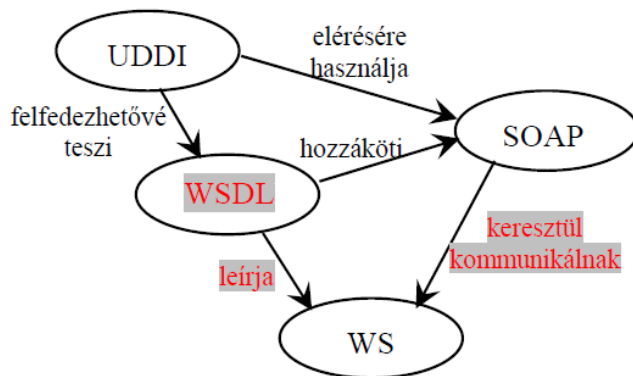


# Software architecture – Megoldások

---

### 2008.06.17 – 3. Feladat

A szolgáltatás-orientált architektúrát bemutató ábrán mik a név nélküli elemek?



### 2009.01.27 – 10. Feladat

Az üzleti folyamatok lefutásának két esete az (1) orkesztráció (orchestration) és a (2) koreográfia (choreography). Mi a két megközelítés közötti különbség?

- **orkesztráció (orchestration)** – Egy központ ismeri a teljes folyamatot, az kér szolgáltatást az együttműködőktől, akik csak a saját dolgukat végzik.
- **koreográfia (choreography)** – A folyamat nincs központosítva, minden résztvevő a dolgát elvégezve az általa ismert következő résztvevő(ke)t aktivizálja.

### 2010.01.12 (A) – 4. Feladat

Az esemény alapú implicit hívást használó (event-based implicit invocation) architektúrának mi a két fő komponense és mi a komponensek szerepe?

- **komponensek:**
  - eseményforrás,
  - eseményfigyelő
- **szerepek:**
  - az eseményfigyelők beregisztrálják magukat a forrásoknál,
  - akik a náluk keletkező eseményt kiküldik valamennyi beregisztrált komponensnek.

### 2010.01.12 (A) – 6. Feladat

Az üzleti folyamatok lefutásának két esete az (1) orkesztráció (orchestration) és a (2) koreográfia (choreography). Mi a két megközelítés közötti különbség?

- **orkesztráció (orchestration)** – Egy központ ismeri a teljes folyamatot, az kér szolgáltatást az együttműködőktől, akik csak a saját dolgukat végzik.
- **koreográfia (choreography)** – A folyamat nincs központosítva, minden résztvevő a dolgát elvégezve az általa ismert következő résztvevő(ke)t aktivizálja.

### 2010.05.26 – 6. Feladat

Az esemény alapú implicit hívást használó (event-based implicit invocation) architektúrának mi a két fő komponense és mi a komponensek szerepe?

- **komponensek:**
  - eseményforrás,
  - eseményfigyelő
- **szerepek:**
  - az eseményfigyelők beregisztrálják magukat a forrásoknál,
  - akik a náluk keletkező eseményt kiküldik valamennyi beregisztrált komponensnek.

### 2010.12.21 – 4. Feladat

Egy egyszerű társalgó (chat) program úgy működik, hogy a bejelentkezett felhasználó által küldött üzeneteket a rendszer minden másik bejelentkezett felhasználónak továbbítja. Milyen a szoftver architektúrája?

- Event based implicit invocation (eseményvezérelt).

### 2011.06.07 – 4. Feladat

Az esemény alapú implicit hívást használó (event-based implicit invocation) architektúrának mi a két fő komponense és mi a komponensek szerepe?

- **komponensek:**
  - eseményforrás,
  - eseményfigyelő
- **szerepek:**
  - az eseményfigyelők beregisztrálják magukat a forrásoknál,
  - akik a náluk keletkező eseményt kiküldik valamennyi beregisztrált komponensnek.

### 2011.12.20 – 5. Feladat

Az interpreter architektúra melyik komponensének feleltethető meg egy állapottábla?

- Pseudocode.

### 2012.01.03 – 4. Feladat

Az alábbi táblázatban a fontosabb szoftver architektúra típusokat adtuk meg, minden típushoz egy indexet rendelve. A jobboldali táblázatban szoftver architektúrákkal kapcsolatos fogalmak állnak. Adja meg, hogy az egyes fogalmak melyik architektúra típushoz kapcsolhatóak leginkább! (A fogalom előtti mezőbe írja be a típus indexét!) Ha a fogalom egyik típushoz sem társítható, akkor a mezőbe tegyen X-et!

