

8 הרצת מערכת הליגה

נספח זה מציג הדרכה מעשית להרצת מערכת הליגה המלאה. נדגים כיצד להפעיל את כל הסוכנים ולנהל ליגה עם מנהל ליגה אחד, שני שופטים וארבעה שחקנים. הדוגמאות מבוססות על פרוטוקול league.v2 המתואר בפרקים הקודמים.

8.1 תצורת המערכת

8.1.1 פורטים וטרמינלים

כל סוכן במערכת פועל כשרת HTTP נפרד על פורט ייעודי ב-localhost. בדוגמה זו נריץ 7 טרמינלים:

טבלה 15: הקצאת פורטים וטרמינלים

טרמינל	סוכן	פורט	Endpoint
1	מנהל ליגה	8000	http://localhost:8000/mcp
2	שופט REF01	8001	http://localhost:8001/mcp
3	שופט REF02	8002	http://localhost:8002/mcp
4	שחקן P01	8101	http://localhost:8101/mcp
5	שחקן P02	8102	http://localhost:8102/mcp
6	שחקן P03	8103	http://localhost:8103/mcp
7	שחקן P04	8104	http://localhost:8104/mcp

8.1.2 תפקידי האורקסטרטורים

במערכת שני סוגי Orchestrator:

- **מנהל הליגה** – Orchestrator עליון של הליגה. הוא מקור האמת לטבלת הדירוג, לוח המשחקים ומצב המחזורים.

- **השופטים** – Orchestratorים מקומיים למשחק בודד. כל שופט הוא מקור האמת למצב המשחק שלו.

8.2 סדר הפעלה

8.2.1 עקרון סדר ההפעלה

סדר ההפעלה קריטי לתפקוד תקין של המערכת:

1. מנהל הליגה – חייב לעלות ראשון.
2. שופטים – עולים ונרשמים למנהל הליגה.
3. שחקנים – עולים ונרשמים למנהל הליגה.
4. תחילת הליגה – רק לאחר השלמת כל הרישומים.

8.2.2 טרמינל 1 – מנהל הליגה

הפעלת מנהל הליגה

```
# Terminal 1 - League Manager  
python league_manager.py # Listening on :8000
```

מנהל הליגה מאזין לבקשות POST בכתובת `http://localhost:8000/mcp`.

8.2.3 טרמינלים 2-3 – שופטים

הפעלת שופטים

```
# Terminal 2 - Referee Alpha  
python referee.py --port 8001  
  
# Terminal 3 - Referee Beta  
python referee.py --port 8002
```

כל שופט, בעת עלייתו, מפעיל פונקציה `register_to_league` ששולחת REF-
EREE_REGISTER_REQUEST למנהל הליגה.

הפעלת שחקנים

```
# Terminal 4 - Player P01
python player.py --port 8101
```

```
# Terminal 5 - Player P02
python player.py --port 8102
```

```
# Terminal 6 - Player P03
python player.py --port 8103
```

```
# Terminal 7 - Player P04
python player.py --port 8104
```

כל שחקן שולח LEAGUE_REGISTER_REQUEST למנהל הליגה.

8.3 שלב 1: רישום שופטים

כל שופט, מיד עם עליית השרת שלו, קורא מצד הלקוח למנהל הליגה.

8.3.1 בקשת רישום שופט**REFEREE_REGISTER_REQUEST – בקשת רישום שופט**

```
{
  "jsonrpc": "2.0",
  "method": "register_referee",
  "params": {
    "protocol": "league.v2",
    "message_type": "REFEREE_REGISTER_REQUEST",
    "sender": "referee:alpha",
    "timestamp": "2025-01-15T10:00:00Z",
    "conversation_id": "conv-ref-alpha-reg-001",
    "referee_meta": {
      "display_name": "Referee_αAlpha",
      "version": "1.0.0",
      "game_types": ["even_odd"],
      "contact_endpoint": "http://localhost:8001/mcp",
      "max_concurrent_matches": 2
    }
  },
  "id": 1
}
```

