

# Cloud Services, APIs and Big Data

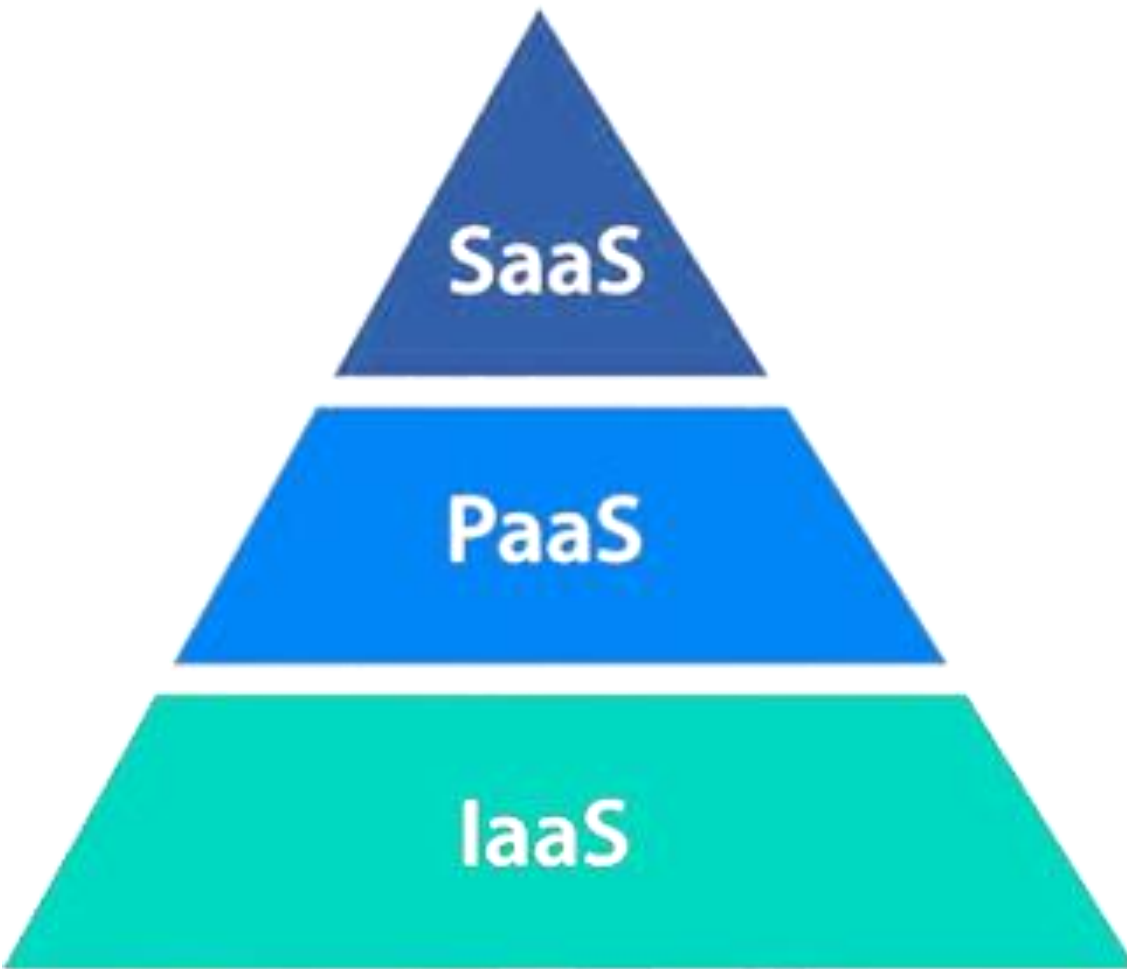
- **Cloud fundamentals**  
+ IaaS / PaaS / SaaS
- **Web Service / API**
- **Big Data– Spark**

