

Használati alapú és modell alapú  
tesztelés kombinálása  
szolgáltatásorientált architektúrák  
teszteléséhez az ipari gyakorlatban

*Nagy Attila Mátyás*

2016.12.07.

# Áttekintés

- Bevezetés
- Megközelítés
- Pilot tanulmányok
- Konklúzió

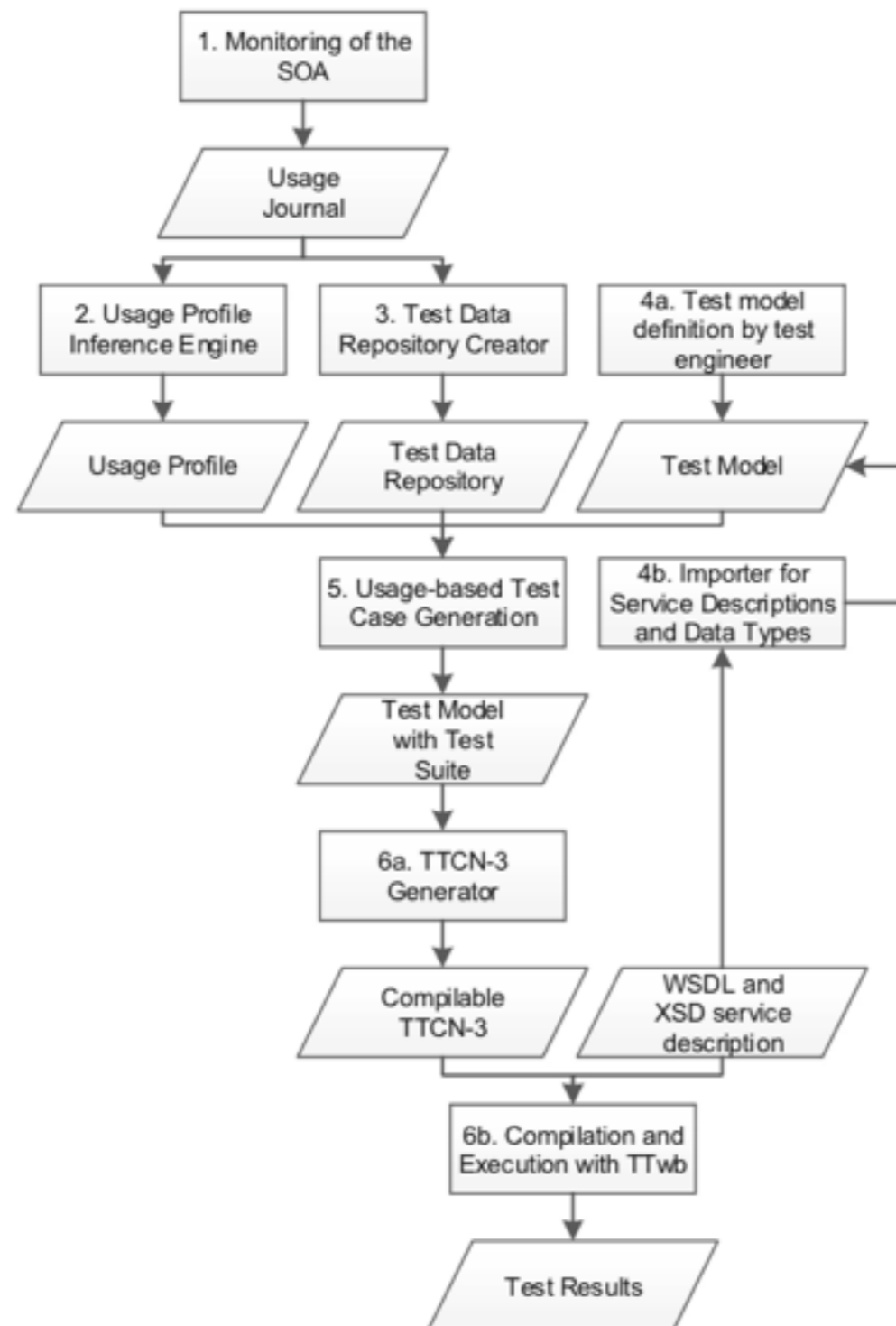
# Bevezetés

- Minőség biztosítása az elérhető erőforrások legjobb kihasználásával
- Lehetséges stratégia: user-experienced minőség a középpontban
- Használati alapú tesztelés: A tesztelendő rendszer(SUT) gyakran használt komponenseire kell fókuszálni  
—> a core funkciókat kell részletesen tesztelni

# Bevezetés

- Használat-alapú teszteléshez usage-profile-ok szükségesek
- Jelenleg ezek a usage-profile-ok manuálisan VAGY automatikusan generáltak —> sok esetben nehézkes automatizálás
- Új megközelítés:
  - SOA-k tesztelésére koncentráltak
  - Cél: minél jobban automatizálni a tesztelést
  - használati adatok alapján profilok készítése
  - Teszt modell készítése modell alapú megközelítésből
  - Ezek kombinálása futtatható teszteké

