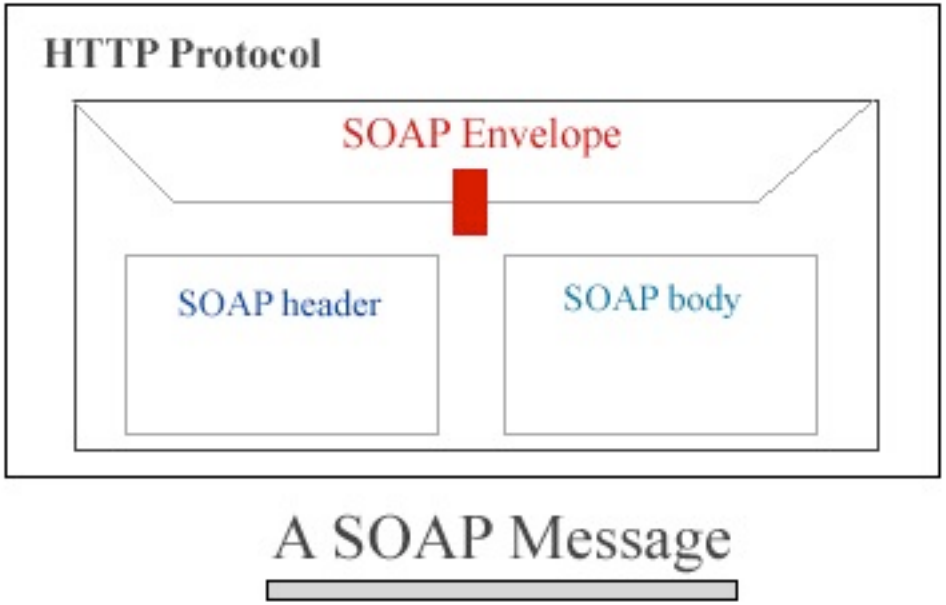
software pilots
TRIFORK.

Hvorfor REST (næsten altid) er bedre end SOAP

“Services” historisk set

- RPC/RMI
- CORBA
  - idl
- RMI-IIOP
  - interoperabilitet ?
  - debug ?
- SOAP
  - interoperabilitet!
  - debug!
- REST
  - interoperabilitet 😊
  - debug 😊

Hvad er SOAP



- Underliggende protokol for webservices
  - Web Service Definition Language

- Baseret på XML

Webservice beskrivelser

- Krav
  - Menneskelig læsbart
  - Entydig
    - Automatisk kodegenerering

- Repræsentation
  - DTD/XSD

- Servicespecifikation
  - WSDL

Beskrivelser - eksempler

Kilde: www.itst.dk/it-arkitektur-og-standarder/standardisering/standarder-for-serviceorienteret-infrastruktur/standarder-for-webservices/filer-til-standarder-for-webservices/OIOWSDL_Vejledning.pdf

```
<?xml version="1.0" encoding="utf-8"?>
<definitions xmlns:tns="http://rep.oio.dk/dkma.dk/homecare/xml.schema/2006.05.11" xmlns:soap="http://schemas.xmlsoap.
wsdl/soap/" xmlns:xsd="http://www.w3.org/2001/XMLSchema" targetNamespace="http://rep.oio.dk/dkma.dk/homecare/
xml.schema/2006.05.11" xmlns="http://schemas.xmlsoap.org/wsdl/">
  <types>
    <xsd:schema elementFormDefault="qualified" targetNamespace="http://rep.oio.dk/dkma.dk/homecare/xml.schema/2006.0
      <xsd:include schemaLocation="../xsd/DKMA_MedicineChestStructure.xsd" />
      <xsd:include schemaLocation="../xsd/DKMA_MedicineChestStructureGet.xsd" />
      <xsd:element name="MedicineChestStructureFault" type="xsd:string" />
    </xsd:schema>
  </types>
  <message name="MedicineChestStructureGetRequest">
    <part name="arg1" element="tns:MedicineChestStructureGet" />
  </message>
  <message name="MedicineChestStructureGetResponse">
    <part name="arg2" element="tns:MedicineChestStructure" />
  </message>
  <message name="MedicineChestStructureFault">
    <part name="fault" element="tns:MedicineChestStructureFault" />
  </message>
```

