הבולטים (שכונה "Pair 4") היה: - שוק A: "המנצח בפופולר ווט היה גם נשיא?" (במיללים אחרים, האם לא יהיה פער בין הצעת העם לקלות האלקטוריום). - שוק B: "מי ניצח בפופולר ווט?" (מוסכם דמוקרטי או רפובליקני).

קשר לוגי: אם שוק A טען שלא יהיה פער (Popular Vote winner = President), ניתן לגוזר הסתברות משוק B (מי הפופולר ווט) וכי הנשיאות. למעשה, זוג שוקים אלו אפשרו לארביטראז' להרכיב פוזיציות שחוקות להימור על "יהיה פער". נמצא שבמהלך הקמפיין היו **אלפי רגעים** בהם השוקים האלו לא היו מתחאים, ונitin היה להרוויח על ידי קנייה/מכירת שילוב כותאים . 4 Pair הנ"ל הציג לא פחות מ-6,630 מדדים . אמנים רובן היו קטנות (פער מחיר של סנטים בודדים) בשל פעילות גבוהה. אך סוחר אוטומטי זריז היה יכול לנצל רבות ממנה. המחקר ציין שהרווח המקסימלי המוצע בארביטראז' בין-שוקים אלה היה סביבה \$100 בלבד ⁵⁰ (מוגבל בervalות של ~\$2,000 בעותם וגעים), לעומת זאת להרוויח ממשמעותית צריך היה לבצע שוב ושוב (מה שากע נעשה: סוחר מוביל ביצע 4,049 עסקאות ארביטראז' בתקופה והרוויח מעל \$2 מיליון!).²

דוגמה מספרית: נניח לפניה הבחירות, שוק A (אין פער) היה ב-0.8% (80% סיכוי שלא יהיה פער), ושוק B הערך 60% סיכוי שהדמוקרט ניצח בפופולר ווט. שילוב הגוני היה צריך לקיים יחסים מסוימים (מורכב מעט לחישוב ידני, אבל אם הדמוקרט מוביל בפופולר והפופולר=נשיאות סביר, אז הדמוקרט גם לנשיאות ~0.8*-0.6=0.48%). אם מצאו אי התאמה – קנו/מכרו.

ב. שוקים קלנדריים על השבעתו של נשיא, 2024-2025:

יראה Polymarket אירחה שוקים כגון "האם ג'ו בידן יהיה נשיא ב-31/12/2024??" לצד "האם ג'ו בידן יהיה נשיא ב-01/06/2025??" (מוסדים שלローンטים לביראותו, אולי). קשר לוגי: אם הוא מכון בסוף 2024, סביר שהיה גם באמצע 2025, אלא אם קורה שהוא ברבעון הראשון. אם השוק הקצר (2024) נסחרגובה מהארוך (2025), היה ניתן לעשות Calendar Arbitrage. ואכן, ניתן רגעים בהם חשש קצר-טויח (למשל שמועה על פרישה) העלה את ההסתברות שלא ישרוד עד 2025 יותר מהסתברות שלא ישרוד עד סוף 2024, באופן לא הגיוני. ארביטראז' זהה היה הכרך במכירת חזים קצרים וקניית ארוכים ולהיפך לפני ההצעה. סוחרים שעקבו אחר מצב בריאותם של מנהיגים השתמשו בה. (אין לנו מספרים מדויקים פומביים, אך זה תר楸יש סביר שקרה).

ג. אירועים גיאופוליטיים – דוגמת המלחמה באוקראינה:

בתחילת 2022 היו שוקים: "האם רוסיה תפלוש לאוקראינה לפני 1 במרץ 2022?" ואלו "האם רוסיה תפלוש לפני אפריל 2022?" (תאריך מאוחר יותר). כשתוחוי הזמן הופפים (מרץ ואפריל), ברור שאם הסיכוי עד מרץ היה בטוח גבו מה עד אפריל, יש ארביטראז'. בתחילת פברואר 2022, למשל, השוק קצר-הטויח אויל היה בתנודות גבוהה (חדשנות יומיות), וייתכן נסחר בפרמייה מול הארוך. סוחרים מקצועיים ניצלו עיוותים כאלה – מוכרים את הקצר וקונים את הארוך. אם הפלישה לא קורתה עד מרץ, הקצר פג ב-S&P (המ זכו שם), אך הארוך עוד אפשרות לפליישה עד אפריל (אולי בסוף קרה בפברואר אז שניהם כן – אך אז הפסידו בקצר, זכו באורך, נראה עידיין רוחחים אם יחש היה נכון). מחקר לא ציין ספציפיות את המקרה הזה, אך כן הזכיר שמאורעות תנודתיים (כמו מלחמה) גורמו **לא-יעילות** זמנית במחירים .² סוחרים הפיקו רווחים משמעותיים בסמוך לאירועים כמו פלישות או הסכמי שלום, כי השוק הכללי הגיב לאו בצורה לא אחידה.

