

## **סיכום שיעור 7**

### **פרוטוקול MCP וliget סוכנים**

**מרצה: ד"ר יורם סגל**

**קורס: סוכני בינה מלאכותית**

#### **מילות מפתח:**

- פרוטוקול תקשורת • סוכנים • ליגת • כלים (Tools) • Client-Server • MCP
- משאבים (Resources) • ניהול שינויים • JSON • ארכיטקטורה מבווארת

## תוכן העניינים

2	1	מבוא ל <u>פרוטוקול MCP</u>
2	1.1	רקע והתפתחות
2	2.1	הבעיה שהפרוטוקול פותר
2	2	<b>ארQUITטורת ב-Client-Server MCP</b>
2	1.2	הגדרות בסיסיות
3	2.2	מחזור חי הchipbor
3	3	<b>רכיבי הפרוטוקול</b>
3	1.3	כליים (Tools)
3	2.3	משאבים (Resources)
3	3.3	הנחיות (Prompts)
3	4	<b>ארQUITטורת הקליענט</b>
3	1.4	שכבות הקליענט
4	2.4	מודולים הכרחיים
4	1.2.4	מנהל הסשנים (Session Manager)
4	2.2.4	רישום כלים (Tool Registry)
4	3.2.4	ניהול תור הודעות (Message Queue)
4	3.4	טיפול בשגיאות
5	5	<b>עבודה עם מספר שירותים</b>
5	1.5	טופולוגיה כוכב
5	2.5	אייזון עומסים (Load Balancing)
5	6	<b>אבטחת מידע</b>
5	1.6	איומים נפוצים
5	2.6	הגנות מומלצות
5	7	<b>מטלה: לגთ סוכנים</b>
5	1.7	תיאור המשימה
6	2.7	חוקי המשחק
6	3.7	עקרונות ארכיטקטוניים
6	4.7	מבנה הودעה ב <u>פרוטוקול</u>
6	8	<b>סיכום</b>
6	1.8	נקודות מפתח
7	2.8	הכנה לפרויקט גמר

# 1 מבוא ל פרוטוקול MCP

## 1.1 רקע והתפתחות

פרוטוקול MCP (Model Context Protocol) הוא תקן תקשורת שפותח על ידי Anthropic לאפשר תקשורת אחדה בין מודלי שפה (LLM) לבין שירותים חיצוניים. ה프וטוקול מהווה מהפכה בתחום סוכני הבינה המלאכותית.

**התפתחות היסטורית:**

- עיתונים – דרך תקשורת ראשונית
- HTTP – פרוטוקול האינטרנט
- USB – סטנדרט להעברת נתונים פיזית
- MCP – פרוטוקול לתקשורת בין סוכני AI

## 2.1 הבניה של פרוטוקול פוטר

לפני MCP, כל סוכן נדרש לאינטגרציה ייחודית לכל שירות. מצב זה יצר:

- מורכבות של  $O(n \times m)$  – למשל, 5 סוכנים כפול 5 שירותי = 52 אינטגרציות
- קושי בתחזוקה והרחבה
- בעיות סקלבilities.

עם MCP: המורכבות יורדת ל- $O(n + m)$  – למשל, 5 סוכנים + 5 שירותי = 10 חיבורים בלבד.

## 2 ארכיטקטורת Client-Server ב-MCP

### 1.2 הגדרות בסיסיות

**קליעין (Client):**

- יוזם את הפניה לשרת
- מכיל את מודל השפה (LLM)
- מנהל שניים, כלים, הודעות ומשאבים
- הוא "המוח" של המערכת

**שרת (Server):**

- פסיבי – ממතין לבקשתות
- מספק כלים (Tools) ומשאבים (Resources)
- הוא "הידיים והרגליים" של המערכת
- מבצע פעולות דטרמיניסטיות

**חשוב להבין:** ב-MCP, הקליעין הוא לא באמת קליעין טהור והשרת הוא לא באמת שרת טהור – שניהם יכולים גם ליוזם וגם לקבל בקשות.

## 2.2 מחזור חיי החיבור

1. **שלב האתחול (Initialize):** פתיחת חיבור בין הקלינט לשרת
2. **שלב הפעולה (Operation):** הקלינט שולח בקשות, השירות מהזיר רשות יכולות וכליים
3. **שלב הסיום (Close):** סגירה מסודרת ושחרור משאבים

## 3 רכיבי ה프וטוקול

### 1.3 כלים (Tools)

כלים הם פונקציות שהמודל יכול להפעיל לביצוע פעולות או חישובים.  
**מבנה הגדרת כלים:**

- name – שם ייחודי לאיזהו
  - description – הסבר למודל מה הכליל עושה
  - input\_schema – הגדרת פרמטרים בפורמט JSON Schema
- דוגמה:** מחשבון – מקבל שני אופרנדים ופעולה, מהזיר תוצאה.

### 2.3 משאבי (Resources)

משאבים הם מקורות מידע שהמודל יכול לקרוא אך לא לשנות (Read-Only).  
**דוגמאות:**

- בסיסי נתונים
- דפי אינטרנט
- קבצי PDF
- תמונות

### 3.3 הנחיות (Prompts)

tabניות מוכנות להנחיית המודל לביצוע שימושות ספציפיות.

