

Adatfolyam alapú szoftverfejlesztés beágyazott rendszerekben

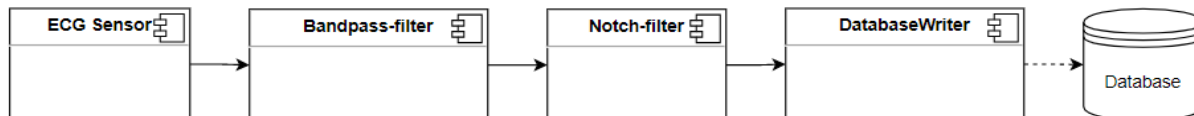
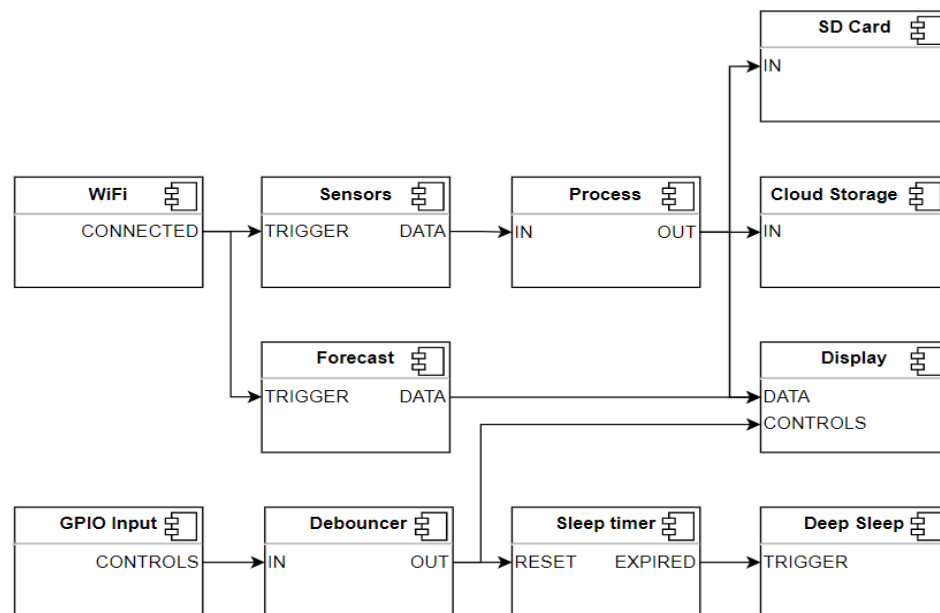
Gyulai Péter
Konzulens: Krébesz Tamás István



