SSEK-gruppen hos SFM (Svenska försäkringsförmedlares förening)

Agenda SSEK-seminarium

09:00 Inledning	Magnus Karlsson - SPP Liv
09:10 SSEK-introduktion	Peter Danielsson - Skandia Liv
09:20 Affär	Ola Tullsten - SEB Trygg Liv
09:35 Juridik	Håkan Sjödin - Skandia Liv
09:50 Säkerhet	Mats Andersson - Skandia Liv
10:05 Teknisk överblick	Johan Lidö - SEB Trygg Liv
10:20 Paus med kaffe & tilltugg samt möjlighet att träffa leverantörer	
10:35 Teknisk överblick - fortsättning	Gustaf Nyman - Skandia Liv
11:00 Arkitektur för SSEK	Gustaf Nyman - Skandia Liv
11:35 Paus	
11:45 - 12:30 Öppen diskussion och möjlighet att ställa frågor	

Varmt välkomna!

www.ssek.org

SSEK

Introduktion

Peter Danielsson, Skandia Liv

- Deltagit i arbetet med framtagandet av SSEK sedan starten 2002
- Projektledare vid framtagande av Skandia Livs plattformar för elektronisk kommunikation
- Systemansvarig för Skandia Livs plattform för Elektronisk Kommunikation

Historik

- 2001, Krav från kundföretag och distributörer på enkel administration och elektronisk informationsöverföring.
- Flera olika kommunikationslösningar användes initialt.
- 2002, SEB och Skandia påbörjar arbetet med att definiera en standard för försäkringsbranschen.
- SSEK 1.0 släpps 2002-09-10
- SSEK 1.1 släpps 2003-04-28
- Ökad användning och behov, nya internetstandarder
- SSEK 2.0 släpps 2006-05-10 !

Vad är SSEK?

- SSEK är en specifikation som definierar hur affärskritisk information ska kommuniceras säkert mellan organisationer över internet.
- SSEK är bransch-neutral, definierar inte vilka affärsmeddelanden som kan kommuniceras.
- SSEK är baserad på vedertagna internetstandarder som stöds av moderna utvecklingsplattformar

Vad är SSEK, forts...

- SSEK är väl etablerat inom försäkringsbranschen och används av försäkringsbolag, mäklare och Min Pension i Sverige.

- Kommersiella produkter för SSEK-kommunikation finns tillgängliga på marknaden.

Fördelar med att använda SSEK.

- Interoperabilitet, många använder SSEK
- SSEK möjliggör för affär och IT att kunna koncentrera sig på affärsmässiga lösningar mot sina affärspartners.
- SSEK ger tekniskt säker lösning för kommunikation av affärskritisk information.
- Tekniken tillsammans med juridiska avtal ger affärsmässig och juridisk hållbar lösning som fungerar i praktiken.

Nytt i SSEK 2.0

- Vid signering används OASIS WS-security 1.0/1.1
- Timestamp hanteras i WS-Security header vid signering
- Omsändning av meddelande tillåts under vissa förutsättningar
- Standardiserad felhantering
- Förenklat asynkront meddelandeflöde för små organisationer.
- Stöd för Policies
- För ökad interoperabilitet, anpassning till Basic Profile och Basic Security Profile
- http://schemas.ssek.org/ssek/2006-05-10/
- Nytt format på specifikationen