Beskrivelser - eksempler

```
<portType name="MedicineChest">
  <operation name="MedicineChestStructureGet">
   <input name="MedicineChestStructureGetRequest" message="tns:MedicineChestStructureGetRequest" />
   <output name="MedicineChestStructureGetResponse" message="tns:MedicineChestStructureGetResponse" />
   <fault name="MedicineChestStructureFault" message="tns:MedicineChestStructureFault" />
  </operation>
 </portType>
 <binding name="MedicineChest" type="tns:MedicineChest">
  <soap:binding transport="http://schemas.xmlsoap.org/soap/http" />
  <operation name="MedicineChestStructureGet">
   <soap:operation soapAction="http://rep.oio.dk/dkma.dk/homecare/xml.schema/2006.05.11/#MedicineChestGet"
style="document" />
   <input name="MedicineChestStructureGetRequest">
    <soap:body use="literal" />
   </input>
   <output name="MedicineChestStructureGetResponse">
    <soap:body use="literal" />
   </output>
   <fault name="MedicineChestStructureFault">
    <soap:fault use="literal" name="MedicineChestStructureFault" namespace="" />
   </fault>
  </operation>
 </binding>
 <service name="MedicineChest">
  <documentation>WSDL file for MedicineChest</documentation>
  <port name="MedicineChest" binding="tns:MedicineChest">
   <soap:address location="http://localhost:8080/wstestservice/services/TestPort" />
  </port>
 </service>
</definitions>
```

Menneskelig læsbart ?

Beskrivelser - eksempler

```
   <xsd:schema elementFormDefault="qualified" targetNamespace="http://rep.oio.dk/dkma.dk/homecare/xml.scher
2006.05.11">
     <xsd:include schemaLocation="../xsd/DKMA_MedicineChestStructure.xsd" />
     <xsd:include schemaLocation="../xsd/DKMA_MedicineChestStructureGet.xsd" />
     <xsd:element name="MedicineChestStructureFault" type="xsd:string" />
   </xsd:schema>
 </types>
```

include:

```
      <include schemaLocation="DKMA_MedicineChestItemStructure.xsd"/>
    <element name="MedicineChestStructure" type="homecare:MedicineChestStructureType">
```

include:

```
    <include schemaLocation="DKMA_MedicineChestItemIdentifier.xsd"/>
    <include schemaLocation="DKMA_MedicineChestItemVersionIdentifier.xsd"/>
    <include schemaLocation="DKMA_MedicineChestItemDate.xsd"/>
    <include schemaLocation="DKMA_MedicineChestItemChoiceStructure.xsd"/>
    <include schemaLocation="DKMA_OwnerTypeText.xsd"/>
    <element name="MedicineChestItemStructure" type="homecare:MedicineChestItemStructureType">
            <annotation>
```

Beskrivelser - eksempler

- Multipel import af samme namespace
  - Fejl ?
  - Ignorer?

"*Note: The above is carefully worded so that multiple <import>ing of the same schema document will not constitute a violation of clause 2 of schema Properties Correct (§3.15.6), but applications are allowed, indeed encouraged, to avoid <import>ing the same schema document more than once to forestall the necessity of establishing identity component by component. Given that the schemaLocation [attribute] is only a hint, it is open to applications to ignore all but the first <import> for a given namespace, regardless of the ·actual value· of schemaLocation, but such a strategy risks missing useful information when new schemaLocations are offered.*"

