

.NET 2006 ja sen jälkeen

Ahti Haukilehto
FC Sovelto Oyj
Microsoft Regional Director, Finland



Superior tools, niin mitkä?

- ▶ Visual Studio Team System
- ▶ Team Foundation Server
- ▶ DSL Tools



Visual Studio Team System

- ▶ Sama väline (lähes) kaikkeen sovelluskehitystyöhön
 - Arkkitehtuurisuunnittelu
 - Vain WebService-arkkitehtuuri
 - Sovelluksen suunnittelu, kehittäminen ja debuggaaminen
 - Testaus
 - Tietomallin/tietokannan suunnittelu, toteutus ja testaus
- ▶ Ei kuitenkaan (vieläkään) kelvollista UML-välinettä
 - Tulevaisuus DSL:ssä, mutta ei (vielä) korvaa puutetta

Team Foundation Server

- ▶ Sovelluskehitystiimin palvelin
 - Lähdekoodin versiohallinta
 - Porttaali
 - WorkItem Tracking (WIT)
 - Raportointi
- ▶ Projektin etenemisen kirjaaminen ei ole enää ylimääräinen työ vaan luonteva osa koodin kirjoittamista
 - Tehdään CheckIn-operaatiossa
- ▶ Mutta
 - 1. versio
 - Vain MS (ja .NET) projekteihin
 - Ei sisällä alkuvaiheen (hankkeen kartoitus, vaatimusmäärittely) työvälineitä

- ▶ Vastaavat tuotteet on saatavissa Java-leirissä
 - Eclipse, SubVersion, Cruiser, jne
 - niin ja ANT, JUnit ja
 - Alustariippumattomasti (sekä kehitys- että tuotteen alusta)
- ▶ MS:n ratkaisu on alustariippuva, mutta helpompi ottaa käyttöön.
 - ”Kaikki yhdellä DVD:llä”

DSL (Domain Specific Languages) Tools

- ▶ Systematisoidaan DSL ja DSM (DS Modelling)
 - oman sovelluskehityksen päälle on mahdollista tehdä - tuottavasti - mallinnuskieli
 - Mallista on mahdollista generoida kehystä käyttävä koodi (siis koodigeneraattorin tekemistä on helpotettu ja systematisoitu)
 - Generoitua koodia on mahdollista erikoistaa esim. partial-luokka –tekniikalla
 - hemmetin yksinkertainen tapa – ja toimiva
- ▶ Tavoitteena tietty abstraktiotason nosto
 - Myös `_vanilla_` in-house –ohjelmoinnissa

Miten SOA näkyy MS-leirissä?