- Cloud fundamentals + IaaS / PaaS / SaaS

## עקרונות מרכזיים של שירותי ענן:

- גישה מרחוק: דרך האינטרנט, מכל מקום
- תשלום לפי שימוש: כמו חשמל – רק כשמשתמשים
- גמישות והתאמה: אפשר לגדול או לקטון לפי צורך (scalability)
- חסכון בניהול תשתיות: לא צריך לקנות, לתחזק או להגן על החומרה

- Cloud fundamentals + IaaS / PaaS / SaaS



Software as a Service



Platform as a Service



Infrastructure as a Service

- Cloud fundamentals + IaaS / PaaS / SaaS



SaaS

### SaaS – Software as a Service – תוכנה כשירות

הספק מספק אפליקציה שלמה דרך האינטרנט. המשתמש צורך את השירות ללא צורך בהתקנה, ניהול או תחזוקה.

PaaS

### PaaS – Platform as a Service – פלטפורמה כשירות

























הספק מספק סביבה מוכנה להרצת קוד – כולל מערכת הפעלה, ספריות, סביבת פיתוח, ולעיתים גם בסיסי נתונים. המשתמש כותב ומעלה קוד בלבד.

IaaS

### IaaS – Infrastructure as a Service – תשתית כשירות

הספק מספק גישה לשרתים וירטואליים, אחסון, רשתות ומערכת הפעלה והמשתמש אחראי על ההתקנות, האפליקציות והנתונים.

- Cloud fundamentals + IaaS / PaaS / SaaS

SaaS	PaaS	IaaS	Traditional (On-Prem)	רכיב
 פק				קוד / אפליקציה
 פק	 פק			ספריות / פלטפורמות
 פק	 פק			מערכת הפעלה (OS)
 פק	 פק	 פק		אחסון (Storage)
 פק	 פק	 פק		שרתים (Servers)
 פק	 פק	 פק		תשתית פיזית (Networking)



## API (ממשק תכנות יישומים) Application Programming Interface

מערכת של כלים ופרוטוקולים המאפשרים לתוכנות שונות לתקשר זו עם זו. באמצעות API, מפתחים יכולים להשתמש בפונקציות ושירותים של תוכנות אחרות ללא צורך לכתוב את הקוד מחדש.

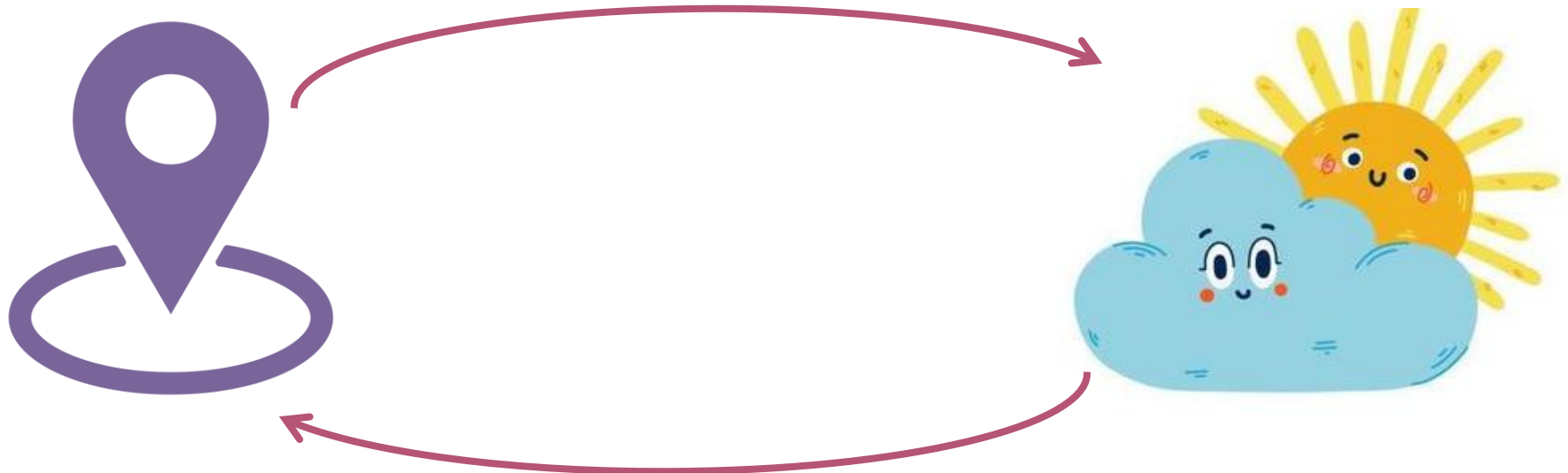
**Web Service (שירות רשת):** שירות המאפשר לתוכנות לתקשר זו עם זו דרך האינטרנט. Web Services משתמשים בפרוטוקולים סטנדרטיים כמו HTTP-XML כדי להעביר נתונים בין מערכות שונות