ד. השקת מוצרי טכנולוגיה (Apple example)

הריצה שוק PolyMarket "האם Apple תשיק קו מוצר חדש עד סוף 2025?" (שהסתומים ב-No), וגם שוקiaroo טוווח יותר: "האם Apple תשיק קו מוצר חדש לפני 2027?" – נכון לעכשיו נסחר בסבירות 83% ⁵³. בתקופת 2025, אם השוק קטן קצת הטווח (עד 2025) היה מחירו גבוה מהשוק הנוכחי (עד 2027), זו הייתה הזדמנות ברורה. בפועל, סביר שהשוק הנוכחי תמיד היה מעיל (כי עד 2027 כמעט ודאי Apple תעשה משהו חדש). ואכן הגיעו: לפני 2027 עמדו ~83% לוננייה לפני 2025 היה 10% וה-2027, 70%, לאربיטראז'ר יש "כספי על הרצפה": מוצר 2025 @10% Yes (או קונה @90% No), וקונה 2027 @70% Yes. עלות: $1.60 = 0.90 + 0.70$. על שתי ייחידות? זה יותר מ-1, רגע – כאן יש יותר מרכיבות כי צריך לנראה לשלב עם עוד מוצר (אולי עוד שוק ב-2026). אבל אם היה 10% מול 70%, כנראה השוק תיעזן מהה. סצנה סבירה יותר: באמצע 2024, שमועות על מוצר AR/VR (שבסוף הוכחה 2023). אולי "עד 2025" נסחר 40%, עד 2027" 60%. אם מישחו חשב ש- אם לא עד 2025 אז כן עד 2027 (די הגיוני), אולי פרופיל מוצרים היה אחרת. בכל מקרה, ארביטראז'ר יוכל היה לעקב. דוגמה נוספת: שוק על "Will SpaceX reach Mars orbit by 2024?" "לעומת" by 2025. פערים יכולים לפעמים מתרחשים כי קהל הסוחרים בכל שוק שונה (אופטימיסטים קצרי טוווח מול ריאלייטים ארוכי טווח).

ה. ארביטראז' בין פלטפורמות (Kalshi vs Polymarket)

לא בדיק Polymarket, אבל קשור ל- "Logical Arbitrage". לעיתים Polymarket ו-Kalshi הציעו את אותו שוק (למשל תוכזאה של בחירות, או מודד כלכלי). למשל בספט' 2025, הבחירה בארה"ב גרמו למחירי שוקים של Polymarket vs Kalshi להיות שונים. מחרקים צינו שפיצול נזילות בין פלטפורמות הביא להפרשי מחירים ניכרים .⁵⁴

זה נוצר ע"י בוטים של "Cross-market arbitrage". לדוגמה, שוק על "Republicans win House in 2024" היה 55% ב-1-60% ב-Polymarket – סוחר היה קונה ב-55% ב-Kalshi, מוכר ב-60% בפולימרקט, ונousel עיר (ברגע האמת שתיהן משלמות \$1 אמר נכוון, אז רוח 5 סנט מינוס עמלות). ארכיטור איז' צהה דורש גם בוט ומתודולוגיה דומה (זיהוי שוקים מקבילים בניסוח שונה – NLP יכול לעזור כאן מאוד גם, כי הפלטפורמות מנחות אחרת). גם דורש התיחסנות בעמלות שונות Kalshi גובה fee על רוח). המערכת שלנו ניתנת להרחבה להה: ה-eScanner/Normalizer היו קוראים משתפי פלטפורמות, מקבצים cross-platform. מנוע התמוך היה בודק גם שם. בפועל, ב-2024-2025 היו כבר כלים (כמו שהמשתמש Reddit ציין⁵⁵) ששולחים קלשי ופולימרקט ומציגים הדמנויות.

1. תוצאות בפועל:

מקורות שונים מערכים שסוחרי ארביטראז' גרכו **עשרות מיליון דולר** בתקופה זו מריבוי הארביטראז'ים.⁵⁶ זה לא מספר מבוטל, ומדגיש עד כמה השוק היה לא-יעיל בזמן סוערים. עוד צוין ש-'עוקר ארביטראז'" ("bot-like bettors") הפכו את Polymarket לסוג של "מכונת ארביטראז' ולא קזינו" לטענות כתבה⁵⁶ – דהיינו, במקום שאנשים יהמרו באשראי, הרבה פעילות נהייתה וניצול שיטתי של פעירים.

סיכום מカリ הובחן: המ מראים שהמערכת שהגדנו אינה תאורטית בלבד. שימוש של NLP לאיתור שוקונים תלויים, וחברות Order Books עם בוט ביצוע מהיר, הוא בדיקת מה שסוחרים מוצלחים עושים בתקופה الأخيرة: - בבחירה, זיהוי תלות פופולר וטונשיות והפעלת בוט שעם כל תנודת סקר ישר מנצל פער בין השוקנים⁴⁹. - בזירות מרבות, השוואת מחיר וניצול שהוא או הבדלי משתמשים⁵⁴. - בזמן אירועים דרמטיים, היכולת להגיב תוך שניות במקביל (להכנס פקודות לפני שהפער נסגר ינית).