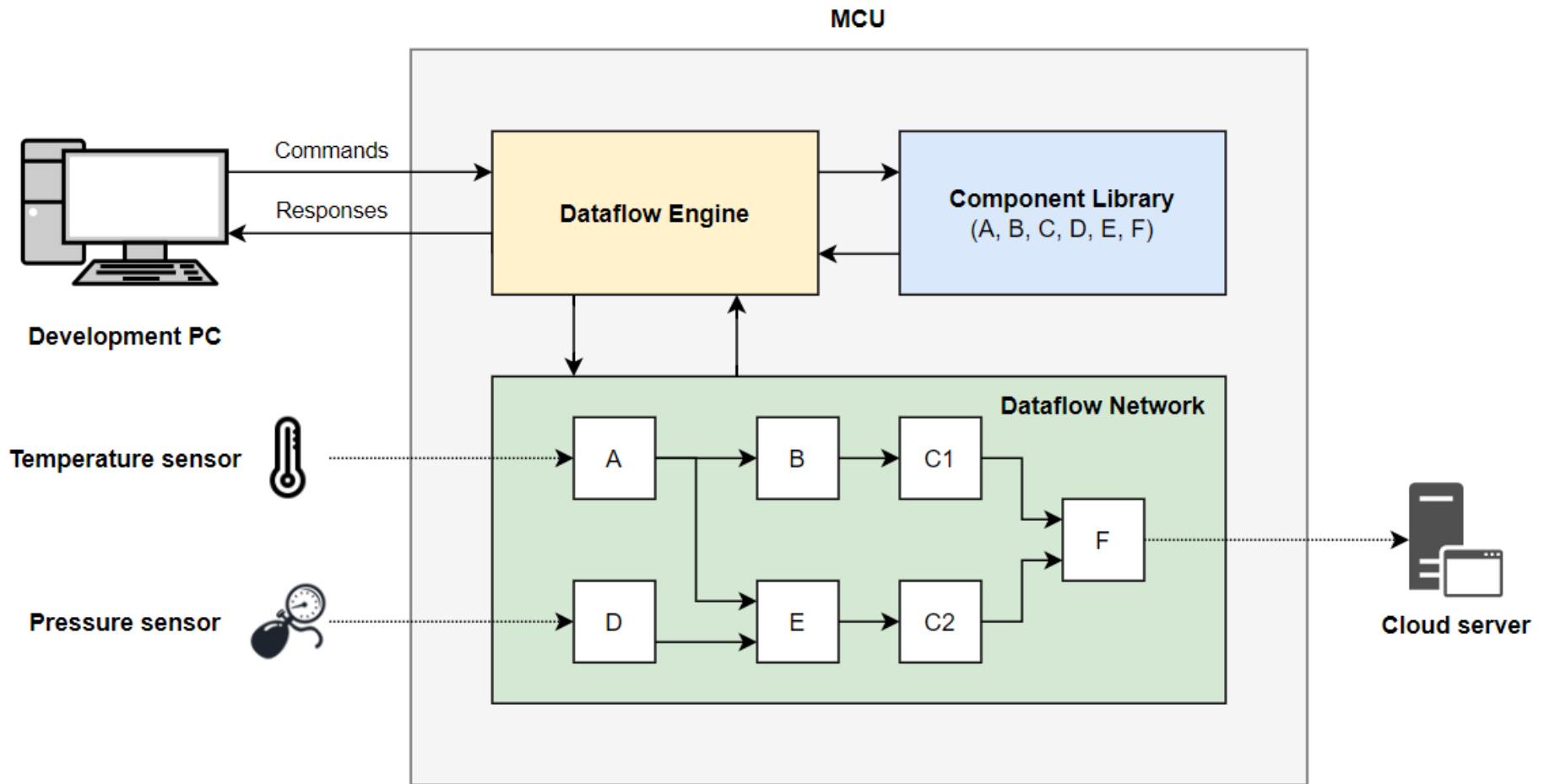
Méréstechnika és
Információs Rendszerek
Tanszék

Adatfolyam alapú programozás

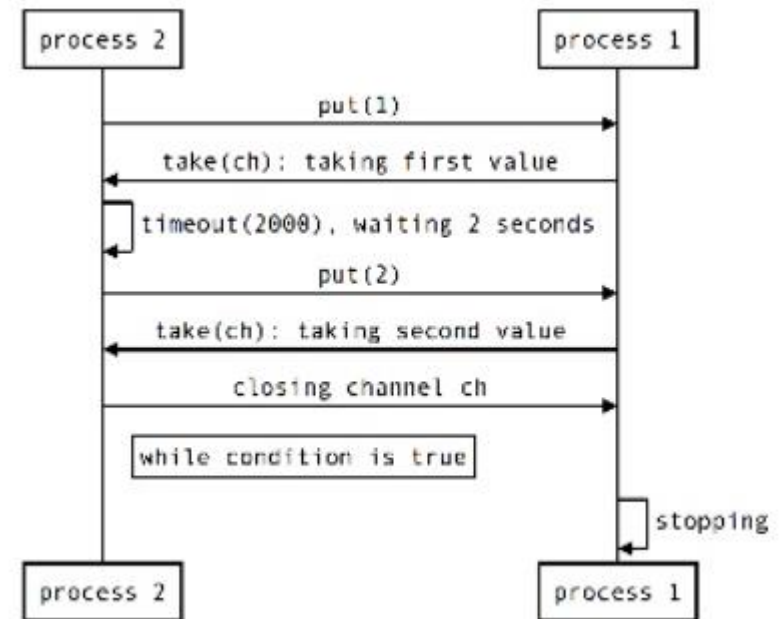
