

Cloud Services, APIs and Big Data



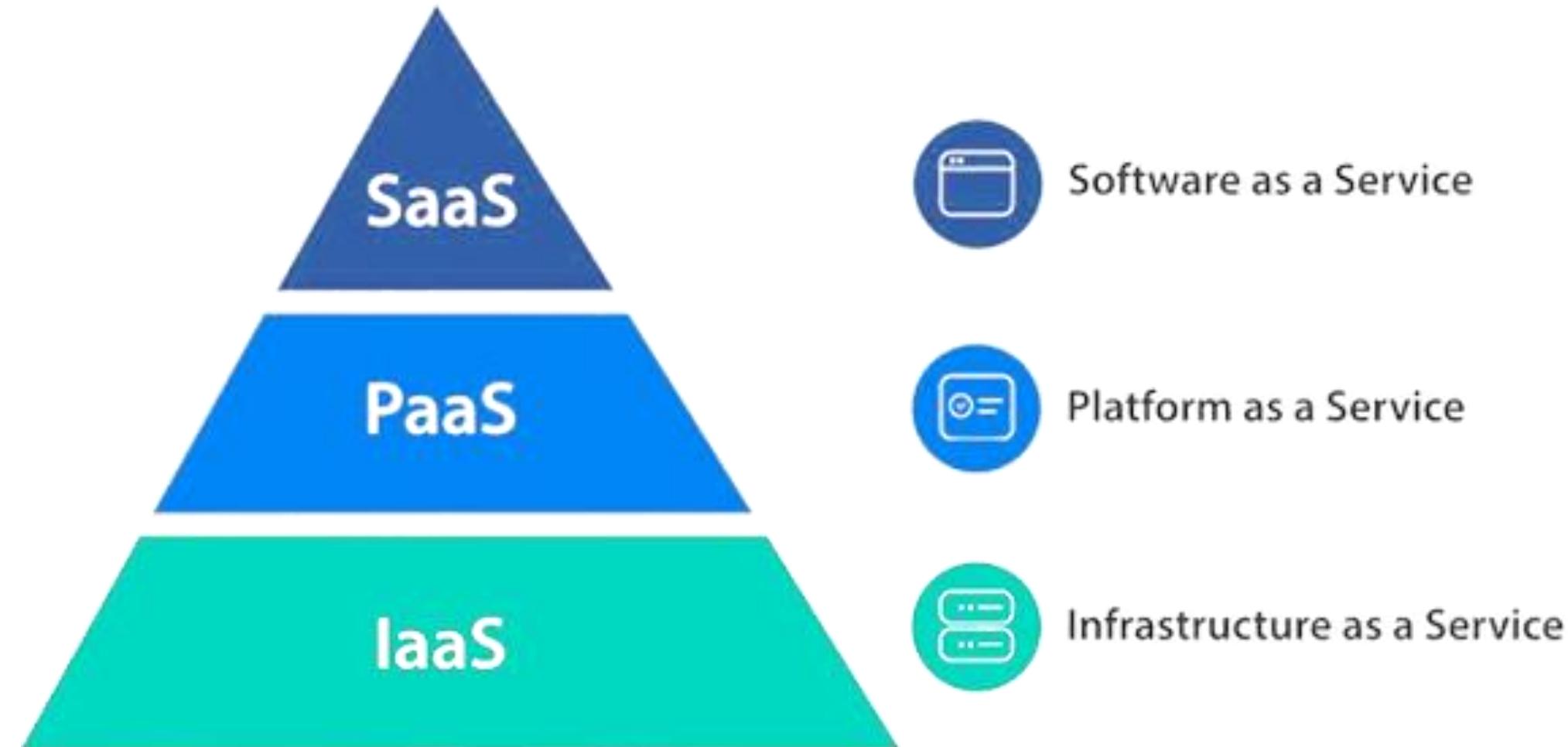
- Cloud fundamentals
+ IaaS / PaaS / SaaS
- Web Service / API
- Big Data- Spark



עקרונות מרכזיים של שירות ענן:

- **גישה מרוחק:** דרך האינטרנט, מכל מקום
- **תשלום לפי שימוש:** כמו חשמל – רק בשימושם
- **גמישות והתאמה:** אפשר לגודל או לקטן לפי צורך (scalability)
- **חסכון בניהול תשתיות:** לא צריך לקנות, לתחזק או להגן על החומרה