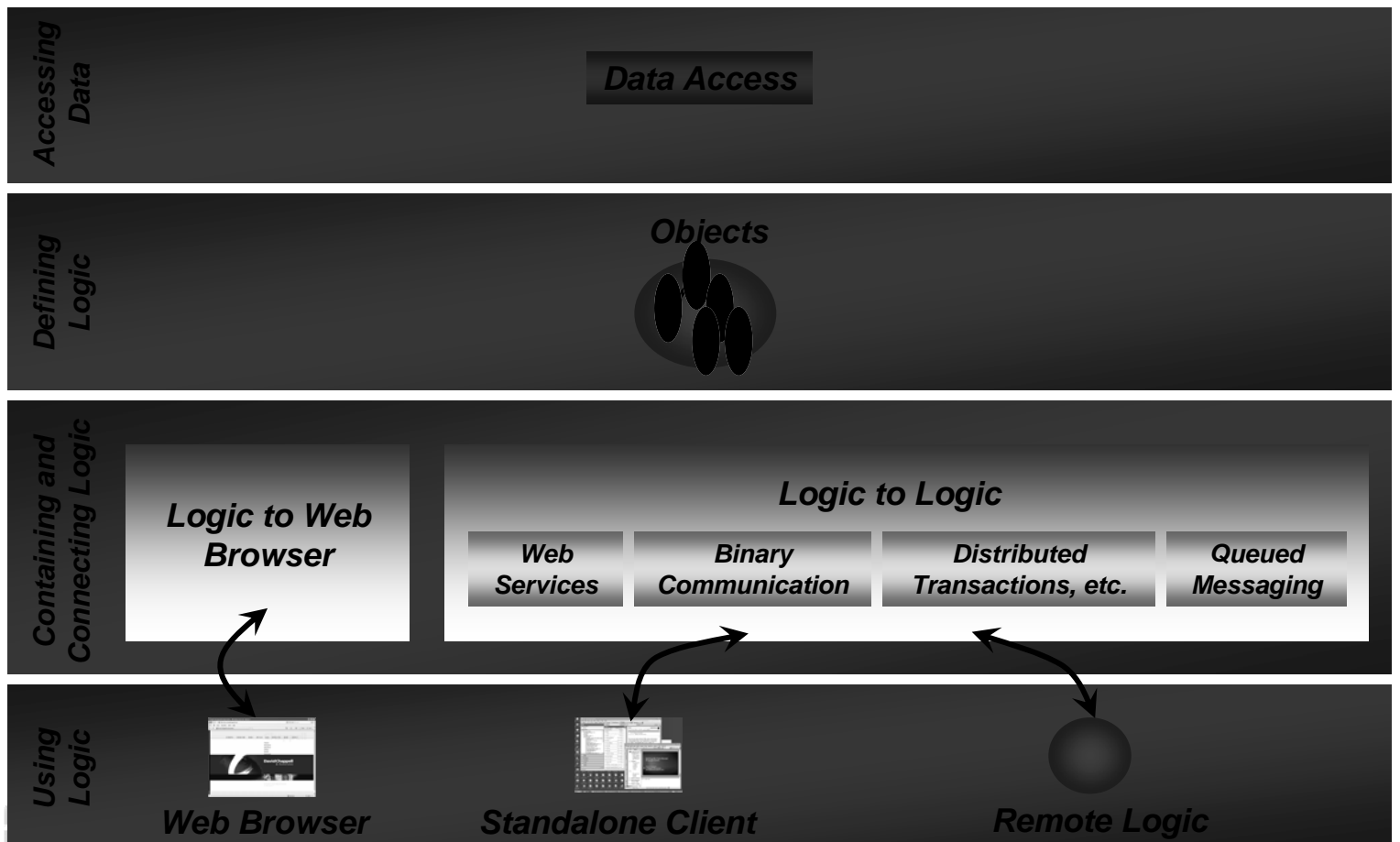
► SOAn kaksi+ määritelmää

- Liiketoimintalähtöinen modulointi
- Viestipohjainen (ei OO) hajautus
- Ja sitten tietty OASIS:en ”määritelmä”

A paradigm for organizing and utilizing distributed capabilities that may be under the control of different ownership domains. It provides a uniform means to offer, discover, interact with and use capabilities to produce desired effects consistent with measurable preconditions and expectations.

Application Platforms Today

Lähde: David Chappell, www.davidchappell.com



Mitä seuraavaksi, .NET 3.0

- ▶ **WPF** (Windows Presentation Foundation)
 - käyttöliittymän kuvaus "relax serialized" UI-olioiden avulla"
 - käyttöliittymätekniikka suoraan DirectX:n päällä
 - UI, joka on Forms-helppo, HTML-määrittelyn tyyppinen, mutta vektorigrafiikkaa, 3D, animaatio jne.
- ▶ **WCF** (Windows Communication Foundation)
 - hajautus "made right"
 - Luovutaan olio-hajautuksesta, SOA loogiseksi tavaksi, fyysinen hajautus konfiguroitavissa
- ▶ **WF** (Windows Workflow Foundation)
 - BPM (Business Process Management) tuotu suoraan frameworkiin – mutta ohjelmointinäkökulma!
 - Käyttötavat vielä alkuvaiheissaan, sivunavigoinnista BL-logiikan irrottamiseen