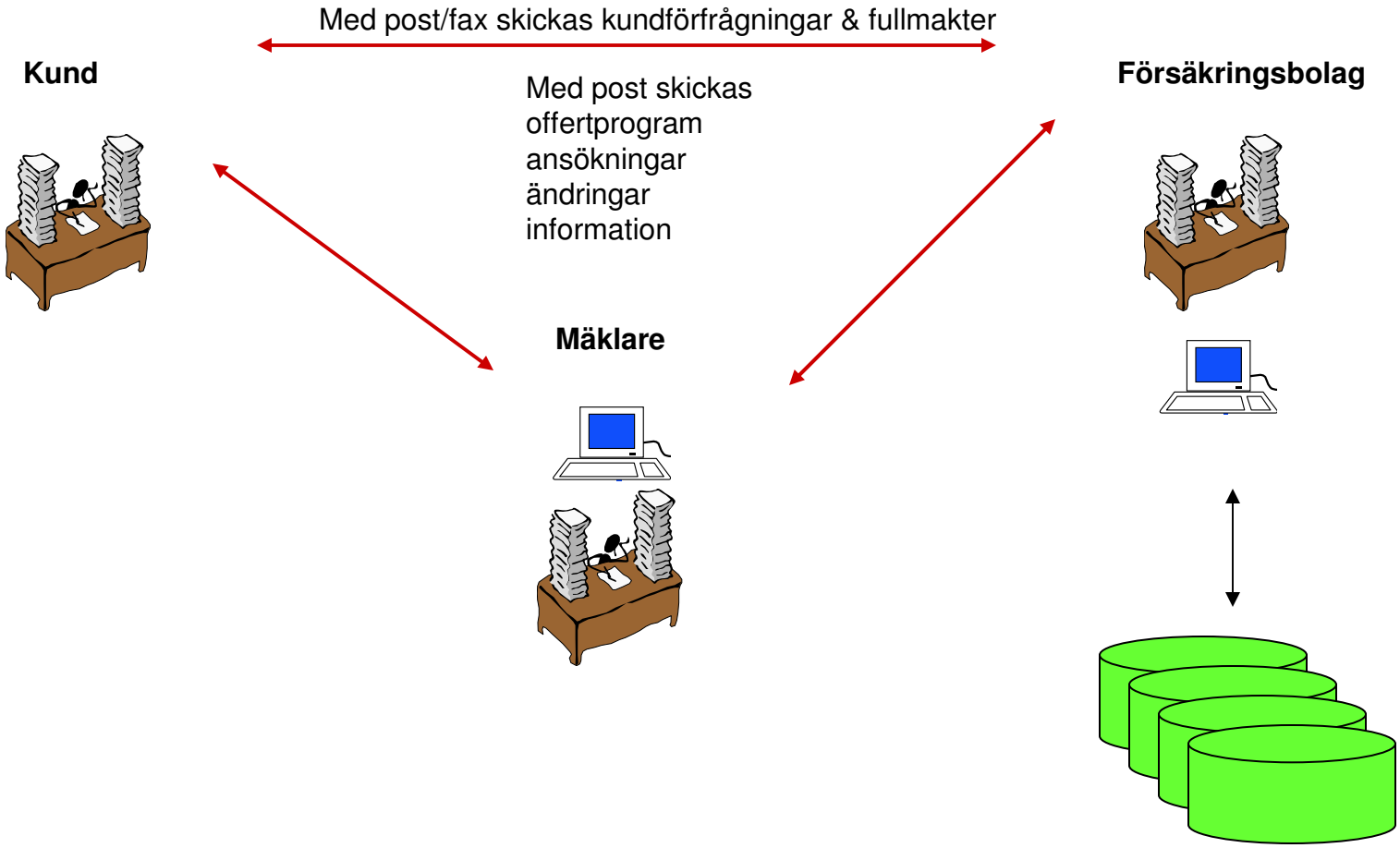
SSEK

Affären

SSEK – Ola Tullsten, SEB Trygg Liv

Mäklarenheten SEB Trygg Liv

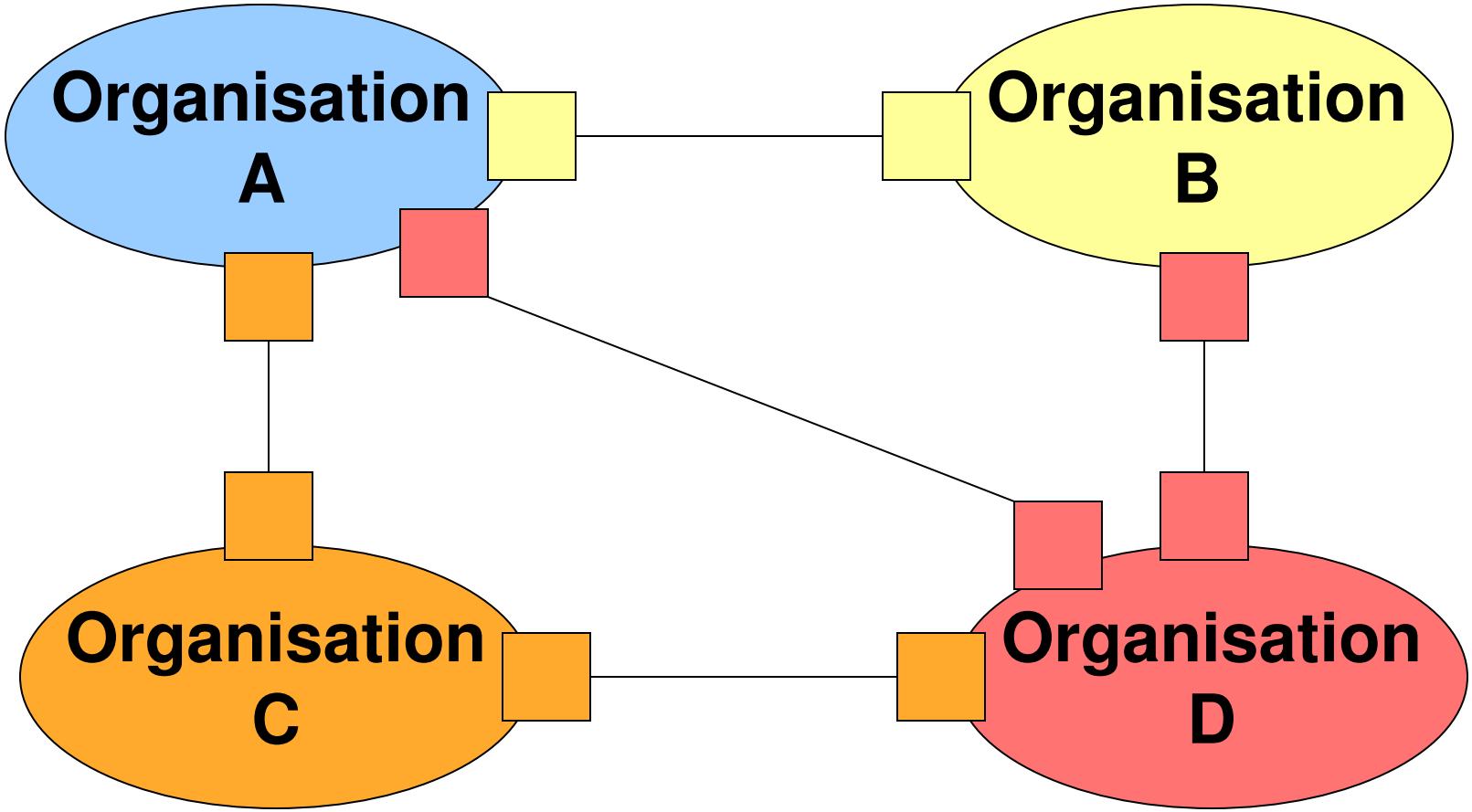
SSEK - Verksamhet i begynnelsen



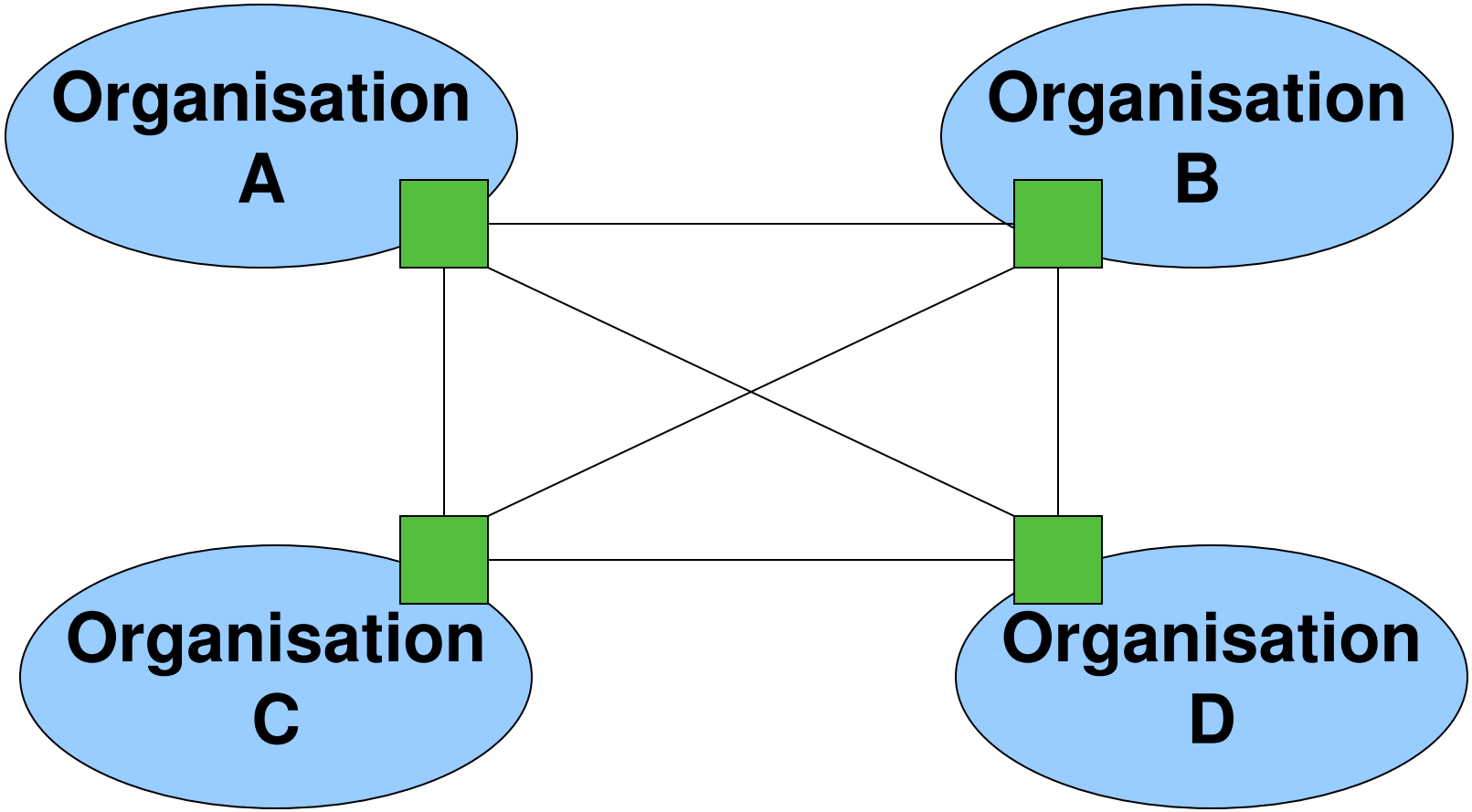
SSEK - Några ”milstolpar”