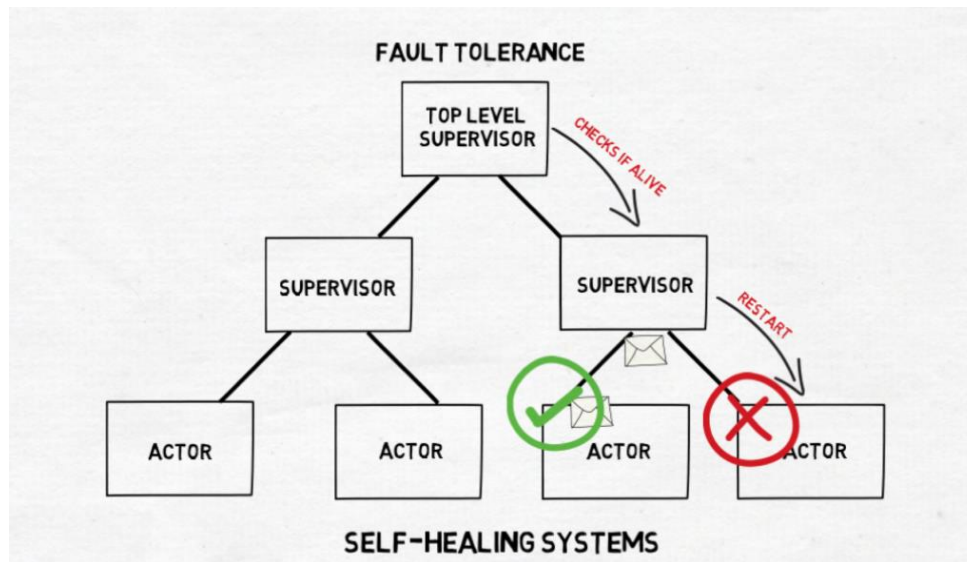
- Imperatív vs. Deklaratív
- *Mit? vs. Hogyan?*
- Komponens-alapú fejlesztés
- Újrafelhasználhatóság
- Párhuzamos végrehajtás
- Szinkronizáció



