

2 פרוטוקול הליגה הכללי

2.1 עקרונות הפרוטוקול

הפרוטוקול מגדיר כלליים אחידים. הכללים מאפשרים לסוכנים שונים לתקשר. כל סטודנט יכול למשוך סוכן בשפה שירצה. כל עוד הסוכן מכבד את הפרוטוקול – הוא ישתתף בליגה.

2.1.1 הפרדה לשלווש שכבות

המערכת מורכבת משלווש שכבות:

1. **שכבת הליגה** – ניהול טורניר, רישום שחknim, טבלת דירוג.
2. **שכבת השיפוט** – ניהול משחק בודד, אימותות מהלכים, הכרזת מנצח.
3. **שכבת חוקי המשחק** – לוגיקת משחק ספציפית (זוגי/אי-זוגי, איקס-עיגול, ועוד').

ההפרדה חשובה. היא מאפשרת להחליף את שכבת המשחק. הפרוטוקול הכללי נשאר קבוע.

2.2 סוגי הסוכנים

2.2.1 מנהל ליגה – League Manager

מנהל הליגה הוא סוכן יחיד. הוא אחראי על:

- רישום שחknim לליגה.
- יצירת לוח משחקים (Round-Robin).
- קבלת תוצאות מהשופטים.
- חישוב ופרסום טבלת דירוג.

מנהל הליגה פועל כשרת MCP על פורט 8000.

2.2.2 שופט – Referee

השופט מנהל משחק בודד. **חשוב:** לפני שהשופט יכול לשפט משחקים, הוא חייב להירשם אצל מנהל הליגה.

השופט אחראי על:

- רישום למנהל הליגה – לפני תחילת הליגה.
- הזמנת שני שחknim למשחק.
- ניהול תורות המשחק.
- אימותות חוקיות מಹלכים.
- הכרזה על תוצאה ודיווח לליגה.

השופט פועל כשרת MCP על פורט 8001. יכולם להיות מספר שופטים במערכת (פורטים

2.2.3 סוכן שחקן – Player Agent

סוכן השחקן מייצג שחקן בליגה. הוא אחראי על:

- הרשמה לליגה.
- קבלת הזמנות למשחקים.
- בחירת מהלכים במשחק.
- עדכון מצב פנימי לפי תוצאות.

כל שחקן פועל על פורט נפרד (8104-8101).

2.3 מזהים בפרוטוקול

כל רכיב במערכת מזוהה באופן ייחודי.

טבלה 1 : מזהים בפרוטוקול הליגה

תיאור	סוג	שם המזהה
מזאה ליגה ייחודי	String	league_id
מספר מחזור בליגה	Integer	round_id
מזאה משחק בודד	String	match_id
סוג המשחק	String	game_type
מזאה שחקן	String	player_id
מזאה שופט	String	referee_id
מזאה שיחה	String	conversation_id

2.3.1 דוגמאות למזהים

```
"league_2025_even_odd" :league_id -  
... ,3 ,2 ,1 :round_id -  
"R1M1" :(מחזור 1, משחק 1) :match_id -  
"tic_tac_toe" או "even_odd" :game_type -  
"P20" ,... , "P02" , "P01" :player_id -  
... , "REF02" , "REF01" :referee_id -
```

2.4 מבנה הודעה כללי – Envelope

כל הודעה בפרוטוקול חייבת לכלול "מעטפת" (Envelope) עם שדות קבועים. הממעטפת מבטיחה עקביות ומאפשרת מעקב אחר ההודעות.

מבנה מעטפת הודעה – Envelope

```
{  
    "protocol": "league.v2",  
    "message_type": "GAME_INVITATION",  
    "sender": "referee:REF01",  
    "timestamp": "2025-01-15T10:30:00Z",  
    "conversation_id": "conv-r1m1-001",  
    "auth_token": "tok_abc123def456...",  
    "league_id": "league_2025_even_odd",  
    "round_id": 1,  
    "match_id": "R1M1"  
}
```

שדות חובה במעטפת 2.4.1

טבלה 2: שדות חובה בכל הודעה

תיאור	סוג	שדה
גרסת הפרוטוקול, קבוע "league.v2"	String	protocol
סוג ההודעה (למשל GAME_INVITATION)	String	message_type
מזהה השולח בפורמט type:id	String	sender
חוותמת זמן ISO-8601 באזור זמן UTC	String	timestamp
מזהה ייחודי	String	conversation_id

2.4.2 דרישת אזור זמן – UTC/GMT

חוובה: כל חותמות הזמן בפרוטוקול חייבות להיות באזור זמן UTC/GMT. דרישת זו מבטיחה עקביות בין סוכנים הפעילים ממקומיים גיאוגרפיים שונים.