- Max Matthiessens UIG (**U**pphandling **I** **G**rupp), 2000
- Bolagens SSEK samarbete, 2003
- Min Pension, 2004-2005
- Kraftig ökning av antalet mäklare/företag som vill arbeta elektroniskt, 2005-2006

Innan SSEK


Organisation A
Organisation B
Organisation C
Organisation D

Efter införandet av SSEK



Fördelar för verksamheten

- Standarden har blivit etablerad
- Försäkringsbolag och Min Pension använder sig av SSEK
- Behöver inte diskutera olika säkerhetslösningar
- Verksamheten kan koncentrera sig på att göra affärer

Framtid

- Snabbare sammanställningar av försäkrings-uppgifter till kund
- Fler säkra elektroniska nytecknings- & ändringsmöjligheter
- Snabbare riskbedömning med säker överföring av hälsouppgifter från sjukhus och försäkringskassor

SSEK

Juridiska frågeställningar

Håkan Sjödin

SSEK - Juridiska frågeställningar

- Hur överförs viljeyttring från A till C via Uppdragstagaren B?
  - Elektroniskt signerade "dokument"
  - Filöverföring

SSEK - Juridiska frågeställningar

- Förutsättning:
- Elektronisk signatur är rent kommersiellt tillgängligt i begränsad utsträckning, inte på individnivå
- Filer kan inte "vidarebefordras" elektroniskt eftersom kommunikationskanalerna går mellan två punkter (A och B)
- organisationscertifikat används i dagens tillämpningar
- grunduppgifter (förändringar i lönesystem) överförs från A
- Detta kräver bearbetning följt av ny överföring (B och C)

SSEK - Juridiska frågeställningar

- Lösning
- Elektroniska kommunikationskanaler kopplas ansvars- och
- avtalsmässigt mellan två punkter i taget:
- Uppdragstagaren B överför grunduppgifter från Kundföretaget A enligt uppdragsavtal
- B bearbetar inkomna uppgifter och överför dessa till Försäkringsgivaren C med bindande verkan för A med stöd av fullmakt

SSEK - Juridiska frågeställningar

- Ansvarsreglering Försäkringsgivare - Kundföretag
  - Försäkringsgivaren ansvarar för försäkringsavtalet, hantering av den information som mottagits och för lämnad information
  - Kundföretaget ansvarar för att gjorda utfästelser till anställd tryggas på överenskommet sätt
  - Kundföretaget ansvarar för egen behandling av information och för information som skickas - direkt eller av Uppdragstagare.
  - Vald teknik är avtalad att vara bevis nog vid tvist
  - Kundföretaget ansvarar för anlitad Uppdragstagare och för att denne har fullmakt som täcker den elektroniska hanteringen
- Den anställdes eventuella valmöjligheter – en intern fråga som dokumenteras inom Kundföretaget!

SSEK - Juridiska frågeställningar

- Ansvarsreglering Uppdragstagare - Kundföretag
  - Ansvarar gentemot Kundföretaget för hanteringen av mottagna uppgifter - bearbetning resp överföring till Försäkringsgivaren
  - Särskilt avtal träffas om rådgivning, bearbetning, överföring och ansvar
  - Ansvarsförsäkring som täcker detta tecknas
  - Fullmakt för Uppdragstagaren att binda Kundföretaget vid uppgifter som överförs av Uppdragstagaren till Försäkringsgivaren
  - Skötselfullmakt täcker ej detta!

  – Om individuell valrätt skall administreras krävs fullmakt från varje individ att binda denne vid överförd information

SSEK - Juridiska frågeställningar

- Ansvarsreglering Administratör
  - Kan avtala med Försäkringsgivare om affärsvillkor samt elektronisk kommunikation - Kundföretag ansluter sig
  - Skall ha fullmakt, som Uppdragstagare, att binda Kundföretaget vid uppgifter som överförs till Försäkringsgivaren
  - Ansvarar gentemot Kundföretaget för hanteringen av mottagna uppgifter: bearbetning resp inleverans till Försäkringsgivaren
  - Ansvarar mot Försäkringsgivaren för att fullmakt finns
  - Ansvarsförsäkring som täcker agerande utan fullmakt
  - Om individuell valrätt skall administreras krävs fullmakt från varje individ att binda denne vid överförd information
    - Individuellt val dokumenteras då av Administratören enligt uppdragsavtal med individen

SSEK

Säkerhet

Mats Andersson

- IT-Säkerhetsarkitekt på Skandia Liv
- Ordförande för SSEK-gruppen hos SFM
- Deltagit i arbetet med SSEK 1.1 och 2.0

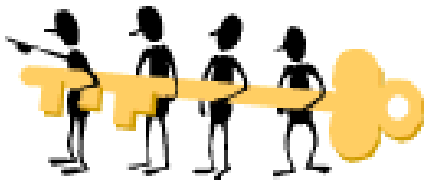
SSEK uppfyller krav från affär och juridik

- "**Säkra** webbtjänster för **affärskritisk** kommunikation"
- Affär som i den manuella världen hade krävt signatur på papper ska kunna ske elektroniskt
- Säkerhetsmodellen i SSEK uppfyller ställda krav

Vad som krävs

- Identifiering av kommunicerande parter (autentisering)
- Skydd mot obehörigas insyn (sekretess)
- Förändringsskydd av information (integritet)
- Binda information till avsändaren (oavvislighet)

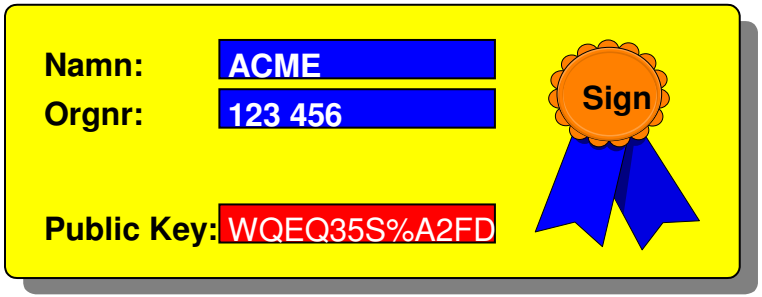
PKI (Public Key Infrastructure)



- Identifiering av kommunicerande parter
  med **digitala certifikat**
- Skydd mot obehörigas insyn
  skapas med **kryptering av kommunikationen**
- Förändringsskydd av information
  skapas med **elektroniska signaturer**
- Oavvislighet
  skapas med **elektroniska signaturer**
  - Genom att arkivera signerade dokumenten kan information säkert kopplas till avsändaren