Az elképzelés

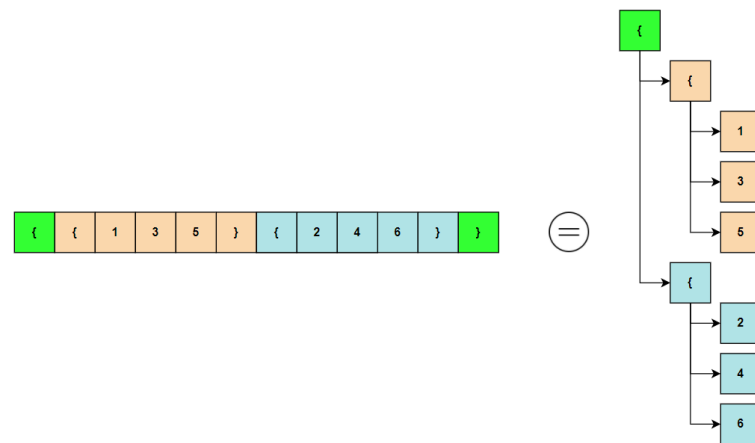


- Actor modell
- Communicating Sequential Processes (CSP)
- **Flow-based Programming (FBP)**



Folyamalapú programozás

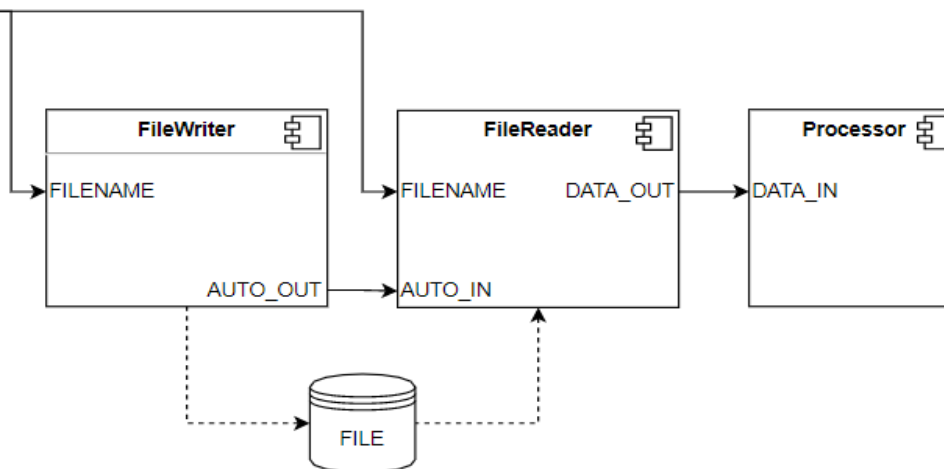
- Párhuzamosan futó komponensek
- FIFO üzenetsorok
- Kimeneti és bemeneti portok
- Strukturált üzenetformátum
- Szinkronizációs portok