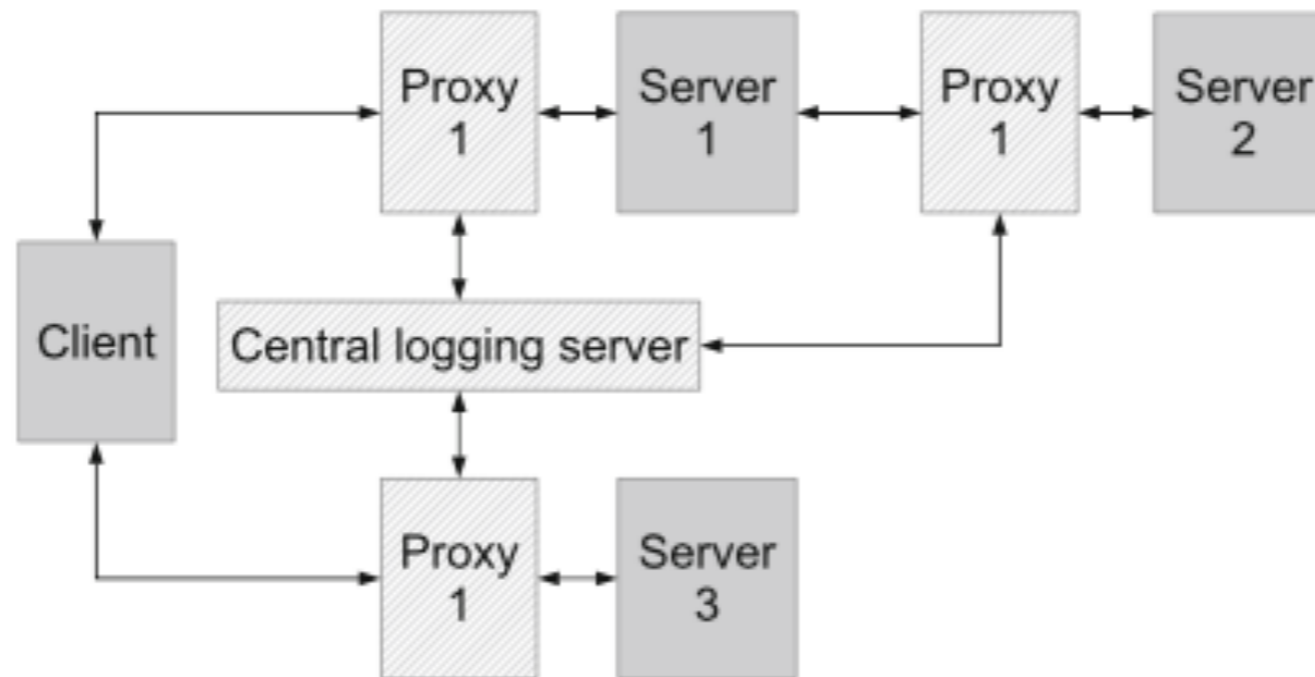
# Megközelítés



# SOA monitorozása

- SOA-k különböző meghívható szolgáltatásokat biztosítanak (kliens-szerver típusú hívások)
- Kérések-válaszok monitorozása
  - ◉ forrás/cél tárolása
  - ◉ Paraméterek értékei
  - ◉ Visszatérési értékek
- Monitorozáshoz proxy-k használata