- API/Web Service

## API (ממשק תכנות יישומים) Application Programming Interface

מערכת של כלים ופרוטוקולים המאפשרים לתוכנות שונות לתקשר זו עם זו. באמצעות API, מפתחים יכולים להשתמש בפונקציות ושירותים של תוכנות אחרות ללא צורך לכתוב את הקוד מחדש.

**לדוגמא: אפליקציית מזג אוויר שולחת בקשת API לשרת של weather.com עם מיקום, ומקבלת טמפ' בחזרה.**





- API/Web Service

## -API Web Service

שירות המאפשר לתוכנות לתקשר זו עם זו דרך האינטרנט Web Services משתמשים בפרוטוקולים סטנדרטיים כמו HTTP ו-XML כדי להעביר נתונים בין מערכות שונות

שיטה (Method)	שימוש עיקרי	פעולה טיפוסית במערכת	מקביל ל-CRUD
GET	בקשת מידע מהשרת	שליפת פרטי משתמש / הצגת רשימה	Read
POST	שליחת מידע חדש לשרת	הרשמה / יצירת פוסט חדש	Create
PUT	עדכון מלא של מידע קיים	שינוי כל פרטי משתמש	Update
PATCH	עדכון חלקי של מידע קיים	שינוי רק כתובת אימייל	Update (חלקי)
DELETE	בקשה למחוק מידע מהשרת	מחיקת משתמש / פוסט	Delete

## Flask

**ספריית פייתון המאפשרת לבנות אפליקציות ווב פשוטות, כולל Web APIs בקלות ובמהירות.**

### מה Flask יודעת לעשות?

- לפתוח שרת ווב קטן על המחשב שלך
- להגדיר מסלולים – (routes) כלומר, כתובות שיפעילו פונקציות
- לשלוח ולהחזיר נתונים בפורמט JSON
- לעבוד עם טפסים, דאטה, בקשות, ועוד...

## מה זה Apache Spark?

- מנוע עיבוד נתונים **מבוזר** (Distributed Processing)
- נועד לעבודה עם **Big Data** נתונים שלא נכנסים לזיכרון
- מסוגל להריץ חישובים במקביל על אשכול מחשבים (או כמה ליבות)
- קוד פתוח – פרויקט של Apache Foundation

## מה זה PySpark?

- הממשק של Spark בשפת Python
- מאפשר לכתוב קוד Python רגיל שמנצל את מנוע העיבוד של Spark

## Lazy Evaluation הערכה עצלה

- Spark לא מריץ כלום עד שצריך תוצאה סופית (Action)
- מאפשר לו לבצע אופטימיזציה ולעבוד יעיל יותר

## Partitions מחיצות

- Spark מחלק את הדאטה ליחידות עבודה (Partitions)
- מריץ כל חלק במקביל (אם יש כמה ליבות/מכונות)

## SparkSession

- "מרכז הפיקוד" של אפליקציית Spark
- דרכו יוצרים DataFrame, טוענים קבצים ומפעילים את המנוע

## Transformations טרנספורמציות

- פעולות שמתארות מה לעשות על הנתונים (כמו filter, groupBy)
- **לא מתבצעות בפועל** – רק נבנה גרף של פעולות

## Actions פעולות סופיות

- גורמות ל Spark-להריץ את כל מה שתיארנו לפני כן
- דוגמאות show(), count(), collect()