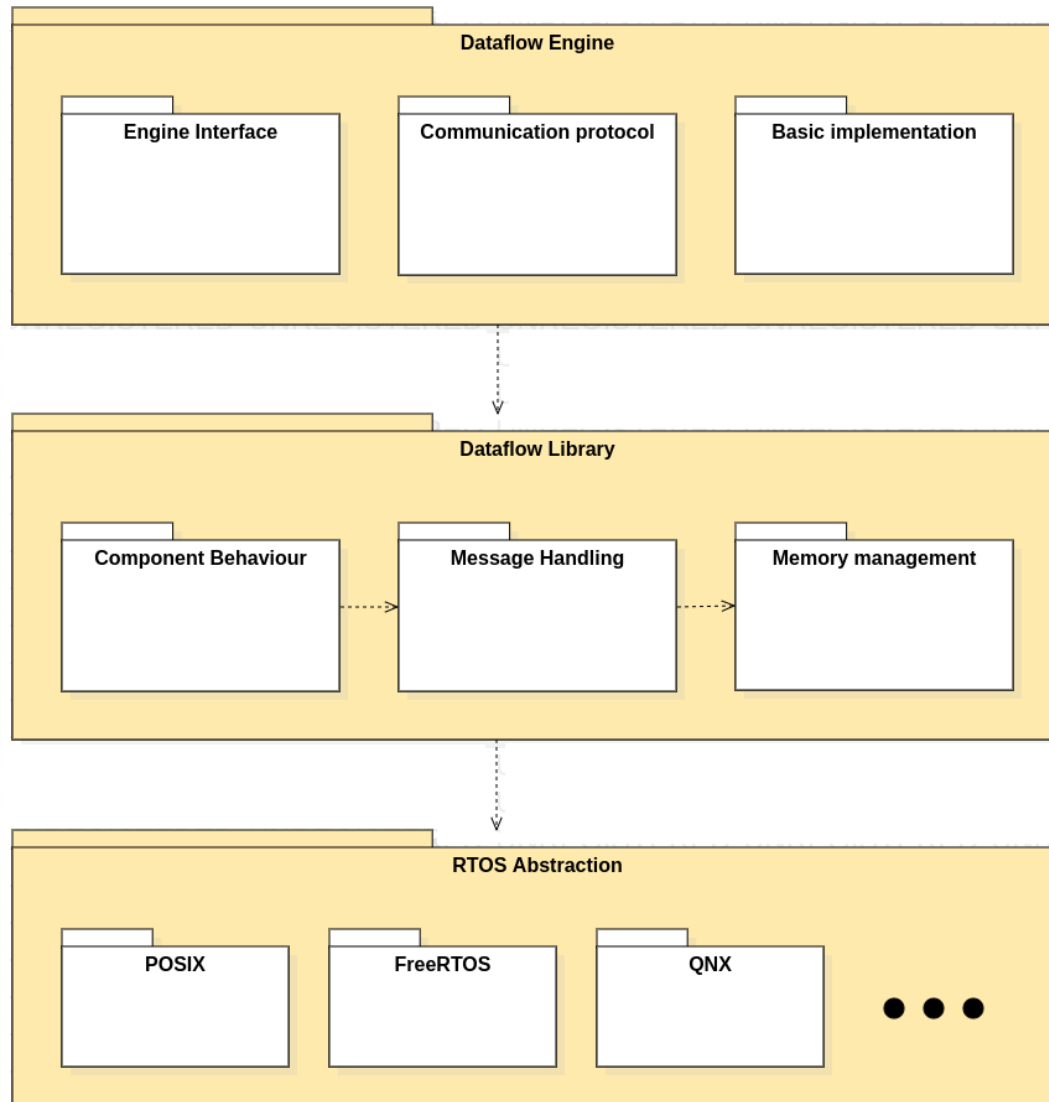
"data.txt" (IIP) 

Fájlműveletek szinkronizációja.