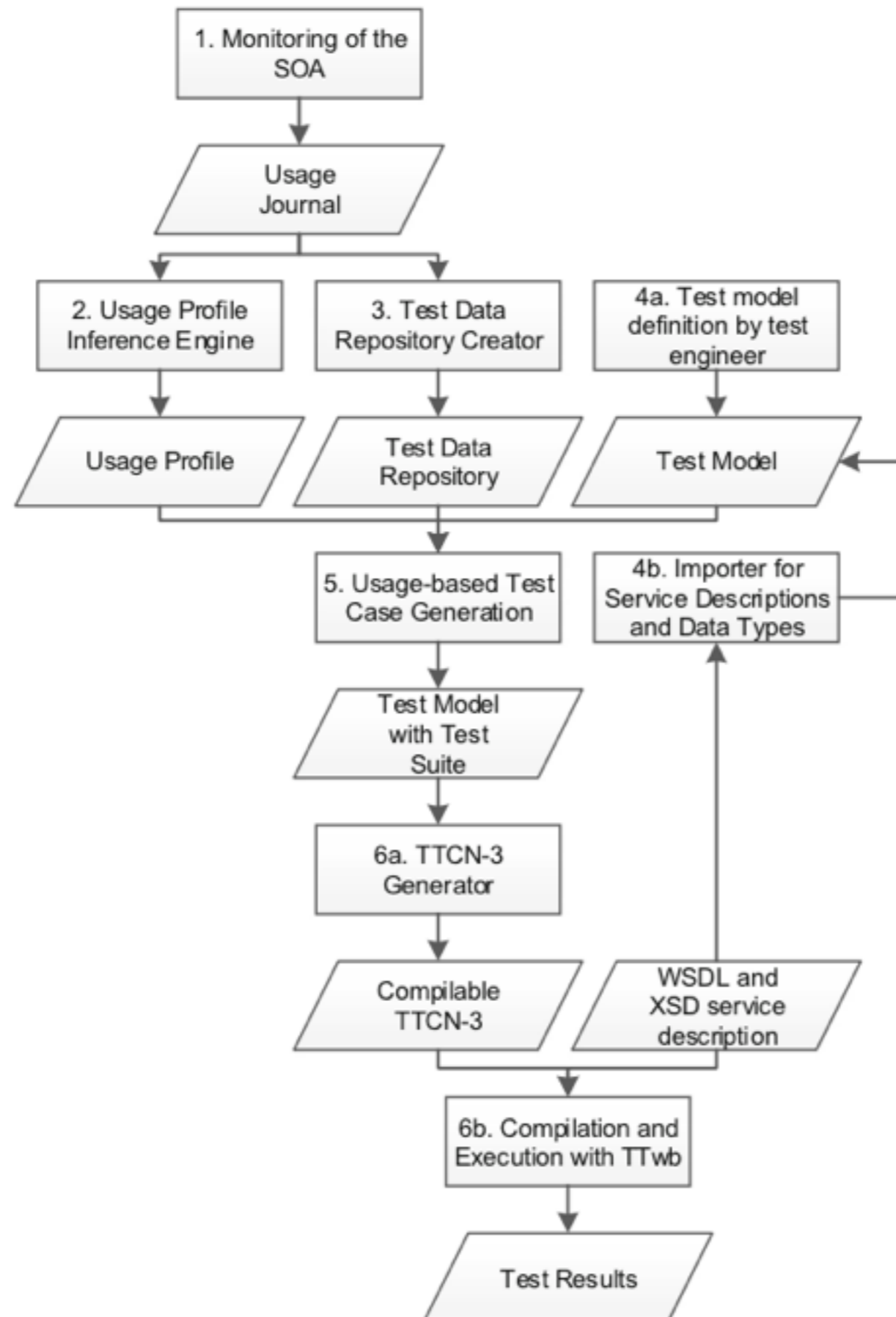
# SOA monitorozása



# SOA monitorozása

- Log file-ok tárolása egy központi szerveren
- Több proxy ugyanahhoz a szerverhez a kliensek pontos azonosításához
- Csak SOA-k vizsgálata HTTP feletti SOAP üzenetek megfigyelésével





# Usage profile-ok készítése

- Bemenete a monitorozás során felvett usage data
- Nagyon fontos az adatok előfeldolgozása, mivel sok zajt tartalmaznak