1	esemény alapú implicit hívást használó (event-based implicit invocation)
2	csővezeték+szűrők (pipes and filters)
3	tábla (blackboard)
4	interpreter
5	szolgáltatás orientált architektúra (SOA)

2	batch (kötegelés)
4	virtuális gép (virtual machine)
1	regisztráció (registration)
3	központi adatbázis (repository)
5	kötés (binding)
1	szórás (broadcast)

### 2013.01.08 – 5. Feladat

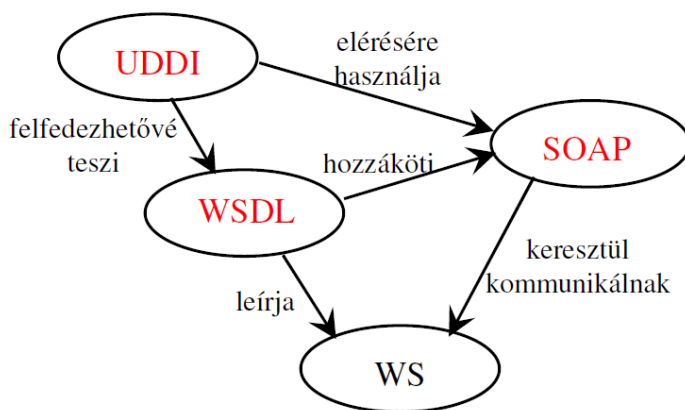
Az alábbi táblázatban a fontosabb szoftver architektúra típusokat adtuk meg, minden típushoz egy indexet rendelve. A jobboldali táblázatban szoftver architektúrákkal kapcsolatos fogalmak állnak. Adja meg, hogy az egyes fogalmak melyik architektúra típushoz kapcsolhatóak leginkább! (A fogalom előtti mezőbe írja be a típus indexét!) Ha a fogalom egyik típushoz sem társítható, akkor a mezőbe tegyen X-et!

1	esemény alapú implicit hívást használó (event-based implicit invocation)
2	csővezeték+szűrők (pipes and filters)
3	tábla (blackboard)
4	interpreter
5	szolgáltatás orientált architektúra (SOA)

5	XML
1	szórás (broadcast)
2	batch (kötegelés)
2	adatfolyam (data flow)
1	előfizet (subscribe)
X	grafikus interfész (GUI)

### 2013.01.08 – 10. Feladat

A szolgáltatás orientált architektúrát bemutató ábrán mik a név nélküli elemek?



### 2013.05.28 – 10. Feladat

Az üzleti folyamatok lefutásának egyik esete az orkesztráció (orchestration). Mi a lényege?

- Egy központ ismeri a teljes folyamatot, az kér szolgáltatást az együttműködőktől, akik csak a saját dolgukat végzik.

### 2014.01.14 – 7. Feladat

Az esemény alapú implicit hívást használó (event-based implicit invocation) architektúrának mi a két fő komponense és mi a komponensek szerepe?

- **komponensek:**
  - eseményforrás,
  - eseményfigyelő
- **szerepek:**
  - az eseményfigyelők beregisztrálják magukat a forrásoknál,
  - akik a náluk keletkező eseményt kiküldik valamennyi beregisztrált komponensnek.

## 2015.01.20 – 2. Feladat

Az alábbi táblázatban a fontosabb szoftver architektúra típusokat adtuk meg, minden típushoz egy indexet rendelve. A jobboldali táblázatban szoftver architektúrákkal kapcsolatos fogalmak állnak. Adja meg, hogy az egyes fogalmak melyik architektúra típushoz kapcsolhatóak leginkább! (A fogalom előtti mezőbe írja be a típus indexét!) Ha a fogalom egyik típushoz sem társítható egyértelműen, akkor a mezőbe tegyen X-et! (3 pont)

1	csővezeték+szűrők (pipes and filters)
2	szolgáltatás orientált architektúra (SOA)
3	rétegelt (layered)
4	tábla (blackboard)
5	esemény alapú implicit hívást használó (event-based implicit invocation)
6	interpreter

<b>3</b>	grafikus interfész (GUI)
<b>5</b>	regisztráció (registration)
<b>2</b>	kötés (binding)
<b>X</b>	egységbezárás (encapsulation)
<b>6</b>	pszeudokód (pseudocode)
<b>2</b>	koreográfia (choreography)

asd

asd

asd