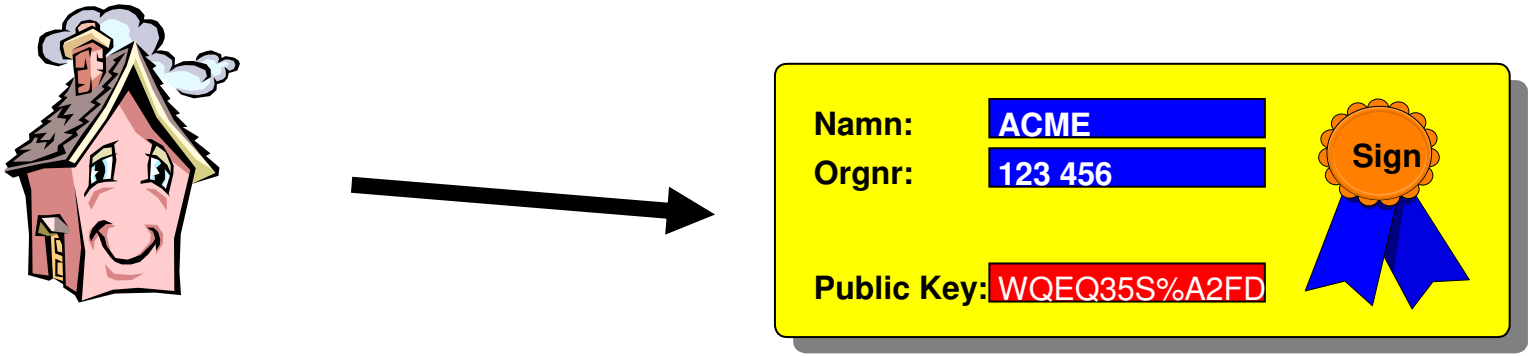
Digitalt certifikat för **organisationer**

- Är en digital legitimation för företag
  - Innehåller publik information om företaget
  - Innehåller **publik nyckel** för kryptering och verifiering av signatur
- Till certifikatet hör en **privat nyckel** som används för signering och för att dekryptera information
- Certifikatet är utformat enligt **X509v3**



Att lita på digitala certifikat

- Certifikatets äkthet garanteras av en betrodd, tredje part - en CA (Certificate Authority)
- En CA ger ut certifikatet och kontrollerar då organisationens identitet
- CA signerar certifikatet elektroniskt
- En CA publicerar även revokeringslistor (CRL)
- SSEK specificerar inte vilken CA som ska utnyttjas