# Usage profile-ok készítése

## 1. Logikai mapping:

- Logikai szerver nevek fordítása

`/ws/WarehouseReferenceImplementation  
Service` → `warehouseService.`

- Logikai kliens nevek fordítása

`transportService.` → `8087` → `warehouseService`

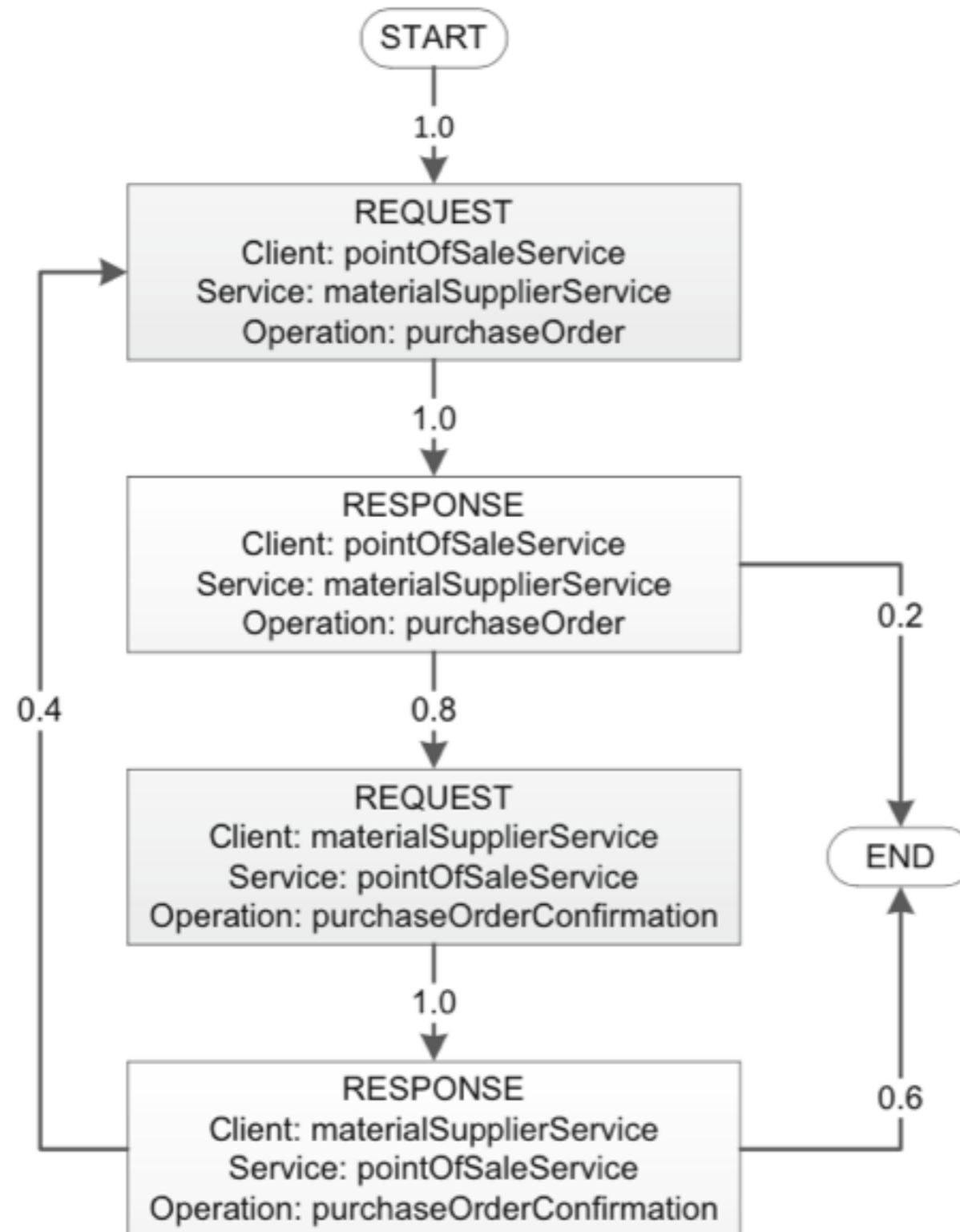
# Usage profile-ok készítése

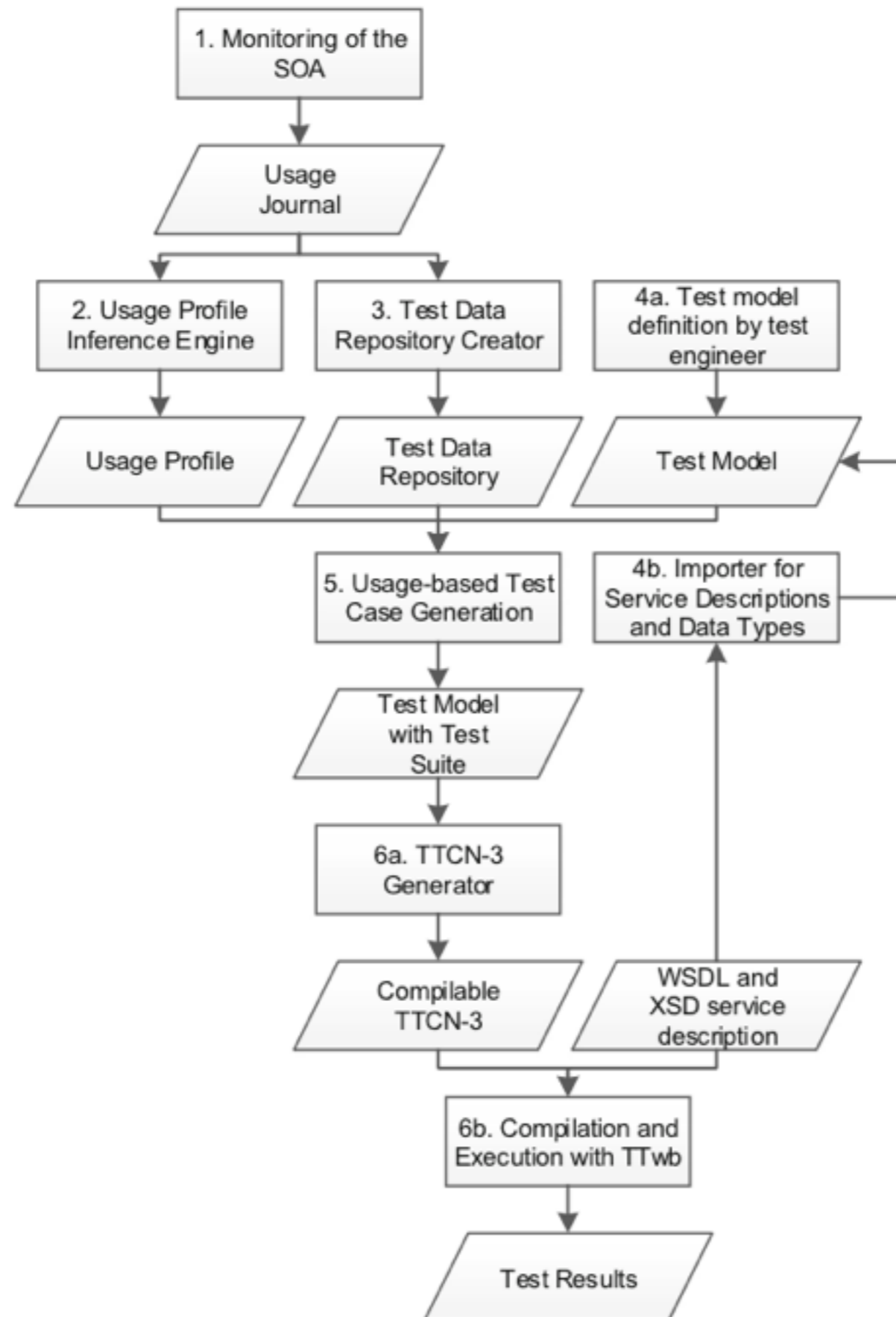
2. Nem SOAP kérések törlése (pl.:WSDL)
3. Kérés válaszok szétbontása külön eseményekbe
4. Események rendezése *orderingID* alapján
5. Request/Response suffix-ek törlése
6. Nem megfigyelt szolgáltatások üzeneteinek törlése

# Usage profile-ok készítése

- Az előfeldolgozott adatokból már készíthető usage profile
- Usage profile:
  - ◉ sztochasztikus leírása a SUT használatának
  - ◉ állapotatlanok
  - ◉ belül Markov modelleket tartalmaznak