REFEREE_REGISTER_RESPONSE – תגובת רישום שופט

```
{
  "jsonrpc": "2.0",
  "result": {
    "protocol": "league.v2",
    "message_type": "REFEREE_REGISTER_RESPONSE",
    "sender": "league_manager",
    "timestamp": "2025-01-15T10:00:01Z",
    "conversation_id": "conv-ref-alpha-reg-001",
    "status": "ACCEPTED",
    "referee_id": "REF01",
    "auth_token": "tok-ref01-abc123",
    "league_id": "league_2025_even_odd",
    "reason": null
  },
  "id": 1
}
```

השופט השני (על פורט 8002) שולח בקשה דומה ומקבל "REF02".referee_id:

8.4 שלב 2: רישום שחקנים

לאחר שהשופטים נרשמו, כל שחקן שולח בקשת רישום.

8.4.1 בקשת רישום שחקן

LEAGUE_REGISTER_REQUEST – בקשת רישום שחקן

```
{
  "jsonrpc": "2.0",
  "method": "register_player",
  "params": {
    "protocol": "league.v2",
    "message_type": "LEAGUE_REGISTER_REQUEST",
    "sender": "player:alpha",
    "timestamp": "2025-01-15T10:05:00Z",
    "conversation_id": "conv-player-alpha-reg-001",
    "player_meta": {
      "display_name": "Agent_αAlpha",
      "version": "1.0.0",
      "game_types": ["even_odd"],
      "contact_endpoint": "http://localhost:8101/mcp"
    }
  },
  "id": 1
}
```

LEAGUE_REGISTER_RESPONSE – תגובת רישום שחקן

```
{
  "jsonrpc": "2.0",
  "result": {
    "protocol": "league.v2",
    "message_type": "LEAGUE_REGISTER_RESPONSE",
    "sender": "league_manager",
    "timestamp": "2025-01-15T10:05:01Z",
    "conversation_id": "conv-player-alpha-reg-001",
    "status": "ACCEPTED",
    "player_id": "P01",
    "auth_token": "tok-p01-xyz789",
    "league_id": "league_2025_even_odd",
    "reason": null
  },
  "id": 1
}
```

באופן דומה:

- שחקן על פורט 8102 מקבל "P02" player_id
 - שחקן על פורט 8103 מקבל "P03" player_id
 - שחקן על פורט 8104 מקבל "P04" player_id

מנהל הליגה שומר מפה player_id → contact_endpoint ומפה referee_id → contact_endpoint.

8.5 שלב 3: יצירת לוח משחקים

לאחר שכל השחקנים והשופטים נרשמו, מנהל הליגה מפעיל לוגיקת create_schedule על רשימת השחקנים ל-Round-Robin.

8.5.1 לוח משחקים לארבעה שחקנים

טבלה 16: לוח משחקים Round-Robin לארבעה שחקנים

Match ID	Player A	Player B
R1M1	P01	P02
R1M2	P03	P04
R2M1	P03	P01
R2M2	P04	P02
R3M1	P04	P01
R3M2	P03	P02

8.6 שלב 4: הכרזה על מחזור

מנהל הליגה שולח לכל השחקנים הודעת ROUND_ANNOUNCEMENT.

הכרזה על מחזור – ROUND_ANNOUNCEMENT

```
{
  "jsonrpc": "2.0",
  "method": "notify_round",
  "params": {
    "protocol": "league.v2",
    "message_type": "ROUND_ANNOUNCEMENT",
    "sender": "league_manager",
    "timestamp": "2025-01-15T10:10:00Z",
    "conversation_id": "conv-round-1-announce",
    "league_id": "league_2025_even_odd",
    "round_id": 1,
    "matches": [
      {
        "match_id": "R1M1",
        "game_type": "even_odd",
        "player_A_id": "P01",
        "player_B_id": "P02",
        "referee_endpoint": "http://localhost:8001/mcp"
      },
      {
        "match_id": "R1M2",
        "game_type": "even_odd",
        "player_A_id": "P03",
        "player_B_id": "P04",
        "referee_endpoint": "http://localhost:8001/mcp"
      }
    ]
  },
  "id": 10
}
```

מרגע שנשלחה ROUND_ANNOUNCEMENT – המחזור התחיל לוגית. המשחקים עצמם מתחילים רק כאשר השופט מזמין את המשתתפים.