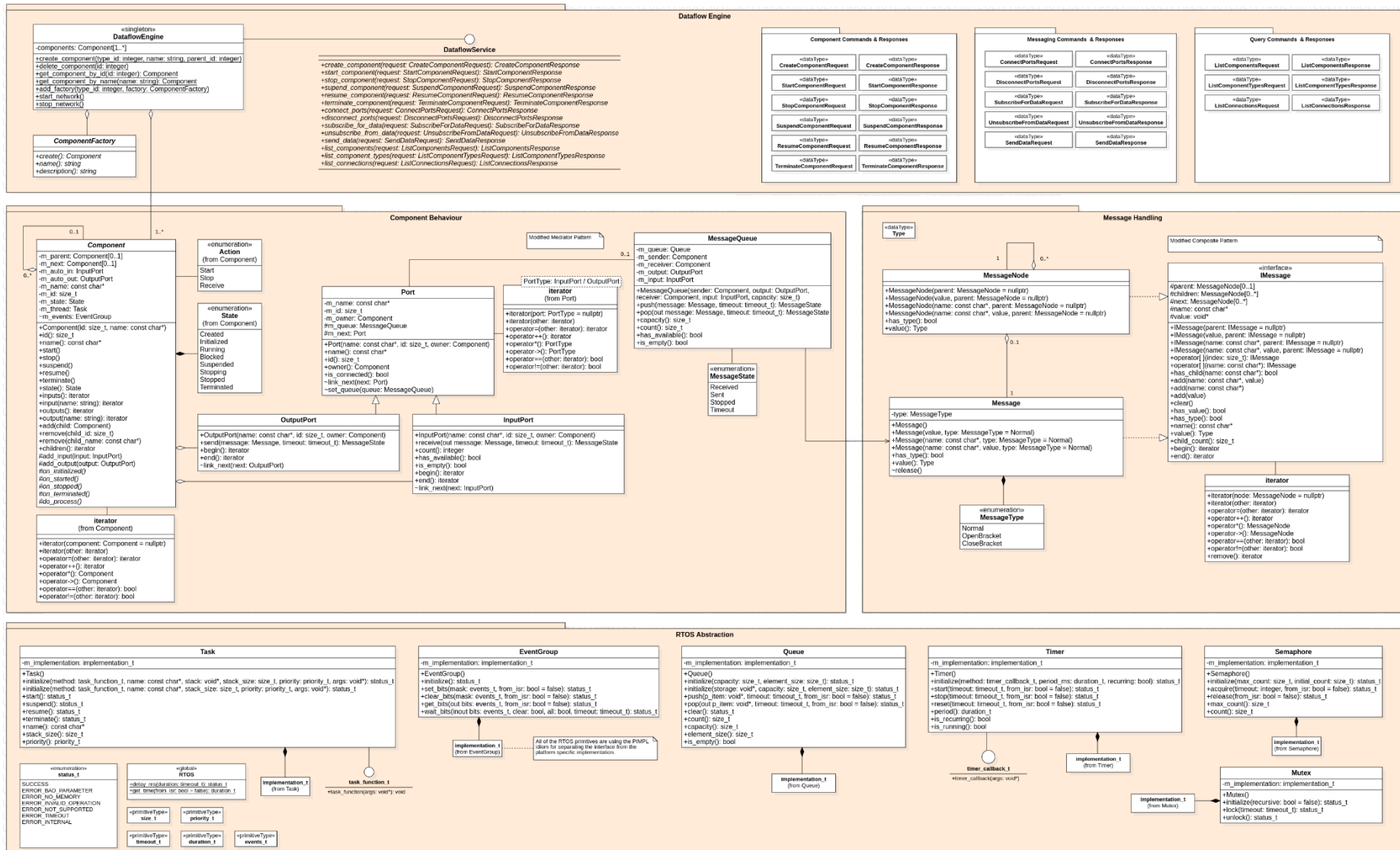
Az olvasó komponens nem aktiválódhat amíg az író komponens be nem fejezte a fájl teljes tartalmának írását. Ezt az automatikus portok összeköttetése biztosítja. A FileReader és a Processor komponens között nincs explicit szinkronizáció, utóbbi akkor kezdi meg a végrehajtást, amikor a bemenetén rendelkezésre áll az adat.