# Usage profile-ok készítése

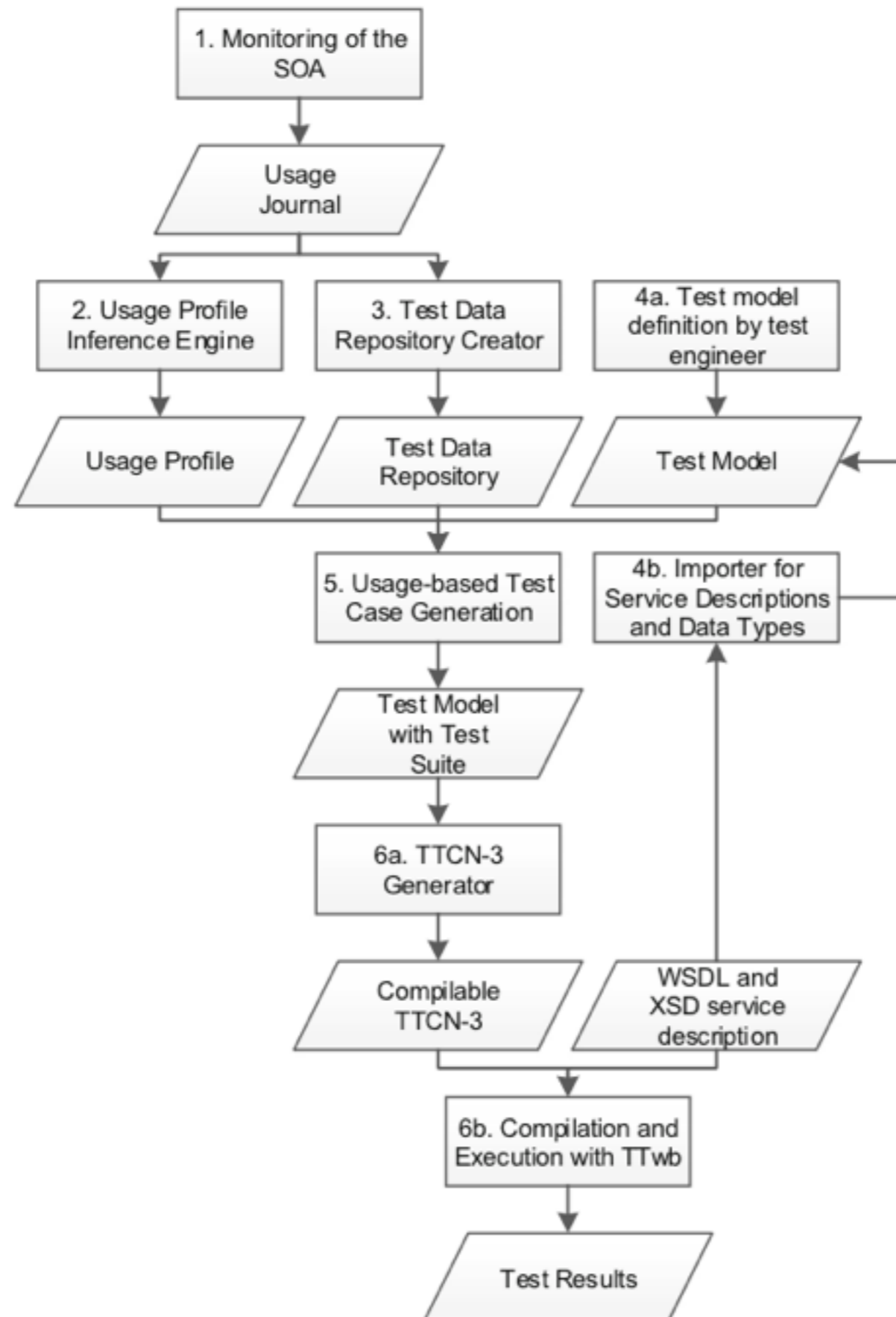




# Teszt adat repository

- Modell alapú megközelítés esetén a különböző interakciók generálása egyszerű
- Viszont honnan jön a teszt adat ezekhez az interakciókhoz
- Korábbi esetekben ez sok esetben manuálisan történt, itt viszont lehet használni a tárolt kommunikációt