## 4 ארכיטקטורת הקלינט

### 1.4 שכבות הקלינט

1. **מודל השפה (LLM):** מקבל החלטות ויוזם פעולות

2. **משק הלקוח (Client Interface):** ה-API שהמודל עובד עליו

3. **LIBT המערךת:** מנהל סשנים ורישום כלים

4. **עיבוד הודעות:** המרה ל-JSON

5. **שכבת תקשורת:** תרגום ל-HTTP או STDIO

## 2.4 מודולים הכרחיים

### 1.2.4 מנהל הסשנים (Session Manager)

- מנהל מחזoor חיים של חיבורים
- מבצע Handshake – אימות שהחיבור הצליח
- מנהל Heartbeat – בדיקות דופק תקופתיות
- מגנון אוטומטי לחידוש קשר Retry Logic

### 2.2.4 רישום כלים (Tool Registry)

- מחזיק רשימה כלים זמינים מכל שירות
- מרכז את המידע לשימוש ה-LLM
- מטפל בהתנגשות שמות בין שירותים שונים

### 3.2.4 ניהול תור הודעות (Message Queue)

- מנהל תור של הודעות כניסה ויצואות
- מטפל בעדיפויות
- מונע הצפה

## 3.4 טיפול בשגיאות

### סוגי שגיאות:

- **Transient:** שגיאות זמניות (רשות, עומס) – שווה לניסות שוב
- **Permanent:** שגיאות קבועות (קובץ חסר, הרשאה) – אין טעם לניסות
- **Timeout:** חריגה בזמן – ניתן להגדיל את זמן המתנה

**אסטרטגיית Exponential Backoff:** זמן המתנה בין ניסיונות גדול באופן אקספוננציאלי:

- ניסיון 1: המתנה קצרה
- ניסיון 2: המתנה כפולה
- ניסיון 3: המתנה פי 4
- וכן הלאה...

**חשוב:** יש להוסיף Jitter (רעש אקראי) כדי למנוע מספר תהליכי שמתעוררים בו-זמנית.

## 5 עבودה עם מספר שירותים

### 1.5 טופולוגיה כוכב

קליענט אחד מרכז את התקשרות מול שירותי רבים. הקליענט:

- מציג ל-LLM רשימה מאוחדת של כלים
- מנהל מרחבי שמות למניעת התנגשות
- מנtab בקשות לשרת המתאים

### 2.5 איזון עומסים (Load Balancing)

אסטרטגיות נפוצות:

- Round Robin: חלוקה מעגלית שווה
- Least Connections: הפניה לשרת הפנוויבי יותר
- Weighted: עדיפות לשירותים חזקים יותר

## 6 אבטחת מידע

### 1.6 איומים נפוצים

התקפות Prompt Injection דרך תוכן באינטרנט:

- האקרים מכניםים הוראות זדוניות לדפי אינטרנט תמים
- ה-LLM קורא את התוכן ומבצע את ההוראות
- אין כיום פתרון מושלם לבעה זו

### 2.6 הגנות מומלצות

- עבודה בתחום Docker או סביבות וירטואליות (Sandbox)
- הגבלת הרשאות הגישה לתיקיות
- הגדרת גבולות נזק מראש
- תיעוד (Logging) מלא של כל הפעולות

## 7 מטלה: לигת סוכנים

### 1.7 תיאור המשימה

בנייה ליגה של סוכני AI שימושיים במשחק "זוג או פרד".

רכיבי המערכת:

1. **מנהל הליגה:** מגדרה מחזורי משחקים, מצוותת שחקנים, ניהול דירוג
2. **שופט:** מנהל משחק בודד בין שני שחקנים
3. **שחקנים:** ארבעה סוכנים שימושיים זה נגד זה

## 2.7 חוקי המשחק

- כל שחקן בוחר "זוגי" או "אי-זוגי"
- השופט מגיריל מספר
- אם שניהם צdkו או שניהם טעו – תיקו
- אם אחד צdk והשני טעה – המנצח הוא מי שצדק
- 03 שניות לתגובה, 3 ניסיונות לפני פסילה

## 3.7 עקרונות ארכיטקטוניים

1.  **הפרדת אחירות:** שכבת הליגה והשיפוט אינן תלויות במהלך המשחק הספציפי
2. **מודולריות:** ניתן להחליף את המשחק בקלות
3. **פרוטוקול אחד:** כל ההודעות ב-JSON עם מבנה קבוע

## 4.7 מבנה הודעה בפרוטוקול

כל הודעה חייבת לכלול:

- protocol – גרסה ה프וטוקול
- message\_type – סוג הודעה
- league\_id – מזהה הליגה
- round – מספר המഴור
- match\_id – מזהה המשחק
- timestamp – חותמת זמן

## 8 סיכום

### 1.8 נקודות מפתח

1. MCP מאפשר תקשורת אחת בין סוכני AI לשירותים חיצוניים
2. הקלינט הוא "המוח" (עム ה-LLM), השרת הוא "הידיים והרגליים"
3. יש להקפיד על ניהול שנים, קלים, והודעות
4. מנגנוני התאוששות ובטחה הם קרייטיים
5. הארכיטקטורה חייבת להיות מודולרית וסקלבילית

## **2.8 הכנה לפרויקט גמר**

המטרה הנוכחית היא הבסיס לפרויקט הגמר:

- בפרויקט הגמר – ליגה של כל הقيתה
- המשחק יהיה מורכב יותר (אולי "איקס עיגול גדול")
- הטעונים ישתמשו באסטרטגיות AI אמיתיות

---

כל הזכויות שמורות לד"ר יורם סגל (c)