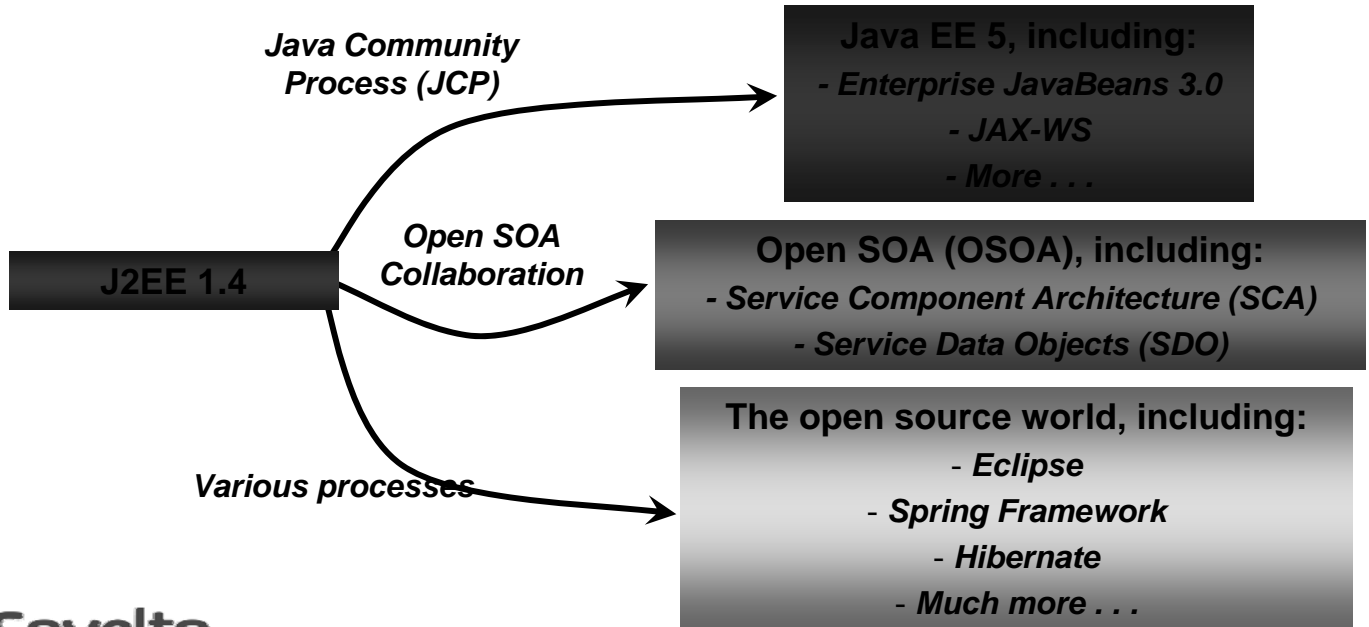
Ja sen jälkeen

- ▶ **LINQ** (Language Integrated Query)
 - ORM "made right"
- ▶ **DataAccess** –logiikan kirjoittaminen on perushommaa – mutta näämmä vaikeaa
 - Molemmissa leireissä on ollut monta iterointikierrosta
 - ORM-välineissä MS-leiri on ollut jäljessä
 - LINQ:n kopiointi Javaan voi olla MissionImpossible – vaikutukset liian laajoja: kielen syntaksista ORM-frameworkiin