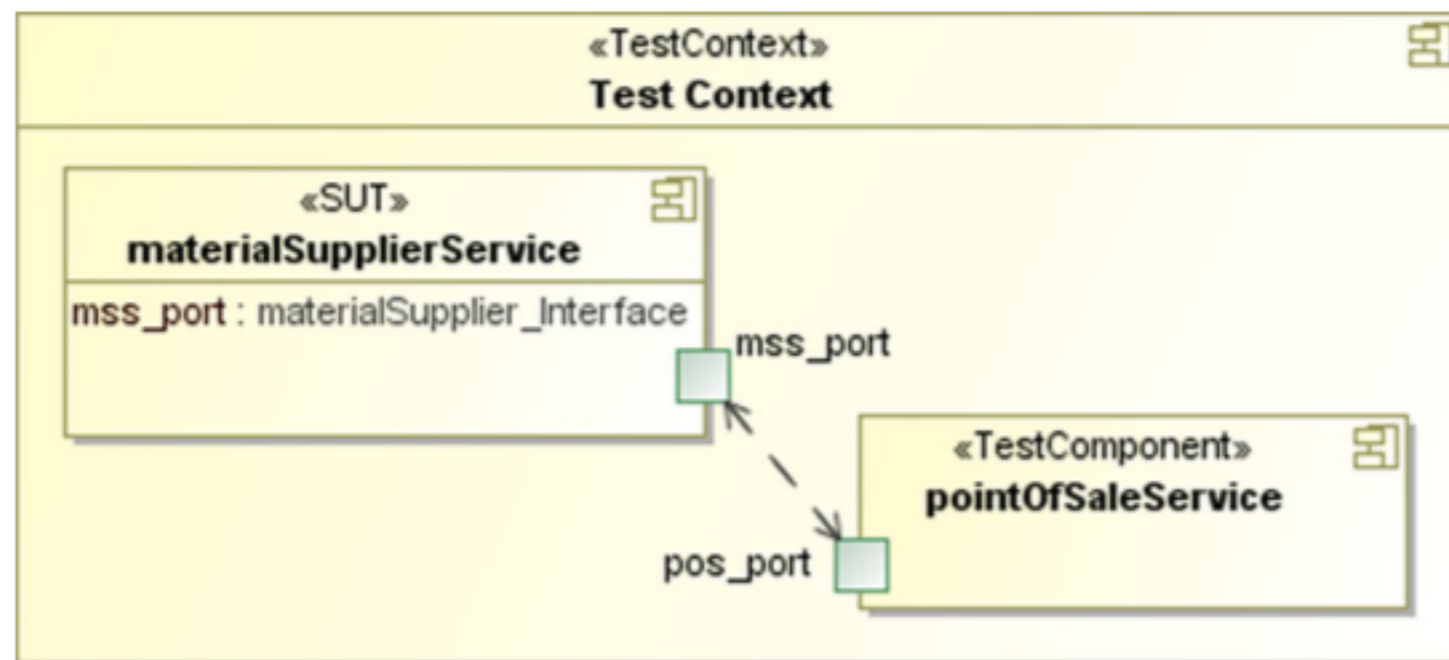


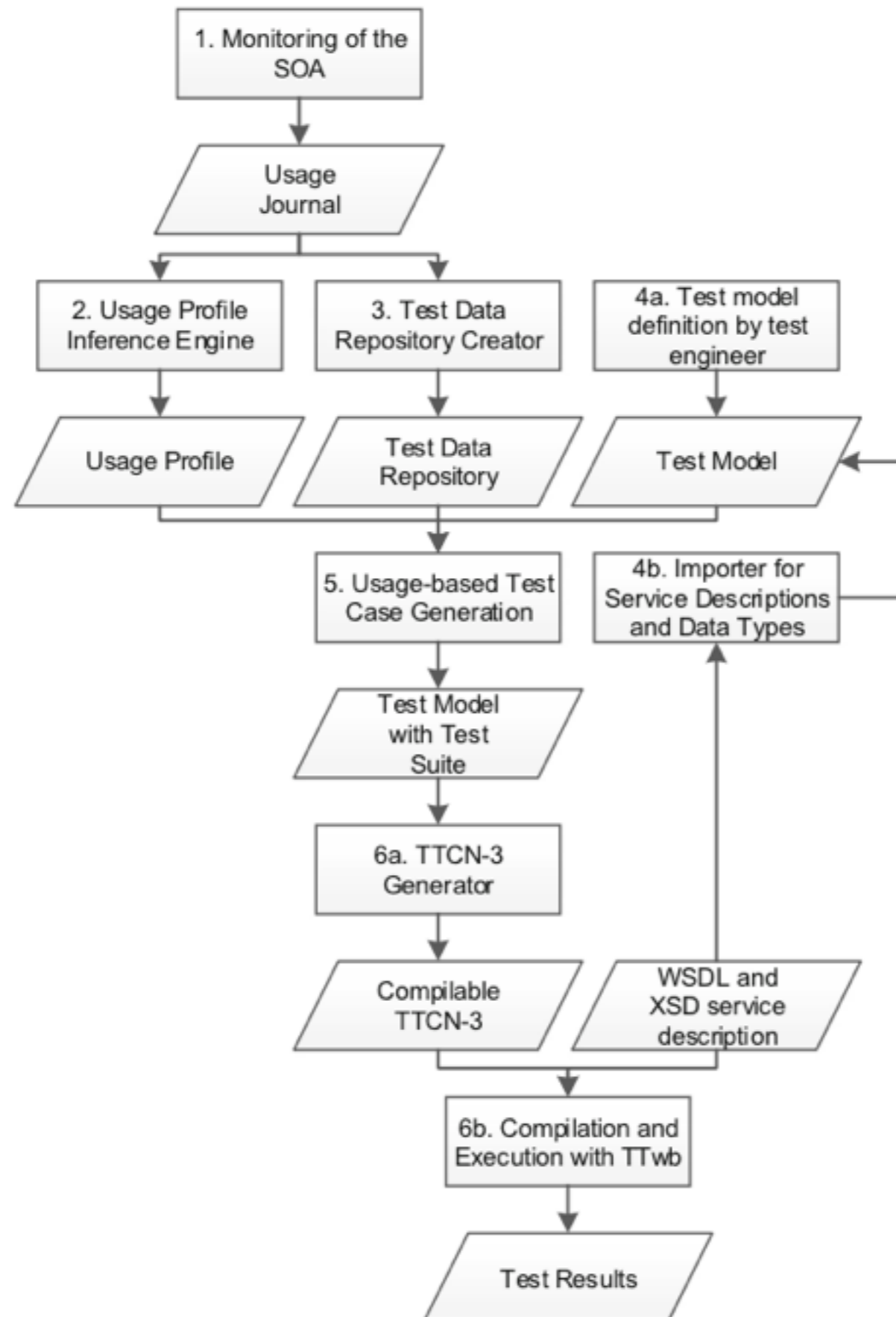


# Teszt modell definíció

- Formális leírás MIDAS DSL-lel:
  - UML és UTP alapú leírás
  - kifejezetten SOA-k közös tesztelésére lett tervezve
- SUT infrastuktúra definíciója (WSDL,XSD)
- Teszt kontextus definíciója (konkrét struktúrája a test bed-eknek, manuális):
  - Logikai szolgáltatás komponensek definíciója (UML komp.,portok)
  - Teszt kontextus készítése (UML komp. diag.)
  - Teszt bed komponensek hozzáadása (minimum 1)
  - Konnektorok hozzáadása