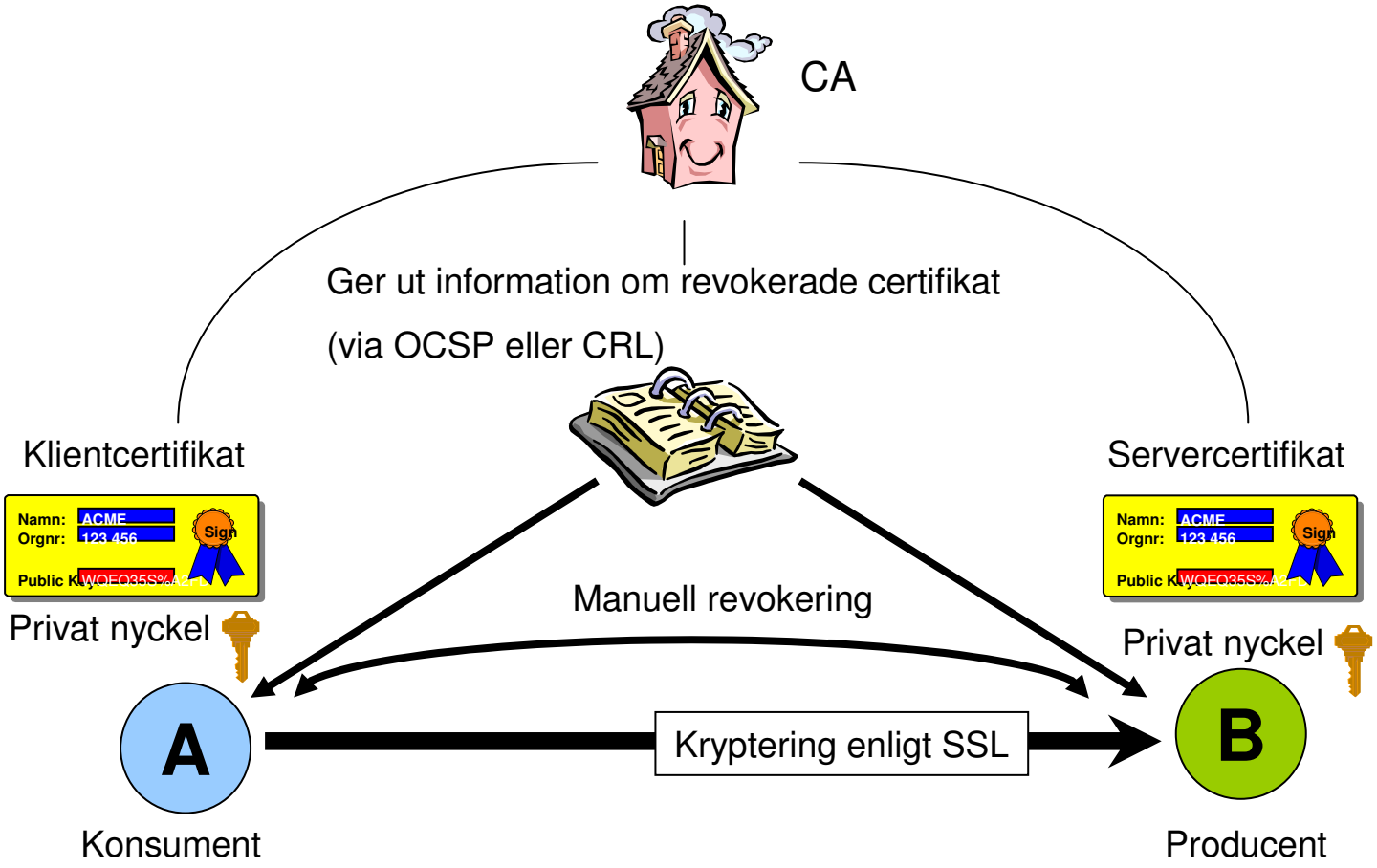


Nivåer i säkerhetsmodellen

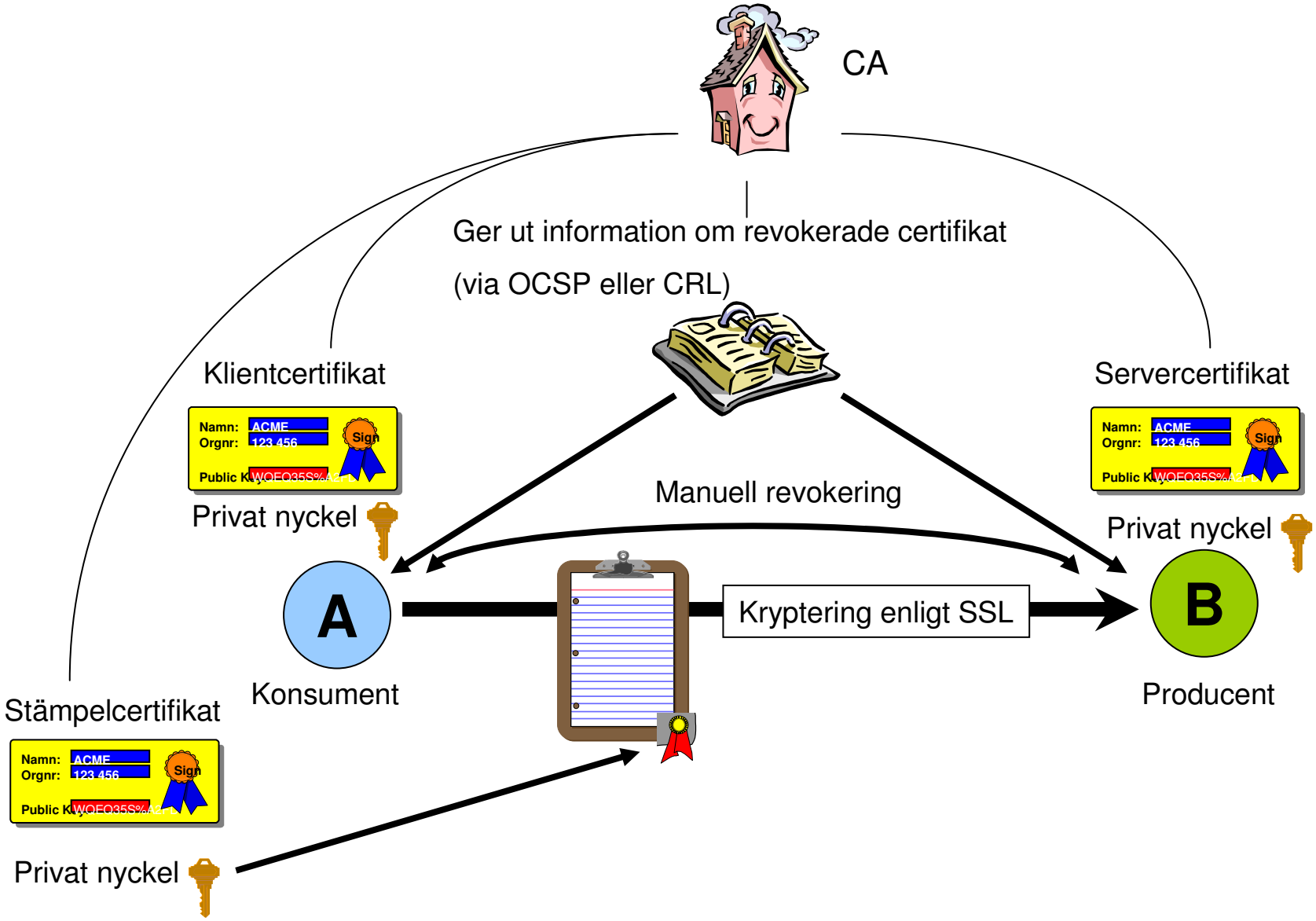
SSEK 2.0

- Transportsäkerhet
  - Sekretess och autentisering av mottagaren
  - Sekretess och autentisering av avsändare och mottagare
- Meddelandesäkerhet
  - Ingen meddelandesäkerhet
  - Oavvislighet och integritet

PKI i SSEK - Transportsäkerhet



PKI i SSEK - Meddelandesäkerhet



Att komma överens om

- CA och certifikattyp
- Säkerhetsnivåer i meddelande och transportsäkerhet
  - Tjänsten avgör lämpligen vilken säkerhetsnivå som bör användas
- Revokeringsmodell
- Komma överrens om namnsättningen av de kommunicerande parterna

Sammanfattning

- SSEK uppfyller krav från affär och juridik
- SSEK utnyttjar PKI för sin säkerhetsmodell
- Högre säkerhet än kryptering av kommunikationen och autentisering av mottagaren är valfri.

SSEK

Teknisk överblick

SSEK – Johan Lidö, SEB Trygg Liv

Utvecklare

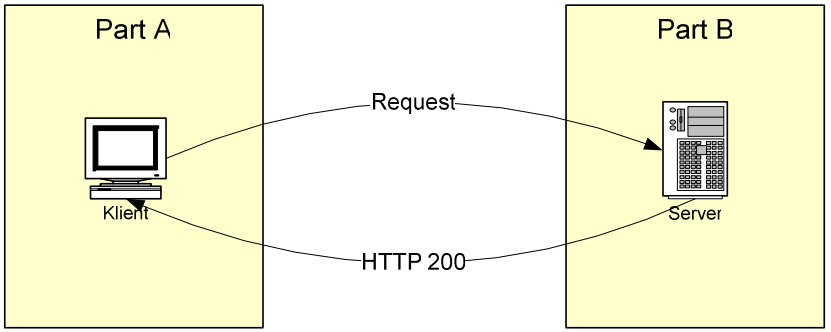
Med i SSEK gruppen sedan starten

SSEK – Meddelandeflöden

- Enkelriktat meddelandeflöde
- Synkront meddelandeflöde
- Asynkront meddelandeflöde med leverans
- Asynkront meddelandeflöde med hämtning

SSEK – Enkelriktat flöde

Informationslämning där konsument inte behöver kvitto eller svar.



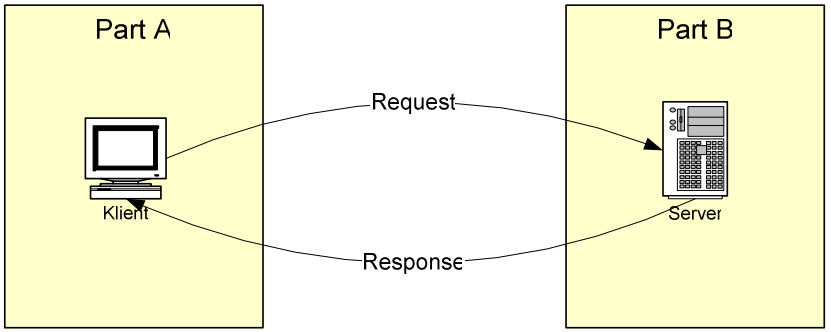
Ex:

Loggning på server

Statusinfo från klient.

SSEK – Synkront flöde

Tjänst där konsument vill ha svar och resultat direkt.



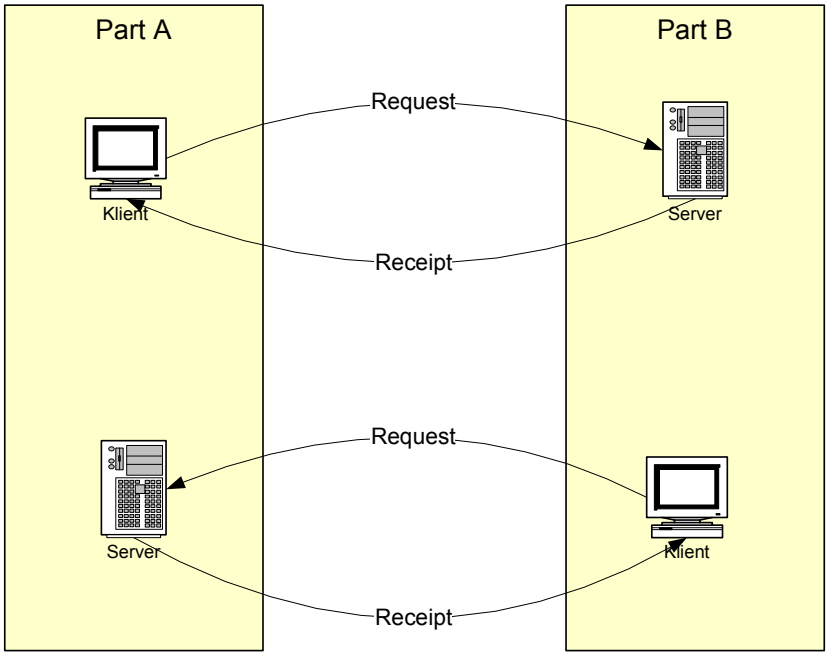
Ex:

Nyteckning av försäkring

SSEK – Asynkront flöde med leverans

Effektivt sätt att behandla data, behandlingen sker när producent tycker det är lämpligt. Server behövs i båda ändar.

