Hva er SOA og Web services?
(SOA: Service-oriented architecture / Tjenesteorientert arkitektur)

Brian Elvesæter, SINTEF IKT
Avdeling for samvirkende og tiltrode systemer, Oslo
brian.elvesater@sintef.no

NorStella-DND Web services seminar, Oslo, 9. juni
09:00-09:30
Tjenesteorientert arkitektur
Web services
Oversikt over utviklingen innen EU på dette området?
Hvilke krav bør stilles til grunnleggende IT-arkitektur?
Hva er hensikten med denne nye teknologien, og hvilke muligheter åpner seg?
Towards an Integrated Enterprise Service Architecture
Tjenesteorientert arkitektur
SOA definition

- Service-oriented architecture (SOA)
  - “A set of components which can be invoked, and whose interface descriptions can be published, discovered and invoked over a network.” (W3C)
  - http://www.w3.org/

- Evolution of styles/approaches to designing software systems
  - Data-orientation
  - Functional-orientation
  - Object-orientation
  - Component- and message-orientation
  - Service-orientation
The goal of a SOA is to allow business activities to be orchestrated as components in applications targeting both internal and extra-organizational actors, ultimately enhancing business agility.

- Knowledge about the business
- What are they?
- Notion of business process
- Value chains traversal
- Beyond the firewall
- Cross “trust boundaries”
- Autonomous / Independent actors
- Business and IT alignment: mapping between business activities and - Flexibility - “Re-wire” as needed

Andreas S. T. Brunvoll, “From EAI to SOA”, Presentation given at The Roots Conference, April 2005, Bergen
SOA characteristics

- The service concept applies equally well to the business as it does to software applications.
  - Services can be seen as business capabilities that support the enterprise.
- Services usually represent a business function or domain.
  - Services provide the ‘units of business’ that represent value propositions within a value chain or within business processes.
- Modular design
  - Compositions and granularity
- Services are loosely coupled
  - From compile-time and deployment-time dependencies to run-time dependencies (services)
  - Dynamic discovery and binding
- Services are standardized (“platform independent”)
  - Using Internet/Web protocols and standards as the common “glue” provide “syntactical interoperability”
Fra monolittiske systemer til en tjenesteorientert arkitektur
Web services
Web service definition

- Web service
  - “Applications identified by a URI, whose interfaces and bindings are capable of being defined, described and discovered as XML artefacts. A Web service supports direct interactions with other software agents using XML-based messages exchanged via Internet-based protocols.” (W3C)
  - http://www.w3.org/

- SOA ~ architectural style
- Web services stack ~ technology/protocol standards
- SOA =/= Web services
The Waves of Client/Server Technology

Base Source: Client/Server Survival Guide, 1994
Robert Orfali, Dan Harkey
OS/2 Edition, VNR Computer library + AJB update 2004
OMG (Object Management Group)
OMA (Object Management Architecture)

Application Objects
Horizontal CORBA Facilities

Vertical CORBA Facilities
Object Request Broker (CORBA)

CORBA Services

Lifecycle
Events
Naming
Persistence
Transactions
Concurrency

Externalization
Security
Time
Properties
Query
Licensing
Web services og port 80

Interessen for Web-tjenester har mye av sitt utgangspunkt i problemet for CORBA, MS DCOM og Java RMI med å slippe igjennom for kommunikasjon med ukjente klienter, på grunn av sperrer i brannmurer.

Det ble raskt oppdaget at port 80 (for http Web-browser) kommunikasjon var åpen i de fleste brannmurer, og man begynte å pakke inn informasjon (tunneling) i meldinger som ble sendt gjennom port 80, først innpakket i HTML, deretter i XML.