- Cloud fundamentals + IaaS / PaaS / SaaS



- Cloud fundamentals + IaaS / PaaS / SaaS



SaaS – Software as a Service

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

PaaS – Platform as a Service

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

IaaS – Infrastructure as a Service

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

- Cloud fundamentals + IaaS / PaaS / SaaS

SaaS	PaaS	IaaS	Traditional (On-Prem)	רכיב
0פְּקָד	0פְּקָד	0פְּקָד	0פְּקָד	קוד / אפליקציה
0פְּקָד	0פְּקָד	0פְּקָד	0פְּקָד	ספריות / פלטפורמות
0פְּקָד	0פְּקָד	0פְּקָד	0פְּקָד	מערכת הפעלה (OS)
0פְּקָד	0פְּקָד	0פְּקָד	0פְּקָד	אחסון (Storage)
0פְּקָד	0פְּקָד	0פְּקָד	0פְּקָד	שרתים (Servers)
0פְּקָד	0פְּקָד	0פְּקָד	0פְּקָד	תשתיות פיזית (Networking)

- API/Web Service

API (ממשק תכנתות יישומים)

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

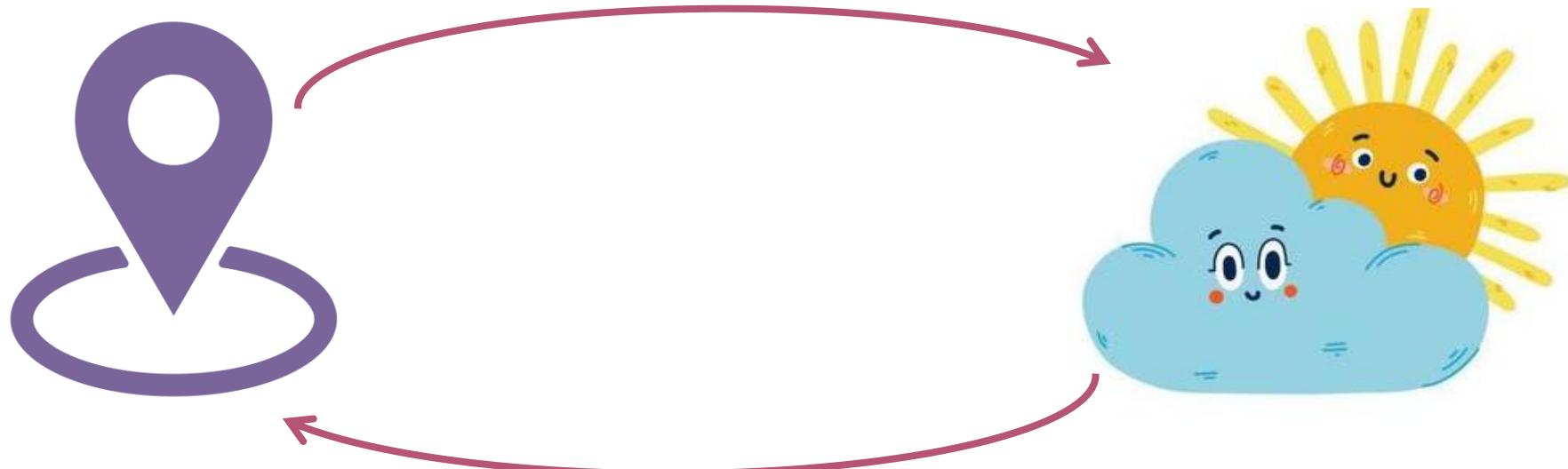
Web Service (שירות רשת): שירות המאפשר לתוכנות לתקשר זו עם זו דרך האינטרנט. Web Services משתמשים בפרוטוקולים סטנדרטיים כמו HTTP-XML כדי להעביר נתונים בין מערכות שונות.

- API/Web Service

API (ממשק תכונות 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 לבתור את הקוד מחדש.

לדוגמה: אפליקציית מזג אוויר שולחת בקשה API לשירות של weather.com עם מקום, ומתקבלת טמפרטורה 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

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

- API/Web Service

Flask

ספרית פיתון המאפשרת לבנות **אפליקציות ווֹב פשוטות**, בולל, **Web APIs** בקלות ובמהירות.

מה Flask יודעת לעשות?

- **לפתח שרת ווֹב קטן על המחשב שלך**
- **להגדיר מסלולים – (routes)** בלוּמֶר, כתובות שייפעלו פונקציות
- **לשЛОח ולהחזיר נתונים בפורמט JSON**
- **לעבוד עם טפסים, DATA, בקשנות, ועוד...**

- Big Data – Spark

מה זה? Apache Spark

- מנוע עיבוד נתוניים **מבודר** (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



- Big Data – Spark

הערכת עצלה Lazy Evaluation

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

מחיצות Partitions

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

- Big Data – Spark

SparkSession

- "מרכז הפיקוד" של אפליקציית Spark
- דרך יוצרים `DataFrame`, מעריכים קבצים ומבצעים את המנווע

Transformations

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

פעולות סופיות Actions

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