Ex:
Nyteckning av försäkring

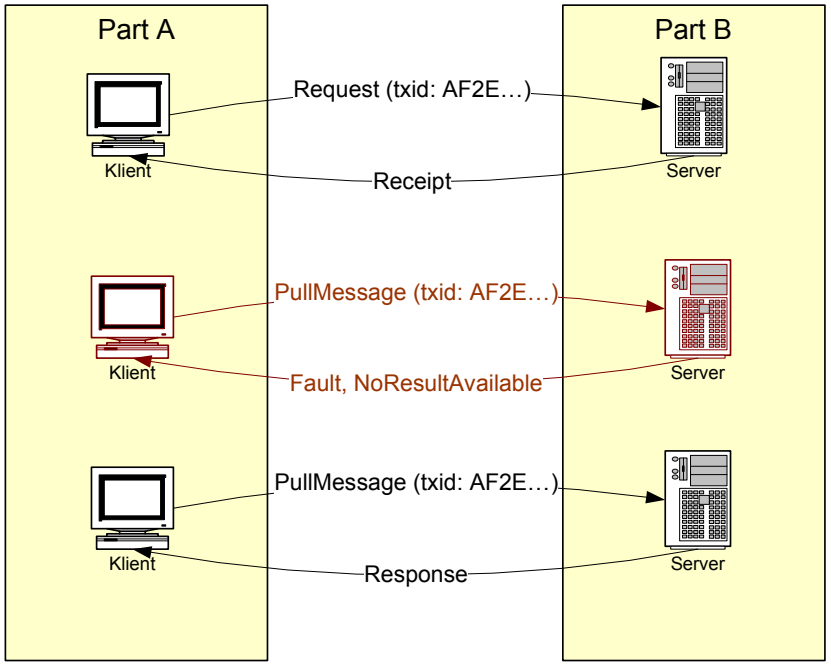


SSEK – Asynkront flöde med hämtning

Effektivt sätt att behandla data, behandlingen sker när producent tycker det är lämpligt. Dock behövs ingen server hos Part A.

Ex:

Nyteckning av försäkring



SSEK – Policy och WSDL (I)

- WSDL
  Vad som kommuniceras.

- Policy
  Hur det kommuniceras.

SSEK – Policy och WSDL (ex:)

```
<definitions name="SSEK_Demo" xmlns:wsp="...">
    <wsp:UsingPolicy/>

    <wsp:Policy wsu:Id="SSEK">
        <sp:ServiceAssertion>
            <sp:IdType>CN</sp:IdType> <sp:UseTxId/>
            <sp:TransportLevelSecurity>SSL</sp:TransportLevelSecurity>
            <sp:MessageLevelSecurity>Signature</sp:MessageLevelSecurity>
        </sp:ServiceAssertion>
    </wsp:Policy>

    <wsp:Policy wsu:Id="SSEKPullPush">
        <sp:OperationAssertion>
            <sp:SupportsAsynchPull />
            <sp:SupportsAsynchPush />
        </sp:OperationAssertion>
    </wsp:Policy>
    <wsp:Policy wsu:Id="SSEKResend"/>

    <types>
        <xsd:schema> <xsd:element name="Nyteckning">
            <xsd:element name="personnummer"/><xsd:element name="premie"/>
         </xsd:element> </xsd:schema>
    </types>
    <message name=”Nyteckning_Request/>
    <portType/>

    <binding name="SSEK_Binding">
        <wsp:PolicyReference URI="#SSEK"/>
        <soap:binding style="document" transport="http://schemas.xmlsoap.org/soap/http" />
        <soap:operation name="Nyteckning">
            <wsp:PolicyReference URI="#SSEKPullPush" />
        <soap:operation />
    </binding>

</definitions>
```

SSEK – AMH exempel


```
<ssek:SSEK xmlns:ssek="…" xmlns:soap="…" xmlns="…" ssek:AsynchMethod="AsynchPull">
        <ssek:SenderID type="CN">Broker</ssek:SenderID>
        <ssek:ReceiverID type="CN">SEB</ssek:Receiver>
        <ssek:TxId>E50321E0-BA38-45C5-85D3-C71435B1ACC4</ssek:TxId>
</ssek:SSEK>
<soap:Body>
      <Nyteckning>
            <personnummer>710319-0123</personnummer>
            <premie>3000</premie>
      </ Nyteckning >
</soap:Body>
</soap:Envelope>
--------
<ssek:SSEK xmlns:ssek="…"xmlns:soap="…" >
        <ssek:TxId>E50321E0-BA38-45C5-85D3-C71435B1ACC4</ssek:TxId>
</ssek:SSEK>
<soap:Body>
      <ssek:PullMessage/>
</soap:Body>
--------
<soap:Fault xmlns:ssek="…" xmlns:soap="…">
      <ssek:faultcode>NoResultAvailable</ssek:faultcode>
</soap:Fault>
--------
<soap:Body xmlns:soap="…" xmlns="…">
      <NyteckningResponse>
            <fnr>97001232221</fnr>
      </NyteckningResponse>
</soap:Body>
```

SSEK

Teknisk översikt del 2

Arkitektur för SSEK

Gustaf Nyman, Skandia Liv

- Systemprogrammerare och arkitekt.
- Arbetat med soap/webservices/soa sedan 1999.
- Deltagit i arbetet med SSEK 1.1 och 2.0.
- Utvecklat system för SSEK 1.1 och 2.0.
- Anställd på Pluvia Konsult AB.
- Representerar Skandia Liv i SSEK-arbetet

SSEK 2.0 designmål

- Tekniskt säkert informationsutbyte mellan organisationer.
- Enkelt omsätta avtalade tjänster i praktiskt tjänsteutnyttjande.
- Interoperabilitet mellan implementationer och plattformar.

SSEK 2.0 designkriterier

- Använda befintliga specifikationer om möjligt
- Egna tillägg där specifikationer saknas
- Enkelt att implementera med befintliga verktyg för utveckling av webservices
- Leverantörsoberoende.
- Betona långsiktighet och stabilitet
- Ta till vara erfarenheter från SSEK 1.1, bland annat:
  - Mer formell specifikation
  - Omsändning
  - Asynkront meddelandeflöde med hämtning

SSEK

- Grundläggande specifikationer
- Meddelandestrukturer
- Meddelandeflöden
- Säkerhet
- Metadata

Grundläggande specifikationer

- Baseras på etablerade standards:
  - XML, XML schema, SOAP 1.1, HTTPS, WSDL, PKI, WS-I Basic Profile, WS-I Basic Security Profile, WS-Security 1.0/1.1, WS-Policy, MTOM med flera
- Fördel:
  - Enkelt implementera med befintliga verktyg.

Meddelandestrukturer

- SSEK-header (TxHeader i v1.1)
- Standardiserade felkoder
- Standardiserat felmeddelande
- Standardkvitto
- Bilagor med MTOM

SSEK-headern

- Soap-header element

- SenderId – avsändande organisation
- ReceiverId – mottagande organisation
- TxId – eventuellt meddelandeflöde meddelandet ingår i
- AsynchMethod – begärd asynkron meddelandeflödestyp

- Organisationer kan identifieras på flera sätt, viket styrs av attributet Type på SenderId och ReceiverId.

