

Web servisai

UDDI

Osvaldas Grigas

Binding

- Surišimas (dar vadinamas susiejimu, saistymu, *binding*) yra tas momentas, kai kvietėjas sužino web serviso adresą ir bendravimo protokolą, kad vėliau galėtų iškviesti serviso operacijas.
- Į šią procedūrą paprastai įeina ir WSDL aprašo nuskaitymas.
- Kvietėjas gali susirišti su web servisu dviem būdais:
 - *static binding* - klientinės programos kompiliavimo/diegimo metu
 - *dynamic binding* - vykdymo metu
- *Static binding* yra paprasčiau realizuoti, tačiau serviso adresas turi būti nekintantis, ir jį reikia žinoti kompiliavimo metu.
- *Dynamic binding* yra kur kas lankstesnis surišimo būdas, nes:
 - nereikia iš anksto žinoti serviso (ir jo aprašo) adreso
 - automatiškai prisitaikoma prie serviso adreso ar protokolo pasikeitimų
- Be abejo, kai naudojamas dinaminis surišimas, reikalingas tarpininkas, kuris padėtų vykdymo metu atrasti norimą servisą.

Dinaminis surišimas naudojamas...

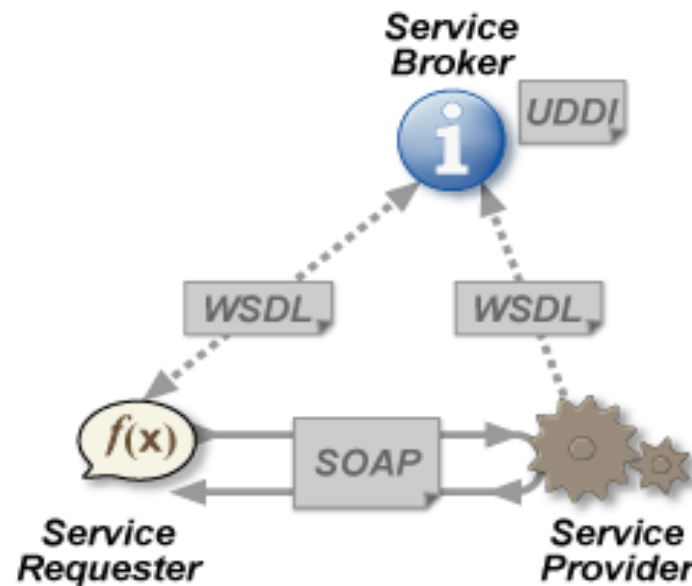
- ... kai serviso adresas negali būti žinomas iš anksto
 - Pavyzdžiui, kiekvieną P2P tinklo dalyvį (*peer*) galima laikyti web servisu. Statinis surišimas neįmanomas, nes dalyviai prisijungia ir atsijungia spontaniškai.
- ... kai serviso adresas gali keistis ateityje
 - Pavyzdžiui, kai programuotojai kuria sistemą, web servisai yra domene "localhost", testavimo metu jie atsiduria domene "testserver", o diegimo metu servisai perkeliama į domeną "services.shoeshop.com".
- ... kai kvietėjui leidžiama rinktis tarp kelių alternatyvių servisų
 - Pavyzdžiui, banko vidiniame tinkle visi pavedimai siunčiami apdoroti bendram transakcijų servisu, kuris veikia serveryje "server1". Jeigu "server1" dėl kokios nors priežasties nustoja veikti, visi kvietėjai dinamiškai prisiriša prie analogiško transakcijų serviso, esančio serveryje "server2", o pavedimai nenutrūksta.
 - Kitas pavyzdys: *online* batų parduotuvė priima atsiskaitymus kreditine kortele, todėl jai reikia kreiptis į vieną iš kortelės nuskaitymo servisų, kuriuos teikia kelios konkuruojančios kompanijos. Paslaugos mokestis jose skiriasi, bet laikui bėgant keičiasi. Todėl batų parduotuvės servisas kasdien dinamiškai susiriša su tuo kortelės nuskaitymo servisu, kurio mokestis tądien yra mažiausias.

Discovery

- Web serviso atradimas (*discovery*) - tai procedūra, kurios metu klientas randa reikalingą servisą ir susiriša su juo dinamiškai.
- Klientas kreipiasi į tarpininką - *servisų registrą* - ir pateikia jam užklausą, kokio serviso ieško.
- Registras randa kriterijus atitinkantį servisą ir grąžina jo adresą bei aprašą (paprastai WSDL) klientui.
- Kad šis modelis veiktų, reikia, kad registre būtų įrašyti servisiškai kartu su papildoma meta-informacija, pagal kurią būtų galima atlikti servisų paiešką.
- Šiuo metu yra keli standartai, aprašantys, kaip turi veikti servisų registrai. Žinomiausi:
 - UDDI (remiamas OASIS, labiausiai paplitęs)
 - ebXML Registry (remiamas OASIS ir UN/CEFACT)