טבלה 3: פורמטים תקינים ולא תקינים לחותמת זמן

הסבר	תקין?	פורמט
סימנת Z מציין UTC	✓	2025-01-15T10:30:00Z
הפרש +00:00 UTC- לשווה ערך	✓	2025-01-15T10:30:00+00:00
אזור זמן מקומי – אסור	✗	2025-01-15T10:30:00+02:00
לא אזור זמן – אסור	✗	2025-01-15T10:30:00

הערה חשובה: סוכן ששולח הודעה עם אזור זמן שאינו UTC יקבל שגיאה E021 (INVALID_TIMESTAMP).

2.4.3 שדות אופציונליים

טבלה 4: שדות אופציונליים לפי הקשר

תיאור	סוג	שדה
אימות טוקן לאחר רישום (חוובת רישום)	String	auth_token
מזהה הליגה	String	league_id
מספר מחזור	Integer	round_id
מזהה משחק	String	match_id

2.4.4 פורמט שדה sender

שדה sender מזהה את שולח הודעה:

- מנהל הליגה – league_manager
- שופט עם מזהה .REF01 – referee:REF01
- שחקן עם מזהה P01 – player:P01

2.4.5 טוקן אימות – auth_token

לאחר רישום מוצלח, כל סוכן מקבל auth_token. הטוקן חייב להופיע בכל הודעה שנשלחת לאחר הרישום. זה מונע התחזות לsoccerists אחרים.

קבלת טוקן בתגובה רישום

```
{  
    "message_type": "LEAGUE_REGISTER_RESPONSE",  
    "status": "ACCEPTED",  
    "player_id": "P01",  
    "auth_token": "tok_p01_abc123def456ghi789..."  
}
```

2.5 זרימת הליגה הכללית

2.5.1 שלב 1: רישום שופטים

בשלב הראשון, כל שופט נרשם לליגה. השופט שולח בקשה רישום למנהל הליגה. מנהל הליגה מקצת referee_id ושומר את כתובות השופט.



2.5.2 שלב 2: רישום שחקנים

לאחר רישום השופטים, כל שחקן נרשם לליגה. השחקן שולח בקשה רישום למנהל הליגה. מנהל הליגה מקצת player_id ומאשר.



2.5.3 שלב 3: יצרת לוח משחקים

לאחר שכל השחקנים נרשמו, מנהל הליגה יוצר לוח משחקים. הלוח מבוסס על שיטת Round-Robin. כל שחקן משחק נגד כל שחקן אחר.

2.5.4 שלב 4: הכרזה על מחזור

לפני כל מחזור, מנהל הליגה מפרסם הודעה ROUND_ANNOUNCEMENT. ההודעה מפרטת את כל המשחקים במהלך. מנהל הליגה מקצת שופט לכל משחק מתוך רשימה של שופטים הרשומים.