Exempel på SSEK-header

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
 <soap:Header>
  <ssek:SSEK xmlns:ssek="http://schemas.ssek.org/ssek/2006-05-10/"
          ssek:AsynchMethod="AsynchPush" soap:mustUnderstand="1">
   <ssek:SenderId ssek:Type="CN">Mäklare</ssek:SenderId>
   <ssek:ReceiverId>Skandia Liv</ssek:ReceiverId>
   <ssek:TxId>c615da82-e3d1-4e82-9520-938b9a0568b9</ssek:TxId>
  </ssek:SSEK>
 </soap:Header>
 <soap:Body>
  <TEST_REQUEST xmlns="http://schemas.skandia.se/test/2006-05">
   <name>Gustaf Wasa</name>
  </TEST_REQUEST>
 </soap:Body>
</soap:Envelope>
```

SSEK-header

- SSEK-headern fyller tre syften:
  - Adressering. Innehåller information om avsändande och mottagande organisationer.
  - Identifiering av meddelandeflöde.
  - Val av asynkron metod.

SSEK adressering

- **SSEK handlar om kommunikation mellan organisationer**
  - Avsändare och mottagare i SSEK avser organisationer, inte någon specifik resurs
  - Ointressant ur affärsperspektiv vilken URL en viss tjänst finns på

- SSEK-meddelanden innehåller i sig all viktig information

Identifiering av organisationer

- Distinguished Name (DN)
  - Skall matcha organisations certifikat. Ex
    C=SE, O=Skandia Liv, OU=Skandia Liv,
    CN=Skandia Liv, L=Stockholm, S=Sverige
- Common Name (CN) - grundvärde
  - Skall matcha organisations certifikat. Ex
    Skandia Liv
- Organisationsnummer (ORGNR)
- Applikation (APP)
  - För internt bruk

- Policy kan styra på vilket sätt identifiering skall ske.

Identifiering av meddelandeflöde

- TxId identifierar ett meddelandes meddelandeflöde.
- 128-bitars automatgenererad identifierare (UUID), ex c615da82-e3d1-4e82-9520-938b9a0568b9
- Genereras TxId enligt specificerad algoritm är de alltid unika.
- Policy styr om en tjänst använder TxId.

Felhantering

- Fel returneras alltid som SoapFault
- FaultData-struktur för mer detaljerad felinformation
- Omsändning

Omsändning av meddelanden

- Omsändning definieras som att ett meddelande med samma innehåll och SSEK-header med Txid skickas på nytt.
- Omsändning är tillåten när felmeddelande mottagits med undantag för ssek:Timeout
  - Mottagaren måste backa ur alla transaktioner om felmeddelande returneras.
- Omsändning är inte tillåten när inget svarsmeddelande har mottagits (timeout). Manuell hantering istället.
- En tjänsts policy kan styra så att omsändning alltid är tillåtet.

Felkoder

- faultcode skall alltid vara ett QName
- SSEK definierar ett antal felkoder i sin namespace: http://schemas.ssek.org/ssek/2006-05-10/
- Andra felkoder finns definierade i bland annat SOAP- och WS-Security-specifikationerna.

Exempel på felmeddelande

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode xmlns:p="http://schemas.ssek.org/ssek/2006-05-10/">p:ContentInvalid</faultcode>
      <faultstring>Personnummer skall innehålla 12 tecken</faultstring>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Felkoder definierade för SSEK

- AsynchMethodUnsupported
- ContentInvalid
- IncorrectContext
- MessageNotProcessed
- NoResultAvailable
- ReceiverIdUnknown
- SenderIdUnknown
- Timeout
- TxIdInvalid
- TxIdMissing
- TxIdNotAllowed
- WebServiceUnavailable
- WebServiceUnsupported

FaultData

- För mer detaljerad felinformation
- FaultData placeras under detail-element och kan innehålla:
  - Det felande meddelande i sin helhet
  - Det felande meddelandets TxId
  - Detaljerade beskrivningar av ett eller flera fel (Code, Description, Location, System)

Exempel på felmeddelande med FaultData

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Body>
    <soap:Fault>
      <faultcode xmlns:p="http://schemas.ssek.org/ssek/2006-05-10/">p:ContentInvalid</faultcode>
      <faultstring>Personnummer skall innehålla 12 tecken</faultstring>
      <detail>
        <ssek:FaultData xmlns:ssek="http://schemas.ssek.org/ssek/2006-05-10/">
          <ssek:FaultingMessage>...</ssek:FaultingMessage>
          <ssek:TxId>7820EAE1-31CE-4648-B70B-A3C065005E17</ssek:TxId>
          <ssek:FaultItems>
            <ssek:FaultItem xmlns:p="http://schemas.skandia.se/faults/2006-05-31/">
              <ssek:Code>p:IllegalPnr</ssek:Code>
              <ssek:Description></ssek:Description>
              <ssek:Location xmlns:m="http://schemas.skandia.se/engagemang/2006-05-31/">
                  //m:GET_ENGAGEMANG/m:PNR</ssek:Location>
            </ssek:FaultItem>
          </ssek:FaultItems>
        </ssek:FaultData>
      </detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Standardiserat kvitto

- Receipt kvitterar mottagandet av ett meddelande
- Användbart vid asynkrona meddelandeflöden.
- Vid signering returneras även det kvitterade meddelandets signatur, som även det signeras, vilket innebär att mottagaren har 'oavvisligen' mottagit avsändarens meddelande.

Exempel på standardkvitto