הארכיטקטורה המודולרית, הטכנולוגיות NLP (Python Async, וכו'), והבנת הסיכוןים – כל אלה היו ככל הנראה את בסיס מערכות ה-HFT (מסחר בתדריות גבוהה) שפעלו בפועל מוקטן. כפי שמצין אחד המקורות, "טובי הארביטראז'רים ביצעו אף עסകות בוטים בתקופת בחירות 2024"², וניצלו את זמני התנודתיות (Debate nights, polling, polling) – זה בדיקת התרחיש שלילית: הבוט מאזין (אולי אף קיבל חדשות בזמן אמיתי) ומכתה בשוק שבו מחירים "לא הסתדרו" לפני שההחל הרחב מעדרן.

עם התבוסות השוק והऋשות גופים מקצועיים (למשל השקעת ICE בכלכלי, ויפוי כניסה שחקנים גדולים ב-2025),
הפערים יצטמצמו.⁵⁷ אך כאמור, נכון ל-25/2024, זו הייתה שעתם היפה של ארכיטראז'רים זריים.

סיכום: ארביטראז' קלנדי ולוגי בשוקי תחזיות הוא תחום מרתך שבו הנחות מתמטיות ולוגיות מותרגמות לאסטרטגיות מחר אלגוריתמיות. באמצעות הרוגה מעמידה של יחסיות בין אירועים, רונייה הוכחות לכוון האրיטראז'. ומתכוון הודמי

מקודם של מערכת אוטונומית – ניתן היה (ועדיין ניתן) לקצור ורוחים כמעט חסרי-סיכון משוקי התחזיות הצומחים. תכננו מסמך אופרטיבי שלם: מהיסודות התאורטיים, דרך ההיבטים המעשיים (عملות, נזילות), ועד לארכיטקטורת תוכנה מבוזרת ומודולרית, שנתמכת בטכנולוגיות עדכניות. דוגמאות מהשיטה הדגימו שהתייאורה פגשה מציאות – והמערכות הללו אכן עובדות. *Implementing* לארביטראז' קלינדרי הואאתגר אך בעל תגמול גבוה, וכשהוא מבוצע נכון, הוא שיטתי, סקלְאַבִּילִי, ומציג כיצד מידע וזמן הופכים לכף בעולם שוקי התחזיות .⁵⁶

ואולי הדבר המרגש ביותר: בעוד משקיעים רגילים "מהמרים" על תוצאות, המערכת שלנו מימושה תפיסה הנדסית – להרוויח ללא קשר לתוצאה, רק בזכות חוסר השלים של שוקי המונחים. זהו הכוח של ארביטראז' לוגי, וכפי שהראינו, ניתן לרתום אותו בהצלחה בשוקי התחזיות של השנים האחרונות.

:Sources

.1 Substack , "Building a Prediction Market Arbitrage Bot: Technical Implementation" ,Navnoor Bawa
31 29 20 2 1 Nov 2025

.2 arXiv 2025 , "Unravelling the Probabilistic Forest: Arbitrage in Prediction Markets" ,Oriol Saguillo et al
50 49 33

.3 Oct ,Well, CAN THEY PREDICT?": Crypto Prediction Contracts" ,Bocconi Students Investment Club
4 3 2023

.4 46 45 43 40 42 8 5 28 27 26 25 2025 ,Arbitrage Bot Code Snippets ,Navnoor Bawa
19 55 Nov 2025 ,Polymarket x Kalshi Arbitrage Finder ,Probutkickerz Reddit user

.5 21 13 arXiv 2025 ,Semantic Trading: Agentic AI for Clustering and Relationship Discovery
35 34 Appendix Unravelling the Probabilistic Forest ,Arxiv

.6 53 51 2025 ,Apple product line 2025 ,Polymarket Market Page
57 54 47 46 45

44 43 42 41 40 39 38 37 36 32 31 30 29 28 27 26 25 24 23 20 11 9 8 7 6 5 2 1
Building a Prediction Market Arbitrage Bot: Technical Implementation
<https://navnoorbawa.substack.com/p/building-a-prediction-market-arbitrage>

Well, CAN THEY PREDICT?": An investigation of Crypto Prediction Contracts' – BSIC | Bocconi" 4 3
Students Investment Club
<https://bsic.it/well-can-they-predict-an-investigation-of-crypto-prediction-contracts>

Unravelling the Probabilistic Forest: Arbitrage in Prediction Markets 50 49 35 34 33 10
<https://arxiv.org/html/2508.03474v1>

Semantic Trading: Agentic AI for Clustering and Relationship 48 21 18 17 16 15 14 13 12
Discovery in Prediction Markets
<https://arxiv.org/html/2512.02436v1>

Polymarket x Kalshi Arbitrage Finder : r/Kalshi 55 22 19
https://www.reddit.com/r/Kalshi/comments/1parkbp/polymarket_x_kalshi_arbitrage_finder

Will Apple release a new product line in 2025? Prediction... | Polymarket 53 52 51
<https://polymarket.com/event/will-apple-release-a-new-product-line-in-2025>

People making silent profits through arbitrage on Polymarket ⁵⁶

<https://www.chaincatcher.com/en/article/2212288>