# Teszt modell definíció

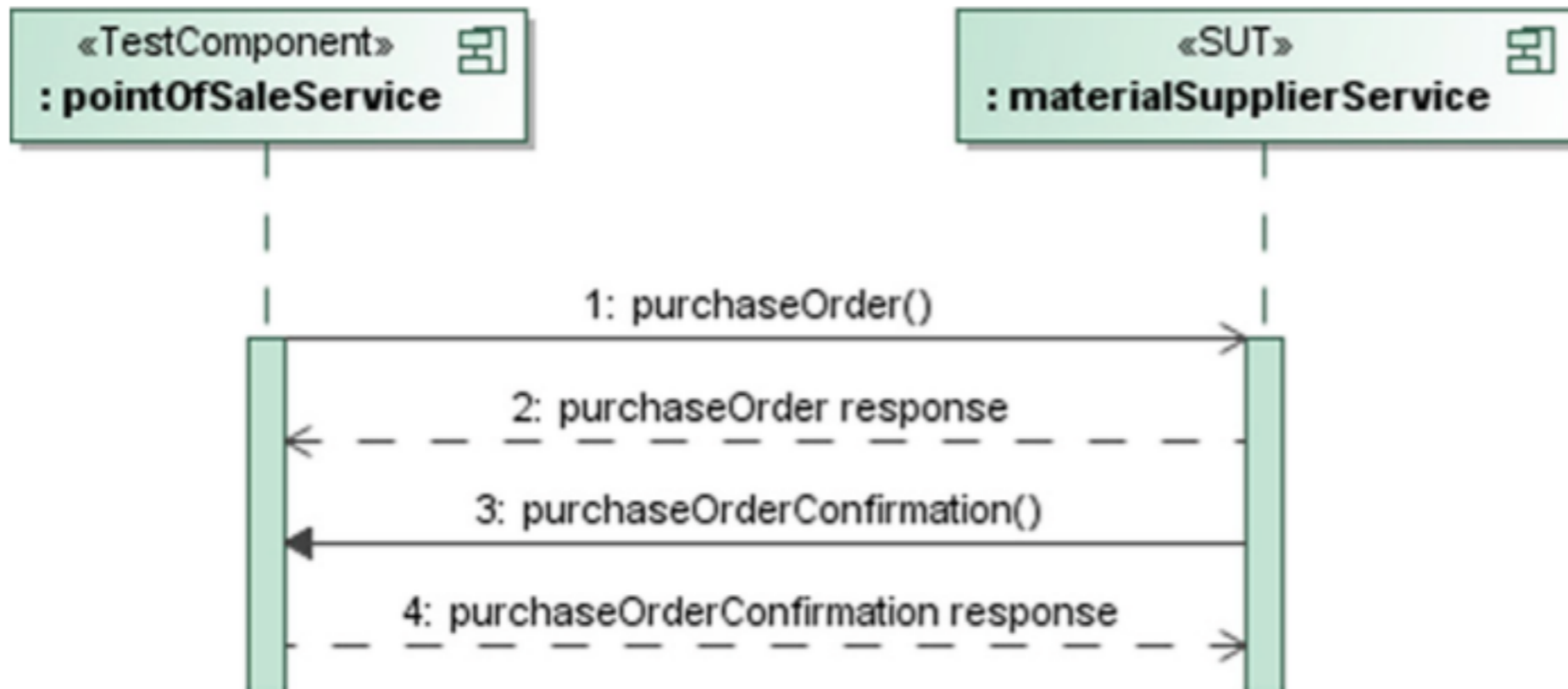


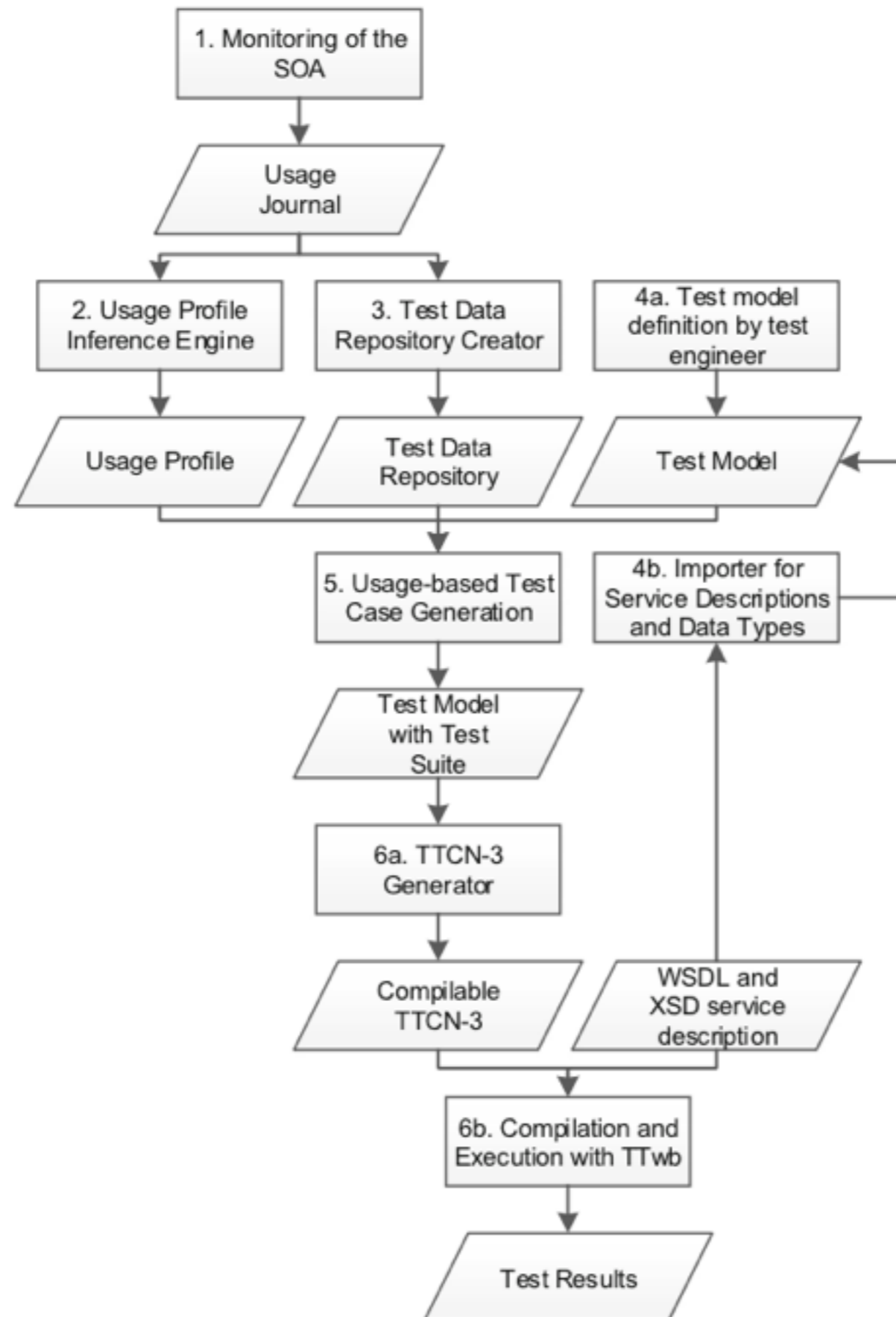


# Teszt generálás

- Két fő komponens:
  - Szolgáltatások közötti interakció szekvenciák generálása a SUT-on belül (usage profile)
  - Az egyes interakciók leírása teszt esetként (teszt modell és használati profil összekapcs.)
  - Tesztekhez az adatok a repositoryból