Dette gav både en teknologi- og markedsmulighet som først Microsoft, deretter IBM var tidlig ute med å utnytte og promotere.
Web services

- Web services can be used to implement service-oriented solutions.
- They adhere to the set of roles and operations specified by the service oriented model.
- They have also managed to establish a standardized protocol stack.
WS-* stack *to-be*

- Simplified version of the to-be WS-* stack
  - Families of related specs not expanded
  - Competing spec families not shown
  - “Historical” or abandoned specs not shown
WS-* stack as-is

- Complete version of the as-is WS-* stack
  - The 3 widely-accepted specs today are the same as 5 years ago
  - Everything else is considered not mature enough
  - Orchestration, discovery and brokering do not exist in today’s world
  - In terms of development process, nothing has changed since CORBA
WS-* specifications / metamodels
Web Service Description Language (WSDL)

- XML-based language for describing functional properties of Web services.
- A service consists of a collection of message exchange end points.
- An end point contains an abstract description of a service interface and implementation binding.
- The abstract description of a service contains:
  - (i) definitions of messages which are consumed and generated by the service
  - (ii) signatures of service operations.
- The implementation binding provides a means to map abstract operations to concrete service implementations.
  - It essentially contains information about the location of a binding and the communication protocol to use (e.g., SOAP over HTTP) for exchanging messages
WSDL 1.1 metamodel

- **WSDL Component**: A container for data type definitions
- **WSDL Document**: A collection of related endpoints
- **Service**: A single endpoint defined as a combination of a binding and a network address
- **Definition**: A concrete protocol and data format specification for a particular port type
- **Port**: An abstract set of operations supported by one or more endpoints
- **Binding**: An abstract, description of an action supported by the service
- **Port Type**: An abstract, typed definition of the data being communicated
- **Message**: An abstract, typed definition of the data type definitions
- **Element**: A container for data type definitions
- **Include**: A single endpoint defined as a combination of a binding and a network address
- **Import**: A collection of related endpoints
- **Definition**: A concrete protocol and data format specification for a particular port type
- **Port**: An abstract set of operations supported by one or more endpoints
- **Binding**: An abstract, description of an action supported by the service
- **Port Type**: An abstract, typed definition of the data being communicated
- **Message**: An abstract, typed definition of the data type definitions
- **Element**: A container for data type definitions
WSDL 2.0 metamodel
Oversikt over utviklingen innen EU på dette området
Interoperability Research

- **Project type:** Network of Excellence (NoE)
- **Full title:** Interoperability Research for Networked Enterprises Applications and Software
- **Project duration:** 3 years
- **Project budget:** 12.0 M€
- **Project funding:** 6.5 M€
- **Partners/contractors:** 50
- **Start date:** Nov 1, 2003
- **Web page:** [www.interop-noe.org](http://www.interop-noe.org)

- **Project type:** Integrated Project (IP)
- **Full title:** Advanced Technologies for Interoperability of Heterogeneous Enterprise Networks and their Applications
- **Project duration:** 3 years
- **Project budget:** 26.5 M€
- **Project funding:** 14.4 M€
- **Partners/contractors:** 19
- **Start date:** Feb 1, 2004
- **Web page:** [www.athena-ip.org](http://www.athena-ip.org)
Rationale

Interoperability, key to increase competitiveness of enterprises

- The cost of non-interoperability are estimated to 40% of enterprises IT budget.

Application integration license revenue

System implementation budget

(Source: the Yankee Group 2001)
The originality of the project is to take a multidisciplinary approach by merging three research areas supporting the development of Interoperability of Enterprise Applications and Software:

- **Architecture & Platforms**: to provide implementation frameworks,
- **Enterprise Modelling**: to define Interoperability requirements and to support solution implementation,
- **Ontology**: to identify Interoperability semantics in the enterprise.
ATHENA - Partners

- AIDIMA
- COMPUTAS
- CRF
- DFKI
- EADS
- ESI
- FORMULA
- FHG/IPK
- GRAISOFT
- IBM
- IC FOCUS
- INTRACOM
- LEKS/IASI-CNR
- SAP
- SIEMENS
- SINTEF
- TXT
- UNI.BORDEAUX I
- UNINOV

Contacts: Rainer RUGGABER; rainer.ruggaber@sap.com
          Klaus-Dieter PLATTE, kdplatte@platteconsult.com
Hvilke krav bør stilles til grunnleggende IT-arkitektur?
Interoperability point of view