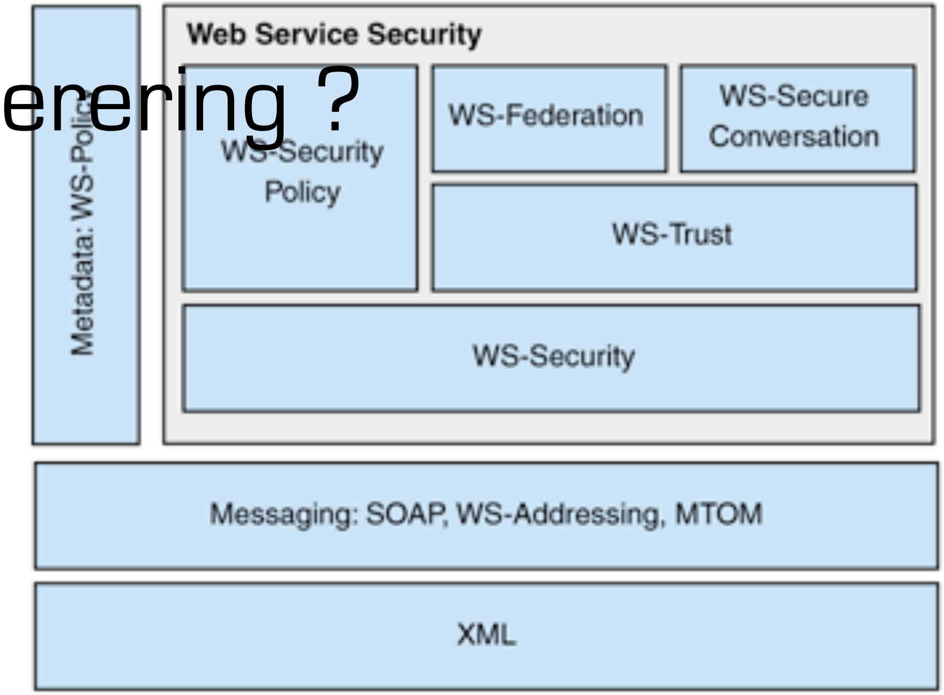
Multipel import af samme namespace

- Kilde...
  - http://www.w3.org/TR/xmlschema-1/

Xerces-2/JAXP1.3 (bl.a. WebLogic 9 og RAD 7.0) -> FEJLER
.NET 1.1 -> OK
.Net 2.0 -> FEJLER
JAXB XJC -> FEJLER
XMLBeans -> OK

Hvad har WS givet os ?

- Interoperabilitet ?
  - Tja...
- Menneskelig læsbart ?
  - Måske...
- Automatisk kodegenerering ?
  - Mmm...joe...
- Løst koblet ?
  - updateCart()
  - calculateInterest()
  - getMedicineChest()



Web Service Security

WS-Security Policy

WS-Federation

WS-Secure Conversation

WS-Trust

WS-Security

Metadata: WS-Policy

Messaging: SOAP, WS-Addressing, MTOM

XML

Hvad er REST ?

- Representational State Transfer
  - Hvad betyder det mon?
- Pragmatisk og simplere tilgang
  - Pragmatisk != ustruktureret
  - Orienteret mod den konkrete opgave

REST - tekniske aspekter

- HTTP
  - GET: Hent ressource
  - PUT: Opdater resource
  - DELETE: Slet ressource
  - POST: Opret (unavngivet) ressource

REST - tekniske aspekter

- Repræsentation
  - XML ?
  - JSON ?
  - ATOM ?

  - Simple repræsentationer baseret på mimetypes
  - Rettet mod genbrug

REST - tekniske aspekter

- Stateless
  - Idempotent
    - Pålidelighed
  - Caching
    - Via udløb? Expires, Max-age, ...
    - Via validering? ETag, ...

REST - tekniske aspekter

- Specifikation
  - WSDL ?
  - WADL ?
  - Fritekst ?
    - Ressource, URL, metode, repr., beskr., statuskoder

- Rettet mod mennesker
  - Simpel fejlfinding

REST - tekniske aspekter

- Ressourceorienteret
  - Alle ressourcer har en URL
  - Hierarkiske URLer
  - Menneskelig læsbare URLer
  - Evt. med request parmetre

  - http://oiorest.dk/danmark/regioner
  - http://oiorest.dk/danmark/regioner/sydjylland
  - http://oiorest.dk/danmark/regioner/sydjylland/byer

REST - tekniske aspekter

- Fordele
  - Performance
    - Stateless
  - Udviklerproduktivitet
    - Test med browser
  - Support
    - Ingen sort magi
  - Mash-ups
- Ulemper
  - End-to-end security
  - Pas på SQL injection!

SQL injection ?

statement:

```
"SELECT * FROM users WHERE name = '" +
userName + "';"
```

setting `username = a' or 't'='t` **yields:**

```
SELECT * FROM users WHERE name = 'a' OR
't'='t';
```

setting `username = a';DROP TABLE users;`

`SELECT * FROM data WHERE 't' = 't`

yields:

```
SELECT * FROM users WHERE name = 'a';DROP
TABLE users; SELECT * FROM DATA WHERE 't'
= 't';
```

Konklusion

statement:

```
"SELECT * FROM users WHERE name = '" +
userName + "';"
```

setting `username = a' or 't'='t` **yields:**

```
SELECT * FROM users WHERE name = 'a' OR
't'='t';
```

setting `username = a';DROP TABLE users;`

`SELECT * FROM data WHERE 't' = 't`

yields:

```
SELECT * FROM users WHERE name = 'a';DROP
TABLE users; SELECT * FROM DATA WHERE 't'
= 't';
```

TRIFORK.

REST

Spørgsmål

TRIFORK.