8.7 שלב 5: ניהול משחק בודד

נתאר את זרימת המשחק R1M1: שחקן P01 נגד שחקן P02, שופט REF01.

השופט מעביר את מצב המשחק ל-WAITING_FOR_PLAYERS ושולח GAME_INVITATION לכל שחקן.

הזמנה ל-P01:

הזמנה למשחק – GAME_INVITATION ל-P01

```
{
  "jsonrpc": "2.0",
  "method": "handle_game_invitation",
  "params": {
    "protocol": "league.v2",
    "message_type": "GAME_INVITATION",
    "sender": "referee:REF01",
    "timestamp": "2025-01-15T10:15:00Z",
    "conversation_id": "conv-rlm1-001",
    "auth_token": "tok-ref01-abc123",
    "league_id": "league_2025_even_odd",
    "round_id": 1,
    "match_id": "R1M1",
    "game_type": "even_odd",
    "role_in_match": "PLAYER_A",
    "opponent_id": "P02"
  },
  "id": 1001
}
```

הזמנה ל-P02:

P02-5 GAME_INVITATION – הזמנה למשחק

```
{
  "jsonrpc": "2.0",
  "method": "handle_game_invitation",
  "params": {
    "protocol": "league.v2",
    "message_type": "GAME_INVITATION",
    "sender": "referee:REF01",
    "timestamp": "2025-01-15T10:15:00Z",
    "conversation_id": "conv-rlm1-001",
    "auth_token": "tok-ref01-abc123",
    "league_id": "league_2025_even_odd",
    "round_id": 1,
    "match_id": "R1M1",
    "game_type": "even_odd",
    "role_in_match": "PLAYER_B",
    "opponent_id": "P01"
  },
  "id": 1002
}
```

8.7.2 שלב 5.2: אישורי הגעה

כל שחקן מחזיר GAME_JOIN_ACK תוך 5 שניות.

אישור מ-P01:

אישור הגעה - GAME_JOIN_ACK מ-P01

```
{
  "jsonrpc": "2.0",
  "result": {
    "protocol": "league.v2",
    "message_type": "GAME_JOIN_ACK",
    "sender": "player:P01",
    "timestamp": "2025-01-15T10:15:01Z",
    "conversation_id": "conv-rlm1-001",
    "auth_token": "tok-p01-xyz789",
    "match_id": "R1M1",
    "player_id": "P01",
    "arrival_timestamp": "2025-01-15T10:15:01Z",
    "accept": true
  },
  "id": 1001
}
```

אישור מ-P02:

אישור הגעה - GAME_JOIN_ACK מ-P02

```
{
  "jsonrpc": "2.0",
  "result": {
    "protocol": "league.v2",
    "message_type": "GAME_JOIN_ACK",
    "sender": "player:P02",
    "timestamp": "2025-01-15T10:15:02Z",
    "conversation_id": "conv-rlm1-001",
    "auth_token": "tok-p02-def456",
    "match_id": "R1M1",
    "player_id": "P02",
    "arrival_timestamp": "2025-01-15T10:15:02Z",
    "accept": true
  },
  "id": 1002
}
```

כאשר השופט קיבל שני ACK חיוביים בזמן המותר, הוא מעביר את מצב המשחק ל-COLLECTING_CHOICES.

השופט שולח CHOOSE_PARITY_CALL לכל שחקן.

בקשת בחירה ל-P01:

בקשת בחירה - CHOOSE_PARITY_CALL ל-P01

```
{
  "jsonrpc": "2.0",
  "method": "choose_parity",
  "params": {
    "protocol": "league.v2",
    "message_type": "CHOOSE_PARITY_CALL",
    "sender": "referee:REF01",
    "timestamp": "2025-01-15T10:15:05Z",
    "conversation_id": "conv-r1m1-001",
    "auth_token": "tok-ref01-abc123",
    "match_id": "R1M1",
    "player_id": "P01",
    "game_type": "even_odd",
    "context": {
      "opponent_id": "P02",
      "round_id": 1,
      "your_standings": {
        "wins": 0,
        "losses": 0,
        "draws": 0
      }
    }
  },
  "deadline": "2025-01-15T10:15:35Z"
},
{id": 1101
}
```

