

## 2 פרוטוקול הליגה הכללי

### 2.1 עקרונות הפרוטוקול

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

#### 2.1.1 הפרדה לשלוש שכבות

המערכת מורכבת משלוש שכבות:

1. **שכבת הליגה** – ניהול טורניר, רישום שחקנים, טבלת דירוג.
  2. **שכבת השיפוט** – ניהול משחק בודד, אימות מהלכים, הכרזת מנצח.
  3. **שכבת חוקי המשחק** – לוגיקת משחק ספציפית (זוגי/אי-זוגי, איקס-עיגול, וכד').
- 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 שחקנים 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 לשפוט משחקים, הוא חייב להירשם אצל מנהל הליגה.

השופט אחראי על:

- **רישום למנהל הליגה** – לפני תחילת הליגה.
- הזמנת שני שחקנים למשחק.
- ניהול תורות המשחק.
- אימות חוקיות מהלכים.
- הכרזה על תוצאה ודיווח לליגה.

השופט פועל כשרת MCP על פורט 8001. יכולים 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 (8101-8104).

**2.3 מזהים בפרוטוקול**

כל רכיב במערכת מזוהה באופן ייחודי.

טבלה 1: מזהים בפרוטוקול הליגה

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

**2.3.1 דוגמאות למזהים**

- league\_id: "league\_2025\_even\_odd"
- round\_id: 1, 2, 3, ...
- match\_id: "R1M1" (מחזור 1, משחק 1)
- game\_type: "even\_odd" או "tic\_tac\_toe"
- player\_id: "P01", "P02", ..., "P20"
- referee\_id: "REF01", "REF02", ...

## 2.4 מבנה הודעה כללי – Envelope

כל הודעה בפרוטוקול חייבת לכלול "מעטפת" (Envelope) עם שדות קבועים. המעטפת 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"
}
```

### 2.4.1 שדות חובה במעטפת

טבלה 2: שדות חובה בכל הודעה

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

## 2.4.2 דרישת אזור זמן – UTC/GMT

**חובה:** כל חותמות הזמן בפרוטוקול חייבות להיות באזור זמן UTC/GMT. דרישה זו מבטיחה עקביות בין סוכנים הפועלים ממיקומים גיאוגרפיים שונים.

טבלה 3: פורמטים תקינים ולא תקינים לחותמת זמן

פורמט	תקין?	הסבר
2025-01-15T10:30:00Z	✓	סיומת Z מציינת UTC
2025-01-15T10:30:00+00:00	✓	הפרש +00:00 שווה ערך ל-UTC
2025-01-15T10:30:00+02:00	✗	אזור זמן מקומי – אסור
2025-01-15T10:30:00	✗	ללא אזור זמן – אסור

**הערה חשובה:** סוכן ששולח הודעה עם אזור זמן שאינו UTC יקבל שגיאה E021 (IN-VALID\_TIMESTAMP).

## 2.4.3 שדות אופציונליים

טבלה 4: שדות אופציונליים לפי הקשר

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

## 2.4.4 פורמט שדה sender

שדה sender מזהה את שולח ההודעה:

- league\_manager – מנהל הליגה.

- referee:REF01 – שופט עם מזהה REF01.

- player:P01 – שחקן עם מזהה 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 לסוכנים אחרים.

##### קבלת טוקן בתגובת רישום

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

### 2.5 זרימת הליגה הכללית

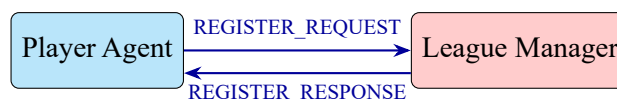
#### 2.5.1 שלב 1: רישום שופטים

בשלב הראשון, כל שופט נרשם לליגה. השופט שולח בקשת רישום למנהל הליגה. מנהל הליגה מקצה referee\_id ושומר את כתובת השופט.



#### 2.5.2 שלב 2: רישום שחקנים

לאחר רישום השופטים, כל שחקן נרשם לליגה. השחקן שולח בקשת רישום למנהל הליגה. מנהל הליגה מקצה player\_id ומאשר.



#### 2.5.3 שלב 3: יצירת 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 הודעת ROUND\_ANNOUNCEMENT. ההודעה מפרטת את כל המשחקים במחזור. מנהל הליגה מקצה שופט לכל משחק מתוך רשימת השופטים הרשומים.

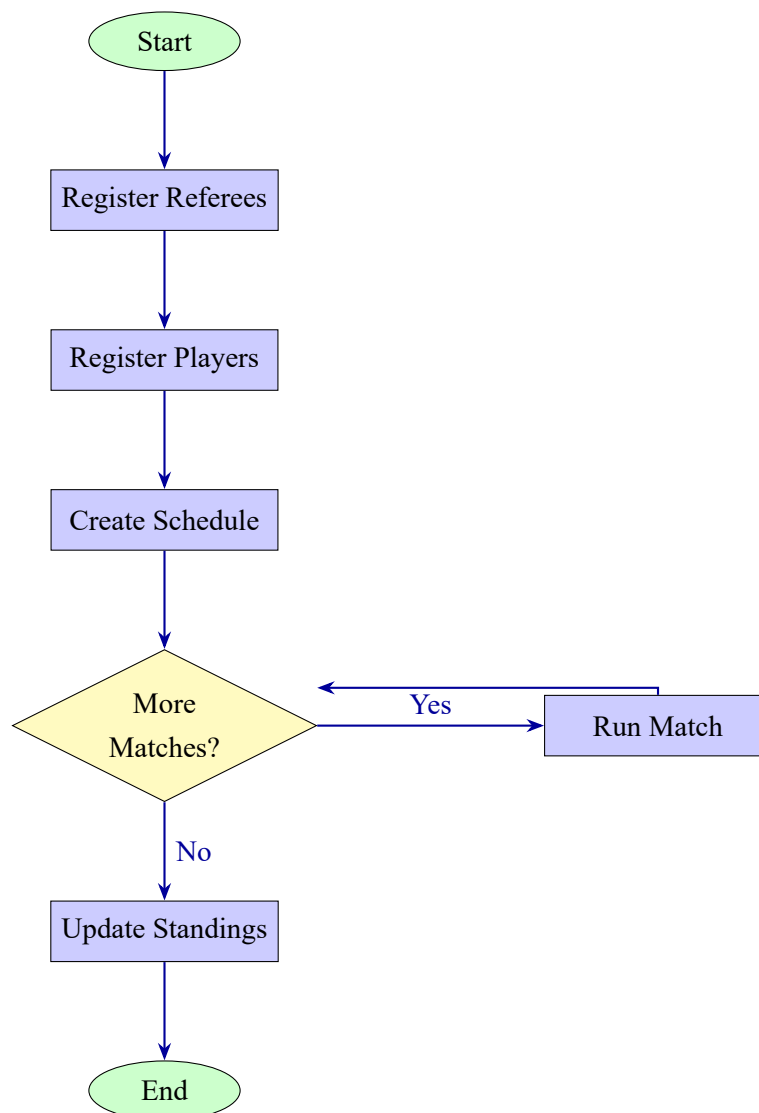
### 2.5.5 שלב 5: ניהול משחקים

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 הודעה מוגדר זמן תגובה מקסימלי. אם הסוכן לא עונה בזמן, הפעולה נחשבת לכישלון.

טבלה 5: זמני תגובה לפי סוג הודעה

הערות	Timeout	סוג הודעה
רישום שופט לליגה	10 sec	REFEREE_REGISTER
רישום שחקן לליגה	10 sec	LEAGUE_REGISTER
אישור הגעה למשחק	5 sec	GAME_JOIN_ACK
בחירת זוגי/אי-זוגי	30 sec	CHOOSE_PARITY
קבלת תוצאת משחק	5 sec	GAME_OVER
דיווח תוצאה לליגה	10 sec	MATCH_RESULT_REPORT
שאלת מידע	10 sec	LEAGUE_QUERY
ברירת מחדל	10 sec	כל השאר

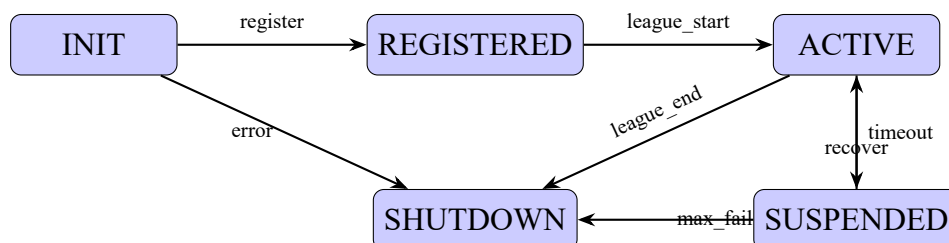
## 2.8 מחזור חיים של סוכן – Agent Lifecycle

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

### 2.8.1 מצבי הסוכן

- INIT – הסוכן הופעל אך טרם נרשם.
- REGISTERED – הסוכן נרשם בהצלחה וקיבל auth\_token.
- ACTIVE – הסוכן פעיל ומשתתף במשחקים.
- SUSPENDED – הסוכן מושעה זמנית (לא מגיב).
- SHUTDOWN – הסוכן סיים פעילות.

### 2.8.2 דיאגרמת מעברי מצבים



## 2.9 טיפול בשגיאות

הפרוטוקול מגדיר שני סוגי הודעות שגיאה:

### 2.9.1 שגיאת ליגה – LEAGUE\_ERROR

מנהל הליגה שולח הודעה זו כאשר מתרחשת שגיאה ברמת 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_ID_not_found_in_registry",
  "context": {
    "player_id": "P99"
  },
  "retryable": false
}
```