A keretrendszer felépítése

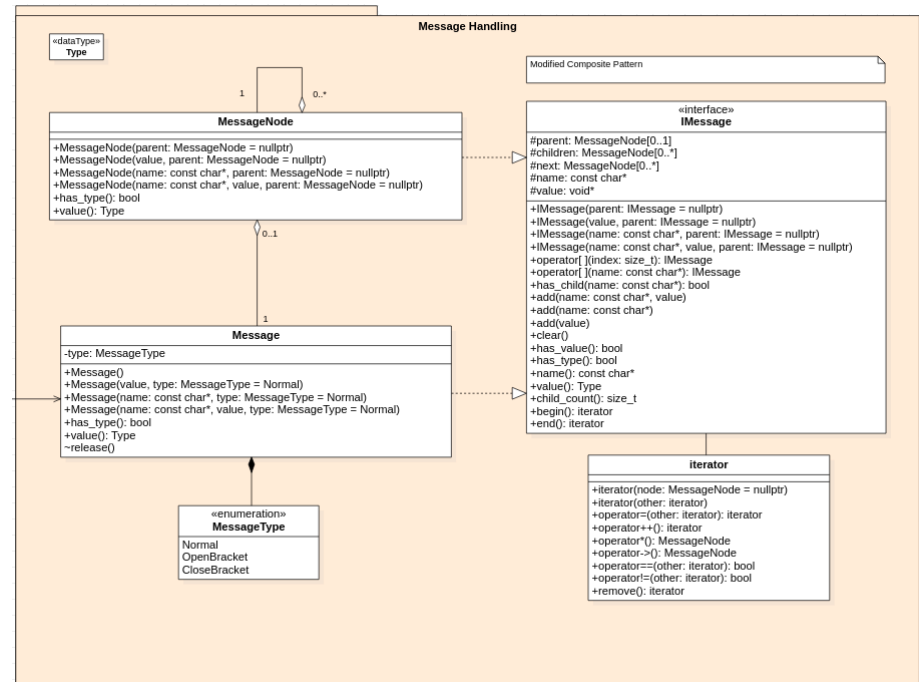


Osztálydiagram

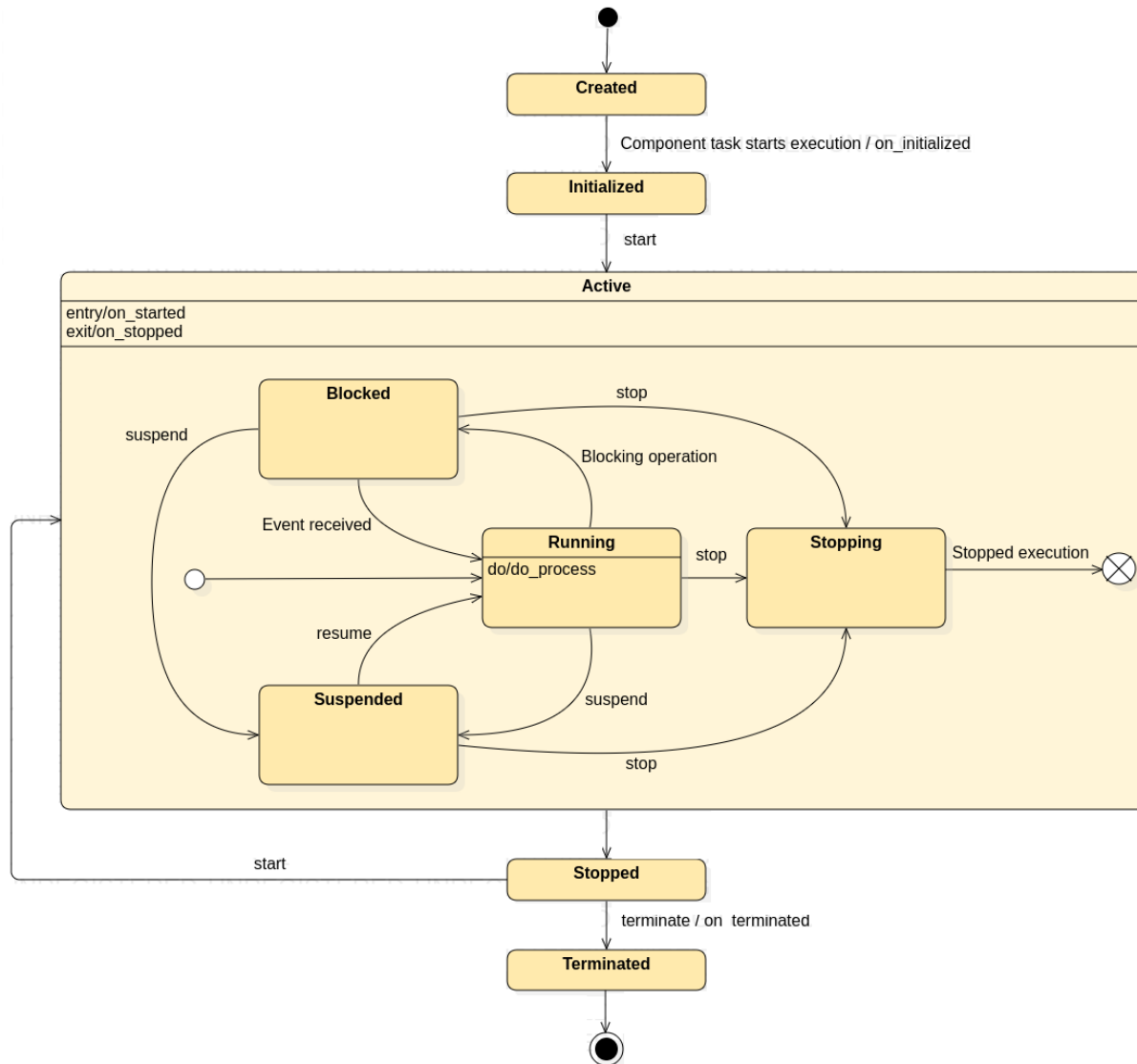


Üzenetkezelési stratégiák

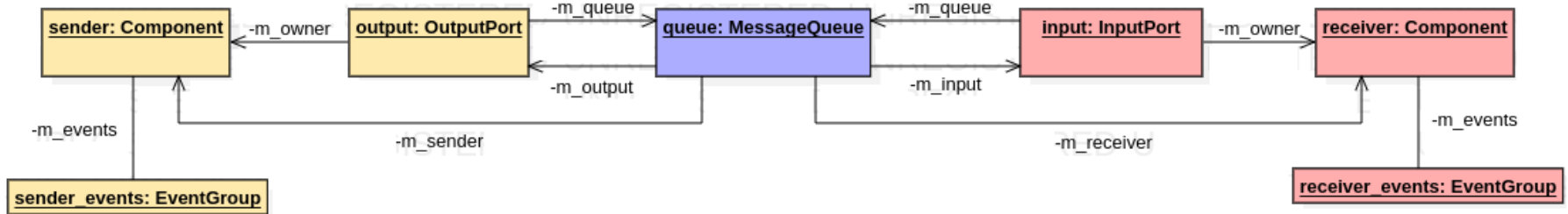
- Hova tegyük az üzeneteket?
- Statikus vs. Dinamikus memória
- Hogyan ellenőrizzük az üzenetek típusát?
- Statikus típusazonosító, virtuális tábla, dinamikus típuskonverzió