תגובת P01 (בחר "even"):

תגובת בחירה - P01-מ CHOOSE_PARITY_RESPONSE

```
{
  "jsonrpc": "2.0",
  "result": {
    "protocol": "league.v2",
    "message_type": "CHOOSE_PARITY_RESPONSE",
    "sender": "player:P01",
    "timestamp": "2025-01-15T10:15:10Z",
    "conversation_id": "conv-rlm1-001",
    "auth_token": "tok-p01-xyz789",
    "match_id": "R1M1",
    "player_id": "P01",
    "parity_choice": "even"
  },
  "id": 1101
}
```

תגובת P02 (בחר "odd"):

תגובת בחירה - P02-מ CHOOSE_PARITY_RESPONSE

```
{
  "jsonrpc": "2.0",
  "result": {
    "protocol": "league.v2",
    "message_type": "CHOOSE_PARITY_RESPONSE",
    "sender": "player:P02",
    "timestamp": "2025-01-15T10:15:12Z",
    "conversation_id": "conv-rlm1-001",
    "auth_token": "tok-p02-def456",
    "match_id": "R1M1",
    "player_id": "P02",
    "parity_choice": "odd"
  },
  "id": 1102
}
```

כאשר שתי הבחירות התקבלו תקין ובזמן, השופט מעביר את מצב המשחק ל-DRAWING_NUMBER.

8.7.4 שלב 5.4: הגרלת מספר וקביעת מנצח

השופט מגריל מספר בין 1 ל-10, למשל 8. הוא מפעיל את מודול חוקי המשחק:

$\text{drawn_number} = 8$ -

```

number_parity = "even" -
בחירת P01 → "even" = צדק -
בחירת P02 → "odd" = טעה -
winner_player_id = "P01" -
status = "WIN" -
מצב המשחק עובר ל-FINISHED.

```

8.7.5 שלב 5.5: הודעת סיום לשחקנים

השופט שולח GAME_OVER לשני השחקנים:

סיום משחק - GAME_OVER

```

{
  "jsonrpc": "2.0",
  "method": "notify_match_result",
  "params": {
    "protocol": "league.v2",
    "message_type": "GAME_OVER",
    "sender": "referee:REF01",
    "timestamp": "2025-01-15T10:15:30Z",
    "conversation_id": "conv-r1m1-001",
    "auth_token": "tok-ref01-abc123",
    "match_id": "R1M1",
    "game_type": "even_odd",
    "game_result": {
      "status": "WIN",
      "winner_player_id": "P01",
      "drawn_number": 8,
      "number_parity": "even",
      "choices": {
        "P01": "even",
        "P02": "odd"
      },
      "reason": "P01 chose even, number was 8 (even)"
    }
  },
  "id": 1201
}

```

כל שחקן מעדכן מצב פנימי (סטטיסטיקות, היסטוריה) ומחזיר תשובה כללית.

השופט מדווח MATCH_RESULT_REPORT לליגה:

MATCH_RESULT_REPORT - דיווח תוצאה

```
{
  "jsonrpc": "2.0",
  "method": "report_match_result",
  "params": {
    "protocol": "league.v2",
    "message_type": "MATCH_RESULT_REPORT",
    "sender": "referee:REF01",
    "timestamp": "2025-01-15T10:15:35Z",
    "conversation_id": "conv-r1m1-report",
    "auth_token": "tok-ref01-abc123",
    "league_id": "league_2025_even_odd",
    "round_id": 1,
    "match_id": "R1M1",
    "game_type": "even_odd",
    "result": {
      "winner": "P01",
      "score": {
        "P01": 3,
        "P02": 0
      },
      "details": {
        "drawn_number": 8,
        "choices": {
          "P01": "even",
          "P02": "odd"
        }
      }
    }
  },
  "id": 1301
}
```

מנהל הליגה מעדכן טבלת נקודות בהתאם לטבלת הניקוד (ניצחון = 3 נקודות).