UDDI

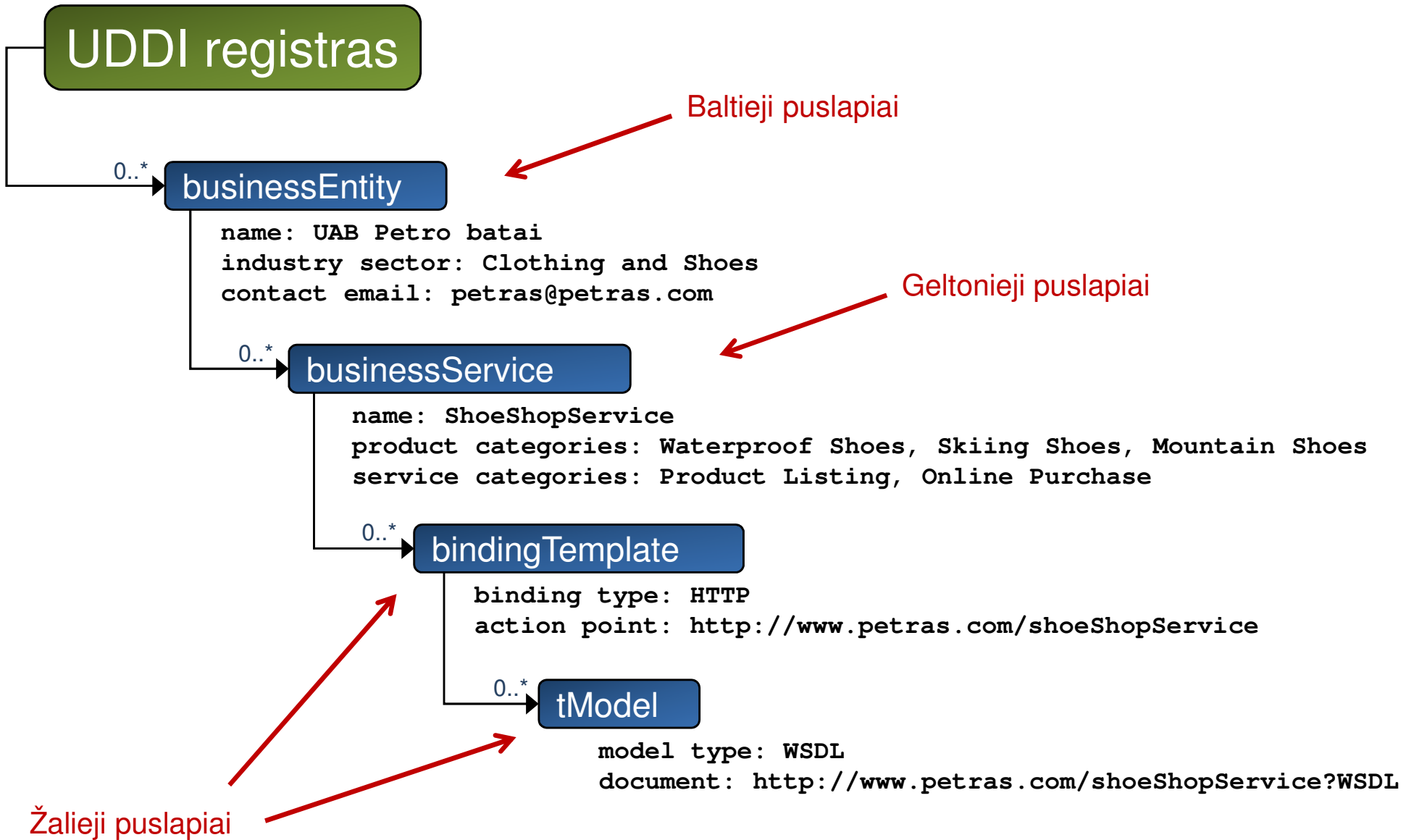
- *UDDI* = Universal Description, Discovery and Integration
- OASIS standartas, aprašantis:
 - kaip paslaugų teikėjai gali registruoti save ir savo servigus UDDI registre
 - kaip potencialūs paslaugų naudotojai gali rasti servigus UDDI registre
- SOAP pranešimai naudojami informacijos apsikeitimui su registru (tiek paieškai, tiek ir servisų registracijai).



UDDI registras

- Duomenys saugomi ir pateikiami XML formatu
- Baltieji puslapiai (*White Pages*)
 - elementai *businessEntity*
 - saugo įmonių ir organizacijų kontaktinę informaciją
 - klasifikuoja pagal veiklą, pramonės sektorių ir pan.
- Geltonieji puslapiai (*Yellow Pages*)
 - elementai *businessService*
 - saugo servisų (paslaugų) pavadinimus
 - klasifikuoja pagal paslaugos pobūdį, produktų rūšį ir pan.
- Žalieji puslapiai (*Green Pages*)
 - elementai *bindingTemplate*
 - nurodo surišimo tipą (protokolą) ir serviso adresą
 - elementai *tModel*
 - laiko nuorodą į serviso techninį aprašą (paprastai WSDL)

UDDI registre saugoma informacija



UDDI interfeisai

- Prieigą prie UDDI registro suteikia keletas standartizuotų API, realizuotų web servisais. Šie web servisai palaiko SOAP/HTTP protokolą.
- Svarbiausi API yra šie:
 - *Publication API*, servisų registracijai:
 - operacija *save_business* - registruoja paslaugų teikėją
 - operacija *save_service* - registruoja servisą
 - operacija *delete_service* - išregistruoja servisą
 - ir kt.
 - *Inquiry API*, servisų paieškai:
 - operacija *find_business* - ieško paslaugų teikėjo
 - operacija *find_service* - ieško serviso
 - operacija *get_tModelDetail* - grąžina serviso techninį aprašą (ppr. WSDL)
 - ir kt.