### 2.9.2 שגיאת משחק – GAME\_ERROR

השופט שולח הודעה זו כאשר מתרחשת שגיאה במשחק.

#### 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_not_received_within_30_seconds",
  "game_state": "COLLECTING_CHOICES",
  "retryable": true,
  "retry_count": 1,
  "max_retries": 3
}
```



### 2.9.3 קודי שגיאה נפוצים

טבלה 6: קודי שגיאה עיקריים

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

### 2.9.4 מדיניות ניסיון חוזר – Retry Policy

שגיאות מסוימות ניתנות לניסיון חוזר:

- מקסימום ניסיונות: 3
- השהייה בין ניסיונות: 2 שניות
- שגיאות הניתנות לניסיון: E001 (tuomit), E009 (noitcennoc)
- לאחר מיצוי הניסיונות – הפסד טכני (TECHNICAL\_LOSS).

### 2.10 תאימות גרסאות

### 2.10.1 הכרזת גרסה

בעת רישום, כל סוכן מכריז 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_Alpha",
    "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 (E018- PROTOCOL\_VERSION\_MISMATCH).

## 2.11 עקרונות חשובים

### 2.11.1 מקור אמת יחיד

השופט הוא מקור האמת למצב המשחק. שחקנים לא שומרים מצב משלהם. הם מסתמכים על המידע שהשופט שולח.

### 2.11.2 תקשורת דרך Orchestrator

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

### 2.11.3 טיפול בכשלים

אם שחקן לא מגיב:

1. השופט שולח הודעת GAME\_ERROR עם retryable=true.
2. השחקן מקבל עד 3 ניסיונות.
3. לאחר מיצוי הניסיונות – הפסד טכני (TECHNICAL\_LOSS).