Entä Java

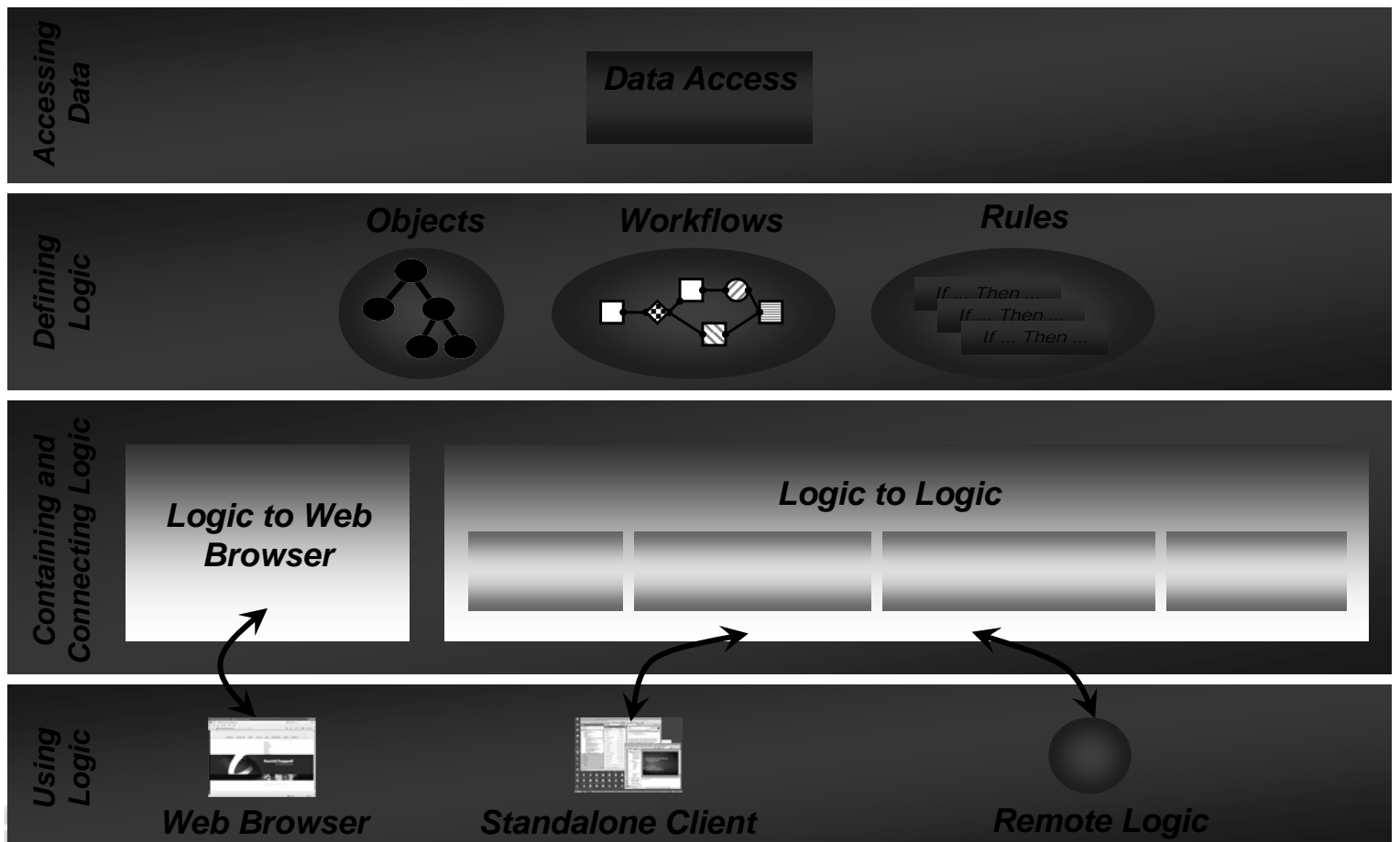
► JCP on tullut tiensä päähän – tai ainakaan ei ole enää ainut Java-kehityspolku

- Lähde: David Chappell, www.davidchappell.com



Application Platforms in 2008

Lähde: David Chappell, www.davidchappell.com



- ▶ Kehityspolku uuteen sovellusarkkitehtuuriin on selkeä
 - Tuotteet ovat jo valtaosin tulleet
 - Selvä, uskottava kehityspolku ja tavoite
 - SOA, myös sovelluskehittäjän näkökulmasta
- ▶ Sovelluskehityksen tuottavuus etusijalla
- ▶ ICT :stä takaisin ATK: hon