8.8 שלב 6: סיום מחזור ועדכון דירוג

מחזור מספר 1 מסתיים כאשר לכל משחקי המחזור התקבל MATCH_RESULT_REPORT.
מנהל הליגה:

1. מכריז שהמחזור סגור (אפשר לעבור ל-round_id=2).
2. מחשב טבלת דירוג: points, wins, draws, losses, played לכל שחקן.
3. שולח לכל השחקנים הודעת LEAGUE_STANDINGS_UPDATE.

LEAGUE_STANDINGS_UPDATE - עדכון דירוג

```
{
  "jsonrpc": "2.0",
  "method": "update_standings",
  "params": {
    "protocol": "league.v2",
    "message_type": "LEAGUE_STANDINGS_UPDATE",
    "sender": "league_manager",
    "timestamp": "2025-01-15T10:20:00Z",
    "conversation_id": "conv-round-1-standings",
    "league_id": "league_2025_even_odd",
    "round_id": 1,
    "standings": [
      {
        "rank": 1,
        "player_id": "P01",
        "display_name": "Agent_Alpha",
        "played": 1,
        "wins": 1,
        "draws": 0,
        "losses": 0,
        "points": 3
      },
      {
        "rank": 2,
        "player_id": "P03",
        "display_name": "Agent_Gamma",
        "played": 1,
        "wins": 0,
        "draws": 1,
        "losses": 0,
        "points": 1
      },
      {
        "rank": 3,
        "player_id": "P04",
        "display_name": "Agent_Delta",
        "played": 1,
        "wins": 0,
        "draws": 1,
        "losses": 0,
        "points": 1
      },
      {
        "rank": 4,
        "player_id": "P02",
        "display_name": "Agent_Beta",
        "played": 1,
        "wins": 0,
        "draws": 0,
        "losses": 1,

```

לאחר שליחת עדכון הדירוג, מנהל הליגה שולח הודעת ROUND_COMPLETED לסמן את סיום המחזור:

סיום מחזור - ROUND_COMPLETED

```
{
  "jsonrpc": "2.0",
  "method": "notify_round_completed",
  "params": {
    "protocol": "league.v2",
    "message_type": "ROUND_COMPLETED",
    "sender": "league_manager",
    "timestamp": "2025-01-15T10:20:05Z",
    "conversation_id": "conv-round-1-complete",
    "league_id": "league_2025_even_odd",
    "round_id": 1,
    "matches_played": 2,
    "next_round_id": 2
  },
  "id": 1402
}
```

8.9 שלב 7: סיום הליגה

לאחר סיום כל המחזורים, מנהל הליגה שולח הודעת LEAGUE_COMPLETED:

LEAGUE_COMPLETED – סיום ליגה

```
{
  "jsonrpc": "2.0",
  "method": "notify_league_completed",
  "params": {
    "protocol": "league.v2",
    "message_type": "LEAGUE_COMPLETED",
    "sender": "league_manager",
    "timestamp": "2025-01-15T12:00:00Z",
    "conversation_id": "conv-league-complete",
    "league_id": "league_2025_even_odd",
    "total_rounds": 3,
    "total_matches": 6,
    "champion": {
      "player_id": "P01",
      "display_name": "Agent Alpha",
      "points": 7
    },
    "final_standings": [
      {"rank": 1, "player_id": "P01", "points": 7},
      {"rank": 2, "player_id": "P03", "points": 5},
      {"rank": 3, "player_id": "P04", "points": 4},
      {"rank": 4, "player_id": "P02", "points": 2}
    ]
  },
  "id": 2001
}
```

8.10 טיפול בשגיאות

כאשר מתרחשת שגיאה, מנהל הליגה או השופט שולחים הודעת שגיאה מתאימה.