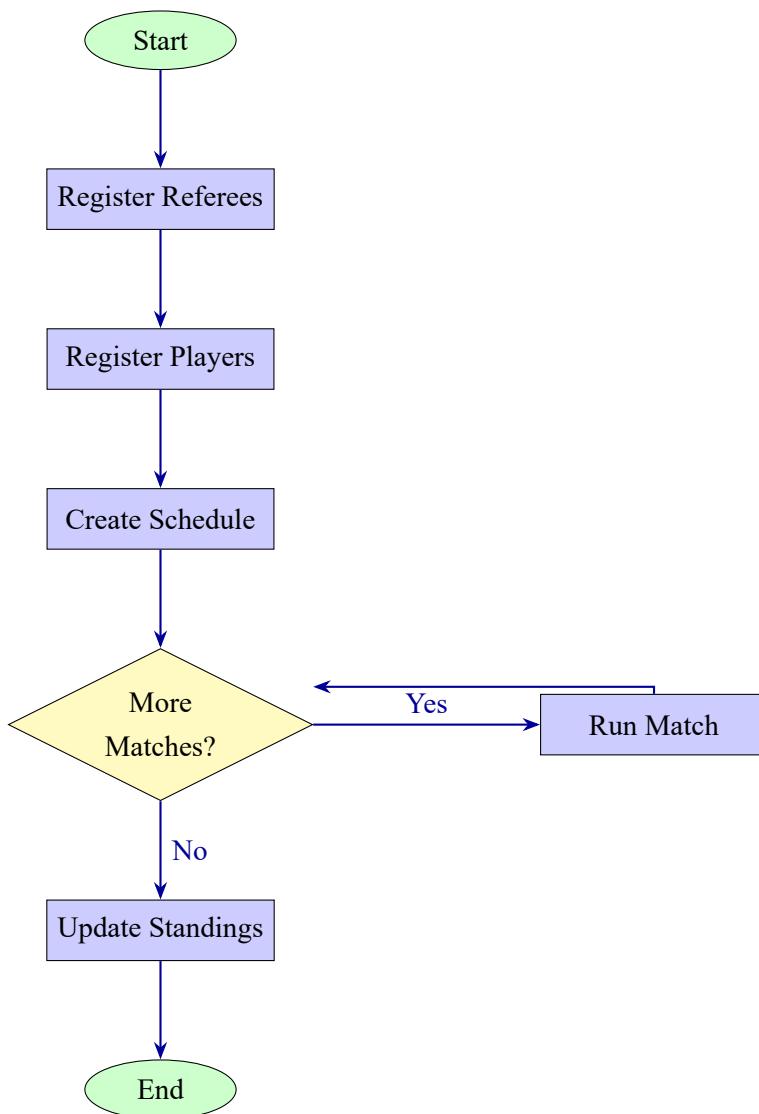
2.5.5 שלב 5: ניהול משחקים

השופט מזמין שחקנים למשחק. הוא מנהל את המשחק לפי חוקי המשחק. בסיום, הוא מדווח תוצאה למנהל הליגה.

2.5.6 שלב 6: עדכון דירוג

לאחר כל מחזור, מנהל הליגה מעדכן את טבלת הדירוג. הוא מפרסם את הטבלה לכל השחקנים.

2.6 דיאגרמת זרימה כללית



2.7 זמני תגובה - Timeouts

לכל סוג הودעה מוגדר זמן תגובה מקסימלי. אם הסוכן לא עונה בזמן, הפעולה נחשבת ככשלון.

טבלה 5: זמני תגובה לפי סוג הودעה

הערות	Timeout	סוג הודעה
רישום שופט לliga	10 sec	REFEREE_REGISTER
רישום שחקן לliga	10 sec	LEAGUE_REGISTER
אישור הגעה למשחק	5 sec	GAME_JOIN_ACK
בחירה זוגי/אי-זוגי	30 sec	CHOOSE_PARITY
קבלת תוצאה משחק	5 sec	GAME_OVER
דיווח תוצאה לliga	10 sec	MATCH_RESULT_REPORT
שאילתת מיידע	10 sec	LEAGUE_QUERY
ברירת מחדל	10 sec	כל השאר

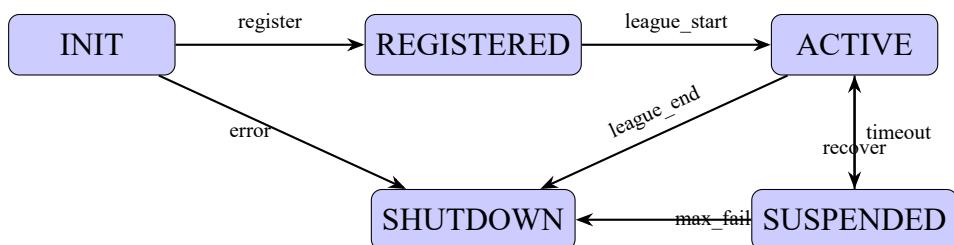
2.8 מחזור חיים של סוכן – Agent Lifecycle

כל סוכן (שחקן, שופט) עובר מצבי מוגדרים במהלך הליגה.

2.8.1 מצבים הסוכן

- INIT – הסוכן הופעל אך טרם נרשם.
- .auth_token - REGISTERED – הסוכן רשם בהצלחה וקיבל auth_token.
- ACTIVE – הסוכן פעיל ומשתתף במשחקים.
- SUSPENDED – הסוכןמושעה זמןית (לא מגיב).
- SHUTDOWN – הסוכן סיים פעילות.