# Teszt generálás

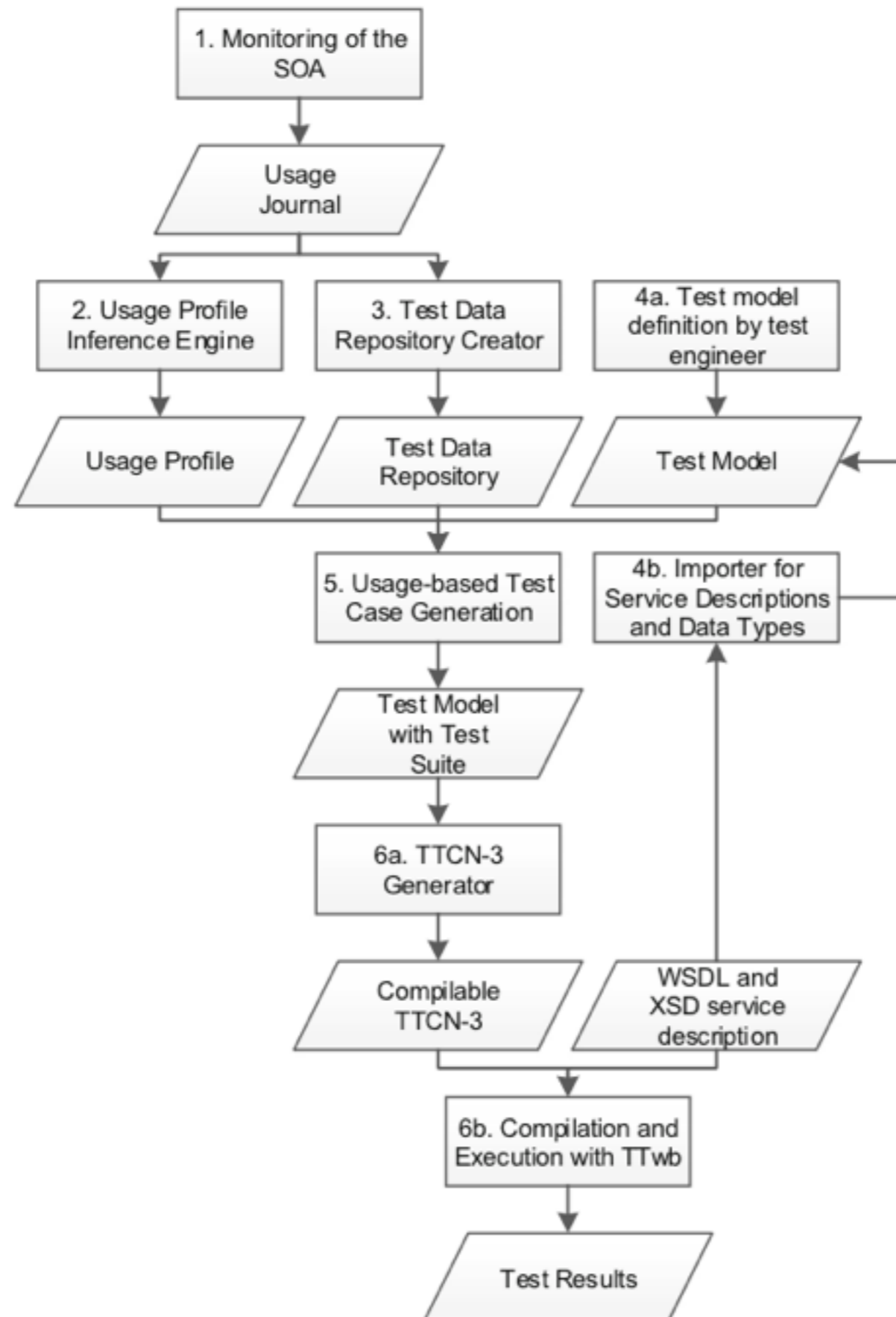




# Teszt futtatása

- Tesztek végrehajtásához a TTCN-3-t használják
- Teszt modellek transzformációja TTCN-3 kompatibilis teszteké
- Ehhez egy TTworkbench nevű tool-t használnak
- Futtatáshoz a TTworkbench:
  - TTthree (TTCN-3-ból futtatható tesztek)
  - TTman (TTCN-3 menedzs. és futtatás)





# Pilot tanulmányok

- Megközelítés használhatóságának bizonyítása
- Két pilot tanulmányban is alkalmazták valós SOA alkalmazásokban
- A tanulmányok célja, hogy betekintést nyújtsanak a usage-based tesztelésbe az ipar szemszögéből
- Tanulmányok:
  - Ellátási lánc menedzsment pilot
  - Egészségügyi ellátás pilot

# Ellátási lánc menedzsment pilot

- GS1 LIM standard alapú SCM rendszer
- A pilot nem implementálta a teljes GS1 LIM-et, csak az alábbiakat vizsgálta:
  - alapanyag ellátói szolg.
  - szállítási szolg.
  - raktár szolg.
  - értékesítés helye szerinti szolg.

# Egészségügyi ellátás pilot

- Health Level Seven International (HL7) standard-ot
- Implementált szolgáltatások:
  - Retrieve Locate Update Service (RLUS)
  - Identity cross-reference service (IXS)

# Konklúzió

- A Usage-based és Model-based megközelítések együttes használata bár kihívások elé állít, számos előnnyel jár
- Usage adatok értékes adatforrások a tesztek számára
- A fejlesztés korai fázisaiban már kell készülni a usage-based tesztelésre, mert a későbbiekben már megvalósíthatatlan feladattá válhat

Köszönöm a figyelmet!