Komponensek életciklusa

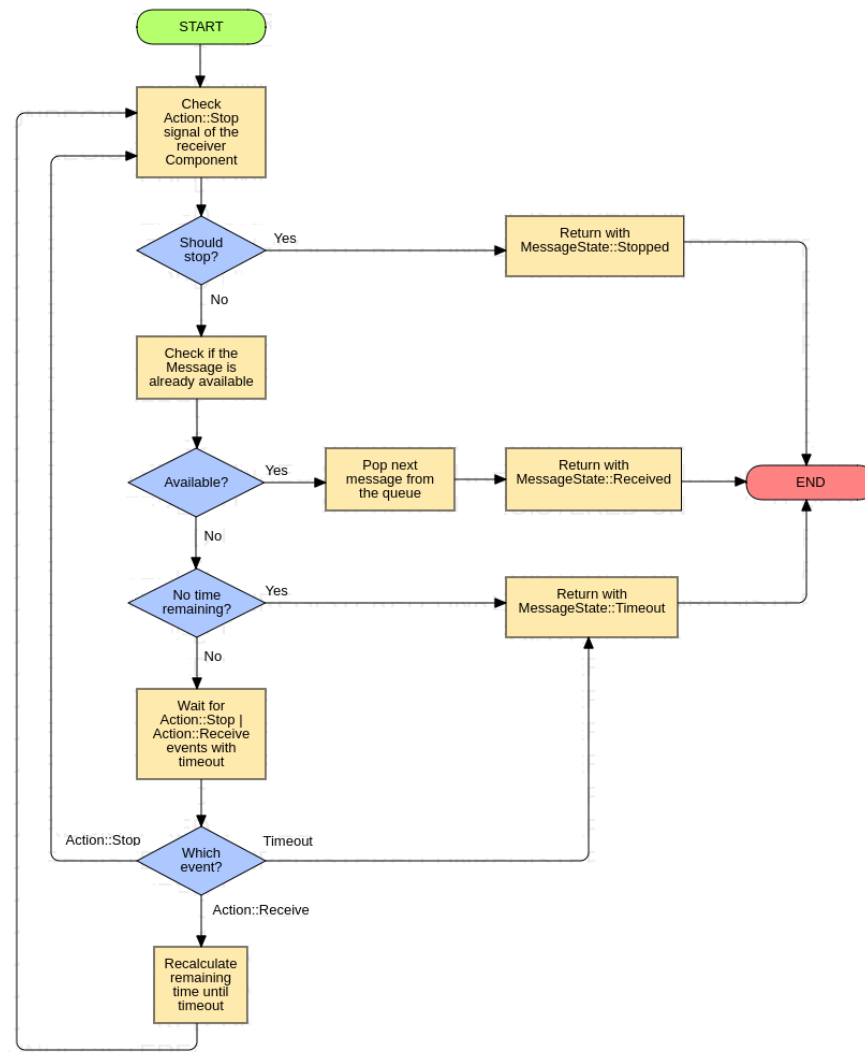
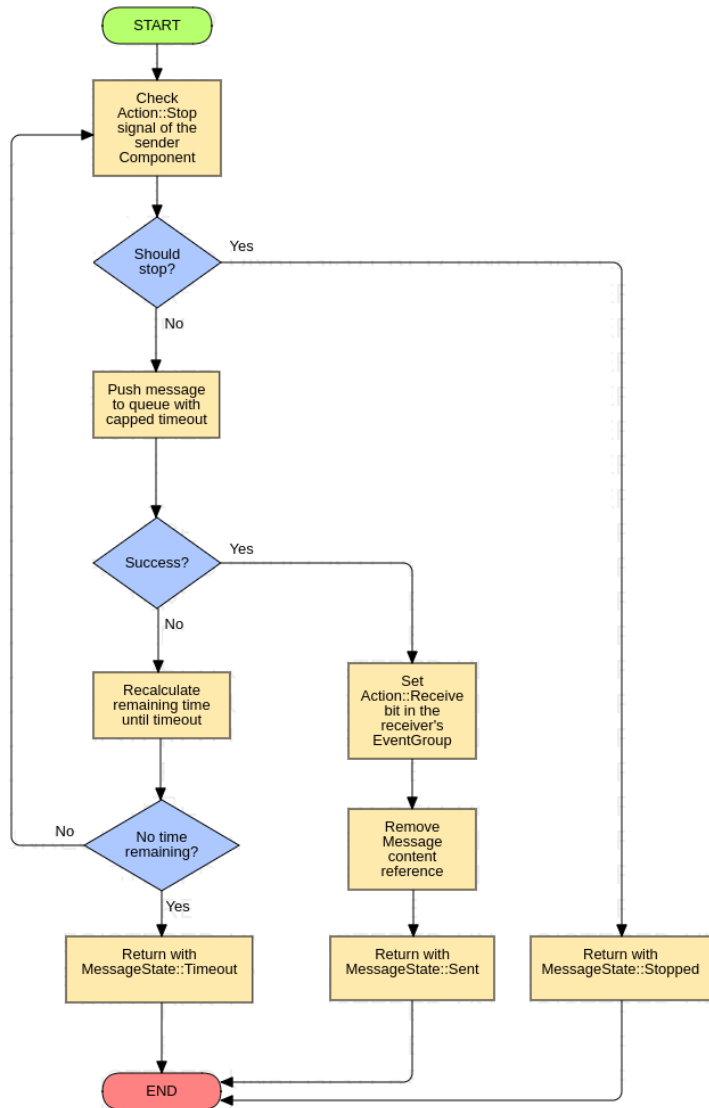


A kommunikáció objektumdiagramja



- Mediátor tervezési minta
- Laza csatolás a portok között
- Eseményvezérelt kommunikáció

Küldés és fogadás folyamatábrája



Implementáció

- Fejlesztőkörnyezet (VSCode, Docker, tools)
- Build-rendszer felélesztése (CMake)
- OS absztrakció megvalósítása – POSIX
- Unit-tesztek (CTest, Unity)

Köszönöm a figyelmet!

- Kérdések?