2.8.2 דיאגרמת מעברי מצבים



2.9 טיפול בשגיאות

הפרוטוקול מגדיר שני סוגי הודעות שגיאות:

2.9.1 שגיאת ליגה – LEAGUE_ERROR

מנהל הליגה שולח הודעה זו כאשר מתרחשת שגיאה בرمת הליגה.

דוגמת שגיאת ליגה

```
{  
    "protocol": "league.v2",  
    "message_type": "LEAGUE_ERROR",  
    "sender": "league_manager",  
    "timestamp": "2025-01-15T10:35:00Z",  
    "error_code": "E005",  
    "error_name": "PLAYER_NOT_REGISTERED",  
    "error_description": "Player\u00d7ID\u00d7not\u00d7found\u00d7in\u00d7registry",  
    "context": {  
        "player_id": "P99"  
    },  
    "retryable": false  
}
```

2.9.2 שגיאת משחק – GAME_ERROR

השופט שולח הודעה זו כאשר מתרחשת שגיאה במהלך משחק.

דוגמת שגיאת משחק

```
{  
    "protocol": "league.v2",  
    "message_type": "GAME_ERROR",  
    "sender": "referee:REF01",  
    "timestamp": "2025-01-15T10:31:00Z",  
    "match_id": "R1M1",  
    "player_id": "P01",  
    "error_code": "E001",  
    "error_name": "TIMEOUT_ERROR",  
    "error_description": "Response\u00d7not\u00d7received\u00d7within\u00d730\u00d7seconds",  
    "game_state": "COLLECTING_CHOICES",  
    "retryable": true,  
    "retry_count": 1,  
    "max_retries": 3  
}
```

2.9.3 קודי שגיאה נפוצים

טבלה 6: קודי שגיאה עיקריים

תיאור	שם	קוד
תגובה לא התקבלה בזמן	TIMEOUT_ERROR	E001
שדה חובה חסר	MISSING_REQUIRED_FIELD	E003
בחירה לא חוקית	INVALID_PRIORITY_CHOICE	E004
שחקן לא רשום	PLAYER_NOT_REGISTERED	E005
כשל בחיבור	CONNECTION_ERROR	E009
токן אימות חסר	AUTH_TOKEN_MISSING	E011
токן לא תקין	AUTH_TOKEN_INVALID	E012

2.9.4 מדיניות ניסיון חוזר – Retry Policy

שגיאות מסוימות ניתנות לניסיון חוזר:

- מקסימום ניסיונות: 3
- השהייה בין ניסיונות: 2 שניות
- **שגיאות שניתנות לניסיון:** (noitcennoc) E009 , (tuoemit) E001 (TECHNICAL LOSS).

2.10 תאימות גרסאות

2.10.1 הכרזת גרסה

בעת רישום, כל סוכן מכיריז על גרסת הפרוטוקול שהוא תומך בה. מנהל הליגה בודק תאימות לפני אישור הרישום.

הכרזת גרסה בבקשת רישום

```
{
  "message_type": "LEAGUE_REGISTER_REQUEST",
  "player_meta": {
    "display_name": "Agent\u2014Alpha",
    "version": "1.0.0",
    "protocol_version": "2.1.0",
    "game_types": ["even_odd"]
  }
}
```

2.10.2 מדיניות תאימות

- **גרסה נוכחית:** 2.1.0
- **גרסה מינימלית נתמכת:** 2.0.0
- סוכנים עם גרסה ישנה יותר יקבלו שגיאת PROTOCOL_VERSION_MIS- E018 .MATCH)

2.11 עקרונות חשובים

2.11.1 מקור אמת יחיד

השופט הוא מקור האמת למצבי המשחק. שחקנים לא שומרים מצב משליהם. הם משתמשים על המידע שהשופט שולח.

2.11.2 תקשורת דרך Orchestrator

שחקנים לא מדברים ישירות זה עם זה. כל התקשרות עוברת דרך השופט או מנהל הליגה. זה מבטיח שהפרוטוקול נשמר.

2.11.3 טיפול בכשלים

אם שחקן לא מגיב:

1. השופט שולח הודעה GAME_ERROR עם .retryable=true עד 3 ניסיונות.
2. השחקן מקבל עד 3 ניסיונות.
3. לאחר מיצוי הניסיונות – הפסד טכני (TECHNICAL_LOSS).