LEAGUE_ERROR – שגיאת אימות

```
{
  "jsonrpc": "2.0",
  "result": {
    "protocol": "league.v2",
    "message_type": "LEAGUE_ERROR",
    "sender": "league_manager",
    "timestamp": "2025-01-15T10:05:30Z",
    "conversation_id": "conv-error-001",
    "error_code": "E012",
    "error_description": "AUTH_TOKEN_INVALID",
    "context": {
      "provided_token": "tok-invalid-xxx",
      "action": "LEAGUE_QUERY"
    }
  },
  "id": 1502
}
```

GAME_ERROR – שגיאת משחק

```
{
  "jsonrpc": "2.0",
  "method": "notify_game_error",
  "params": {
    "protocol": "league.v2",
    "message_type": "GAME_ERROR",
    "sender": "referee:REF01",
    "timestamp": "2025-01-15T10:16:00Z",
    "conversation_id": "conv-r1m1-001",
    "match_id": "R1M1",
    "error_code": "E001",
    "error_description": "TIMEOUT_ERROR",
    "affected_player": "P02",
    "action_required": "CHOOSE_PARITY_RESPONSE",
    "retry_count": 0,
    "max_retries": 3,
    "consequence": "Technical_loss_if_no_response_after_retries"
  },
  "id": 1103
}
```

8.11 כלי שאילתות זמינים

המסמך מגדיר כלי MCP גנריים שכל סוכן יכול לחשוף לצרכי דיבוג ובירור.

8.11.1 שאילתת דירוג ממנהל הליגה

שחקן שרוצה לוודא את דירוגו קורא למנהל הליגה:

LEAGUE_QUERY – שאילתת דירוג

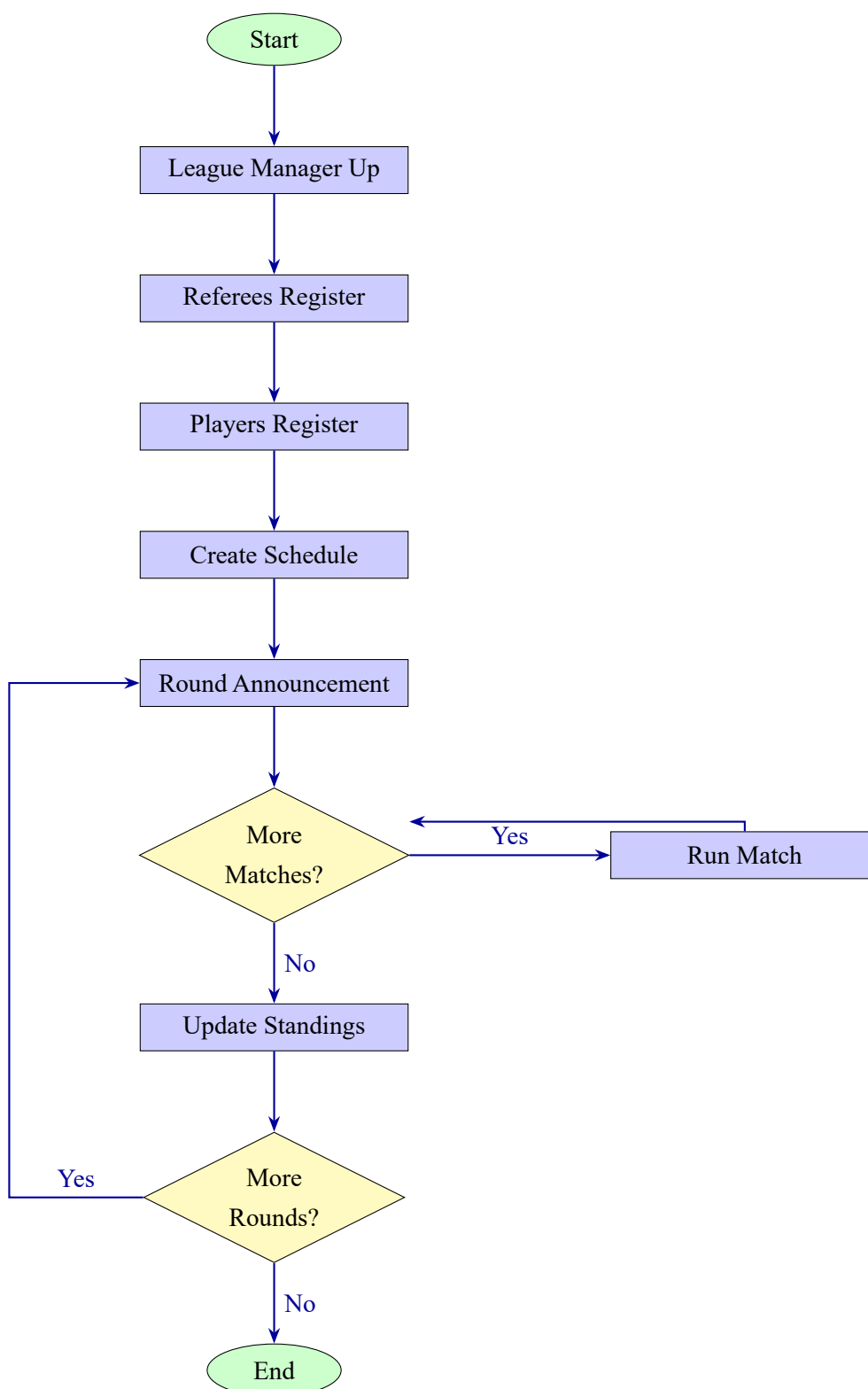
```
{
  "jsonrpc": "2.0",
  "method": "league_query",
  "params": {
    "protocol": "league.v2",
    "message_type": "LEAGUE_QUERY",
    "sender": "player:P01",
    "timestamp": "2025-01-15T10:25:00Z",
    "conversation_id": "conv-query-standings-001",
    "auth_token": "tok-p01-xyz789",
    "league_id": "league_2025_even_odd",
    "query_type": "GET_STANDINGS"
  },
  "id": 1501
}
```