- Enterprises
  - Constantly faced with expectations to change
  - Adapt more quickly to changes in the business and economic market
  - Business agility

- Current ICT solutions
  - Inflexible and difficult to adapt to meet the requirements of those changing enterprises

- Future ICT infrastructures
  - Need to separate out knowledge from non-interoperable application systems – from application systems to design services
  - Capture knowledge as formalised models that can be used to configure and adapt the ICT systems
  - Integration through metamodelling and different views pertinent to stakeholders of an enterprise
  - Sustainable and inherently adaptive and interoperable infrastructures
  - User-interaction
  - Trust and confidence

- SOA and Web services – a step in the right direction
4-layered view of an enterprise

- **Business Operational Architecture**
  - Operations
  - Strategy
  - Governance
  - Laws, rules, principles
  - Agreed norms and practices
  - Procedures and routines

- **Enterprise Knowledge Architecture (EKA)**
  - Enterprise methodology
  - Enterprise models
  - Enterprise templates
  - Metamodels and languages
  - Product models
  - Reference architectures
  - Semantics

- **Information and Communication Technology (ICT) Architecture**
  - Business and user services
  - Infrastructure services
  - EKA services
  - Software platforms
  - Modeling tools
  - Management tools
  - Ontology tools
  - Ontology services

- **Ontology**
  - Methodology
  - Reference ontology

**Semantics**

**Business terms**
Holistic Approach to Interoperability

To achieve meaningful interoperability between enterprises, interoperability must be achieved on all layers:

- Business layer: business environment and business processes
- Knowledge layer: organisational roles, skills and competencies of employees and knowledge assets
- Applications, data and communication components
- Semantics: support mutual understanding on all layers

Use SOA principles and Web services to re-architect application systems

ICT Systems

[Diagram showing Enterprise A and Enterprise B with layers labeled Business, Knowledge, Application, and Data.]
8 SOA challenges

- **Service identification.** What is a service? What is the business functionality to be provided by a given service? What is the optimal granularity of the service?

- **Service location.** Where should a service be located within the enterprise?

- **Service domain definition.** How should services be grouped together into logical domains?

- **Service packaging.** How is existing functionality within legacy mainframe systems to be re-engineered or wrapped into reusable services?

- **Service orchestration.** How are composite services to be orchestrated?

- **Service routing.** How are requests from service consumers to be routed to the appropriate service and/or service domain?

- **Service governance.** How will the enterprise exercise governance processes to administer and maintain services?

- **Service messaging standards adoption.** How will the enterprise adopt a given standard consistently?
“Adaptive” service-oriented architecture (ASOA)

ASOA: “Adaptive” service-oriented architecture
SOA: Service-oriented architecture
ASA: Adaptive software architecture
MDD: Model-driven development
PIM: Platform-independent model
PSM: Platform-specific model
Granularity of services

Too fine-grained services => Scalability problem (performance)
Too coarse-grained services => Adaptability/interoperability problem (flexibility)

Cost, performance, flexibility – choose any two!
eContract, grey-box, ...

Black-box vs. White-box vs. Grey-box (autonomous)
Hva er hensikten med denne nye teknologien, og hvilke muligheter åpner seg?

Towards an Integrated Enterprise Service Architecture
New mode of collaboration

Enterprise X

Collaboration space

Composed business services

Shared business model

Business services

Internal services

Enterprise Service Bus

Public view

Private view

Enterprise A

Enterprise Y

Knowledge model

Service
Integrated Enterprise Service Architecture

- **Service infrastructure**
  - ATHENA Integrated Execution Infrastructure
  - Infrastructure services

- **Enterprise services**
  - Business services (providing the ‘units’ of business operations)
  - EKA services for managing knowledge assets (including models and metamodels)
  - MUP services for developing model-generated workplaces

- **User platforms**
  - Model-generated workplaces and Web portals
  - Modelling tools and rich clients