הנדסת פתרונות תוכנה מתקדמים חלק א'-EASS

אתר הקורס:
https://github.com/EASS-HIT-2022/Part-A-Agenda/tree/main/lectures

ערוץ discord:

https://discord.gg/DcaN8JEn

Lecture 2

מבנה הציון

- Build full REST/HTTP fastapi backend + Dockerization **(due 1/4)**
- 4 modules on AWS course (S3, EC2, EBS, RDS) - if you finish all the course you get +10 bonus points to final grade**(due 30/4)**
- UI (react/streamlit) **(due 1/5)**
- Docker compose the server with UI and backend plus server and write a clear README with git submodules **(due 29/5)**
- Presentation of the system in a demo in a 2-3 minutes video on youtube and clear README **(due 29/5)**

רשימת הנושאים שיילמדו בקורס

- Monolithic vs. Microservices
- Docker
- Client-Server
- REST/HTTP API
- FastAPI
- Pytest
- asyncio
- Frontend (React javascript and Streamlit python)
- Docker compose
- Functional programming
- How to compile a new library

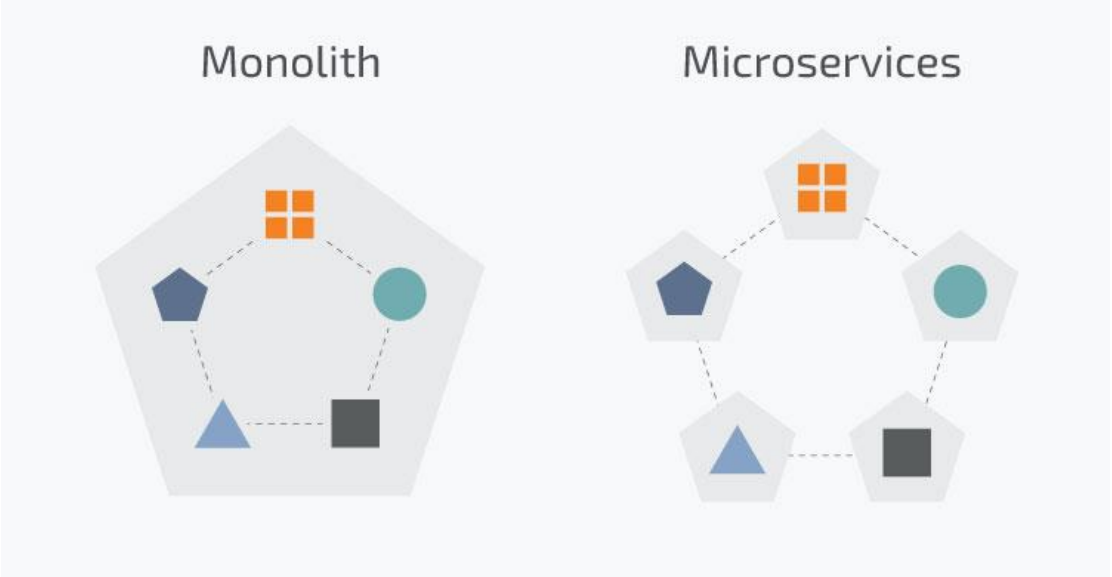
עקרונות הנדסת תוכנה

בכל מערכת נשתדל לשמור על כמה עקרונות בתהליך הפיתוח:

- תכנון מקדים- Measure twice and cut once
- לא לחזור על קוד- Don’t Repeat Yourself (DRY)
- פשטות- keep It Simple -Stupid- (KISS)
- להימנע מאופטימיזציה בשלב מוקדם מדי- Avoid Premature Optimization
- קוד קריא ופשוט(בלי וואסח והתחכמויות מיותרות)
- היררכיה הגיונית של מבנה התיקיות
- S.O.L.I.D
  1. Single responsibility- אחריות אחת למחלקה אחת.
  2. Open-closed- פתוח להרחבה, סגור לשינויים.

3. Liskov substitution- כל מחלקה יורשת, צריכה להמשיך את הייעוד של המחלקה הבסיסית.
4. Interface segregation- לקוח לא צריך להיות תלוי בדברים שהוא לא משתמש בהם.
5. Dependency inversion-המתכנת צריך לעבוד ברמת הממשק, ולא ברמת המימוש.

Monolithic vs. Microservices



נשבור את הארכיטקטורה לחתיכות קטנות יותר.

בתפיסת המיקרוסרוויס, כל חלקיק עומד בפני עצמו-וכולם ביחד על גבי פרוטוקול HTTPS מתאחדים לטובת מערכת אחת.

נעבור כעת לתרגול פקודות ב-docker:

- Dockerhub/Registry
- Dockerfile
- docker build
- docker run
- docker ps
- docker network ls
- docker volumes
- docker expose ports
- docker images
- docker exec
- docker image prune -a