מנהל הליגה מחזיר `result` שכולל אובייקט `standings` בפורמט זה ל-`LEAGUE_STANDINGS_UPDATE`.

8.11.2 כלים נוספים

- **כלי במנהל הליגה:** `get_standings` – מחזיר את מצב הטבלה העדכני.
- **כלי בשופט:** `get_match_state` – מחזיר מצב משחק קיים (לצרכי דיבוג).
- **כלי בשחקן:** `get_player_state` – נותן את היסטוריית המשחקים.

8.12 דיאגרמת זרימה מלאה



8.13 טבלת תפקידי הסוכנים

טבלה 17: תפקידי הסוכנים במערכת

מתקשר עם	תפקיד בליגה	פורט	סוכן
שופטים ושחקנים	Orchestrator ליגה, טבלה, מחזורים	8000	League Manager
מנהל ליגה, שחקנים	Orchestrator משחקים	8001	ahplA Referee
מנהל ליגה, שחקנים	Orchestrator משחקים	8002	ateB Referee
שופט, מנהל ליגה	שחקן, בוחר even/odd	8101	P01 Player Agent
שופט, מנהל ליגה	שחקן	8102	P02 Player Agent
שופט, מנהל ליגה	שחקן	8103	P03 Player Agent
שופט, מנהל ליגה	שחקן	8104	P04 Player Agent

8.14 סיכום

נספח זה הציג את:

- תצורת המערכת: הקצאת פורטים וטרמינלים ל-7 סוכנים.
- סדר ההפעלה: מנהל ליגה → שופטים → שחקנים.
- זרימת הרישום: REFEREE_REGISTER ו-LEAGUE_REGISTER עם קבלת auth_token.
- ניהול מחזור: ROUND_ANNOUNCEMENT ו-ROUND_COMPLETED.
- ניהול משחק: מ-GAME_INVITATION עד GAME_OVER.
- עדכון דירוג: MATCH_RESULT_REPORT ו-LEAGUE_STANDINGS_UPDATE.
- סיום ליגה: LEAGUE_COMPLETED עם הכרזת אלוף.
- טיפול בשגיאות: LEAGUE_ERROR ו-GAME_ERROR.
- שאילתות: LEAGUE_QUERY לקבלת מידע מעודכן.

כל התקשורת מתבצעת באמצעות JSON-RPC 2.0 מעל HTTP. כל ההודעות כוללות מעטפת (Envelope) אחידה עם שדות חובה: timestamp, sender, message_type, protocol (באזור זמן UTC), ו-conversation_id. האורקסטרטורים (מנהל ליגה ושופטים) מנהלים את זרימת ההודעות בכל רגע.