```
<soap:Envelope xmlns:soap="http://schemas.xmlsoap.org/soap/envelope/">
  <soap:Header>
    <ssek:SSEK xmlns:ssek="http://schemas.ssek.org/ssek/2006-05-10/" soap:mustUnderstand="1">
      <ssek:SenderId>Skandia Liv</ssek:SenderId>
      <ssek:ReceiverId>Mäklare</ssek:ReceiverId>
      <ssek:TxId>c615da82-e3d1-4e82-9520-938b9a0568b9</ssek:TxId>
    </ssek:SSEK>
    <wsse:Security xmlns:wsse="....">...</wsse:Security>
  </soap:Header>
  <soap:Body>
    <ssek:Receipt xmlns:ssek="http://schemas.ssek.org/ssek/2006-05-10/">
      <ssek:ResponseCode>OK</ssek:ResponseCode>
      <ssek:ResponseMessage>Thank you, please call again</ssek:ResponseMessage>
      <ssek:RequestSignatureValue>KBxr9Fchqie.....qv1tiVtEzTObUdI=</ssek:RequestSignatureValue>
    </ssek:Receipt>
  </soap:Body>
</soap:Envelope>
```

Signaturer

- All väsentlig information signeras
  - SSEK-header
  - Timestamp
  - Eventuell SignatureConfirmation
  - SoapBody
- Certifikat bifogas meddelandet
- Signering sker enligt OASIS-standard: Web Services Security: SOAP Message Security 1.0/1.1
- Policy styr om en tjänst använder signaturer.

- Arkivering

Signera/verifiera meddelande

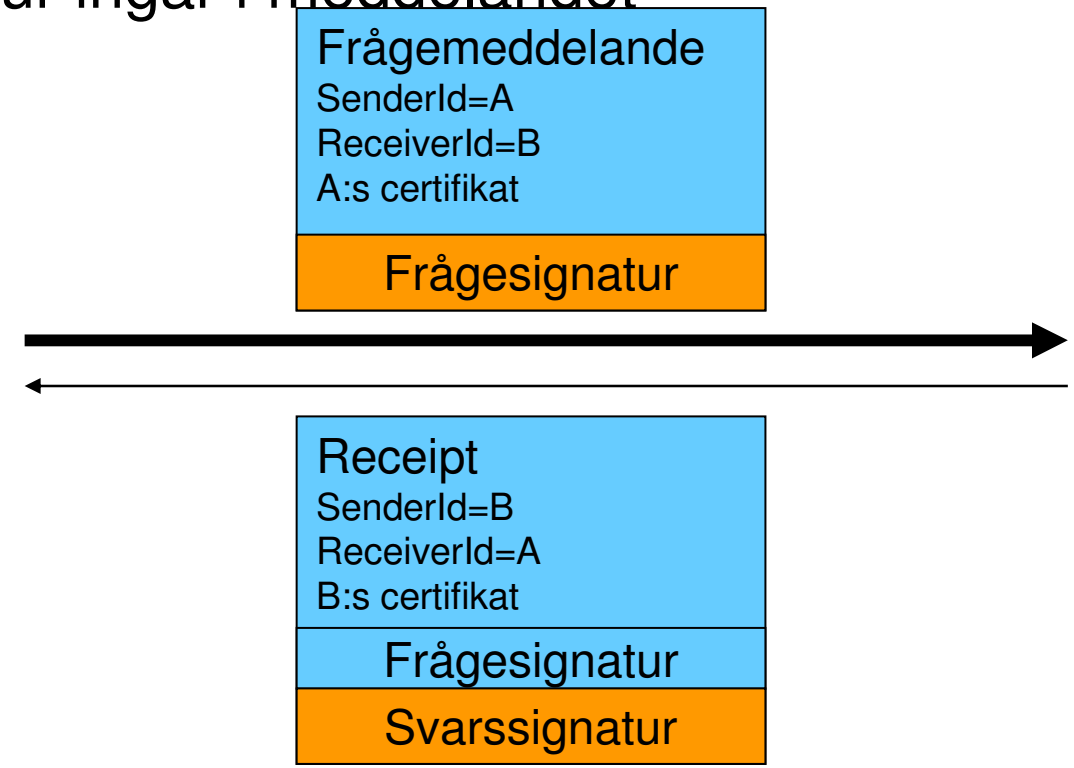
Signering av meddelande

1. Konvertera XML till kanonisk form (canonicalization).
2. Beräkna hashvärden på de delar som skall signeras
3. Beräkna hashvärde på beräknade hashvärden
4. Skapa signatur genom att transformera hashvärde med privat nyckel som hör till certifikat
5. Lagra signatur och certifikat i meddelandet

Verifiering sker med steg 1-3 varefter publik nyckel i bifogat certifikat används för att transformera meddelandets signatur som därefter jämförs med beräknat hashvärde

Signerade meddelandeflöden

- Om signering används så signeras både fråge- och svarsmeddelanden
- Signerat kvitto returneras där frågemeddelandets signatur ingår i meddelandet



Signering av frågemeddelandets signatur

- Två alternativ
  - RequestSignatureValue (från SSEK 1.1)
  - SignatureConfirmation (från WS-Security 1.1)
- Policy styr vilka varianter en tjänst stöder
- RequestSignatureHandling kan vara en av:
  - Receipt
  - SignatureConfirmation
  - ReceiptAndSignatureConfirmation

SSEK-Policy

- Per tjänst kan anges:
  - IdType (CN, DN, ORGNR. APP)
  - TransportLevelSecurity (SSL, SSLWithClientCertificate)
  - MessageLevelSecurity (None, Signature)
  - RequestSignatureHandling (Receipt, SignatureConfirmation, ReceiptAndSignatureConfirmation)
- Per operation kan anges:
  - ResendAllowedOnTimeout
  - SupportsAsynchPull
  - SupportsAsynchPush

Vad som inte definieras i SSEK-spec

- SSEK fokuserar på protokollet
- Viktiga implementationsaspekter och best-practices:
  - Arkivering
  - Administration
  - Driftstöd
  - Infrastrukturfrågor
  - Säkerhetsaspekter som hantering av privata nycklar
  - Test- och verifieringsmiljöer

Arkivering av meddelanden

- Med arkivering ger SSEK spårbarhet över tiden.
- SSEK-meddelanden innehåller i sig all viktig information:
  - Meddelandets informationsinnehåll.
  - Avsändande organisations identitet
  - Mottagande organisations identitet
  - Avsändande organisations signatur
  - Avsändande organisations certifikat
  - Frågemeddelandets signatur vid kvittering
  - Tidpunkt för meddelandet.
  - Identitet för relaterade meddelanden.
- SSEK-meddelanden kan arkiveras utan metadata.
  - Undantag CA-certifikat och avtalsrelaterade dokument.

Framtiden för SSEK

- SSEK 2.0 finns idag!
- Troligt tillägg till SSEK 2.0 för WS-Reliable Messaging när denna spec väl är fastställd. Oasis har draft idag.
- Visst arbete kring certifikat.
- Diskussionsforum kommer.

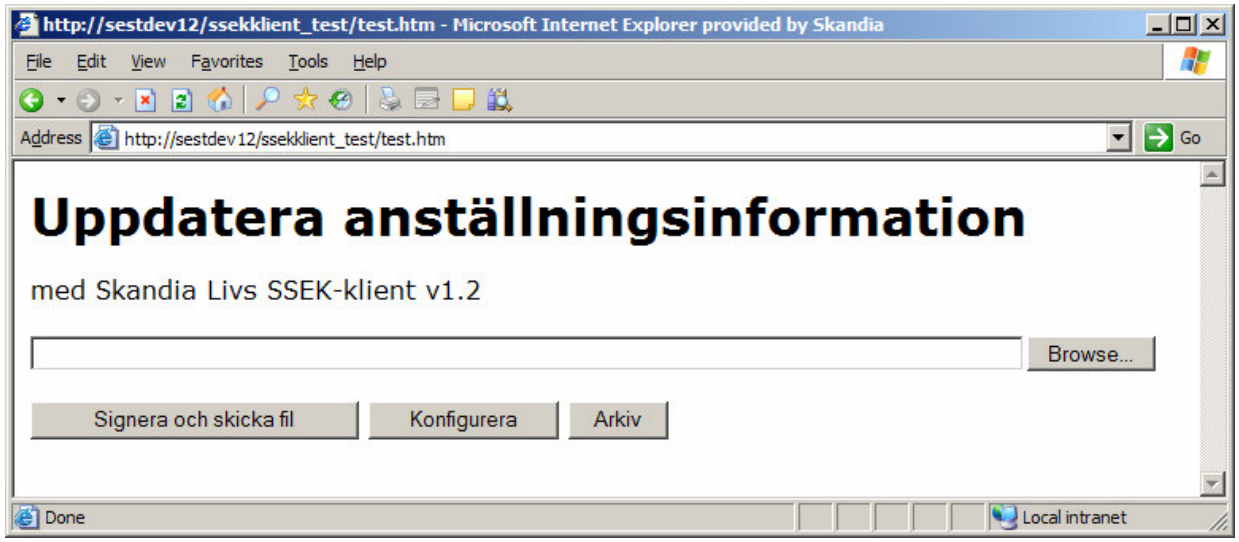
- Hur vill ni att SSEK skall utvecklas vidare?
- Vilka behov ser ni?

Arkitekturer för SSEK

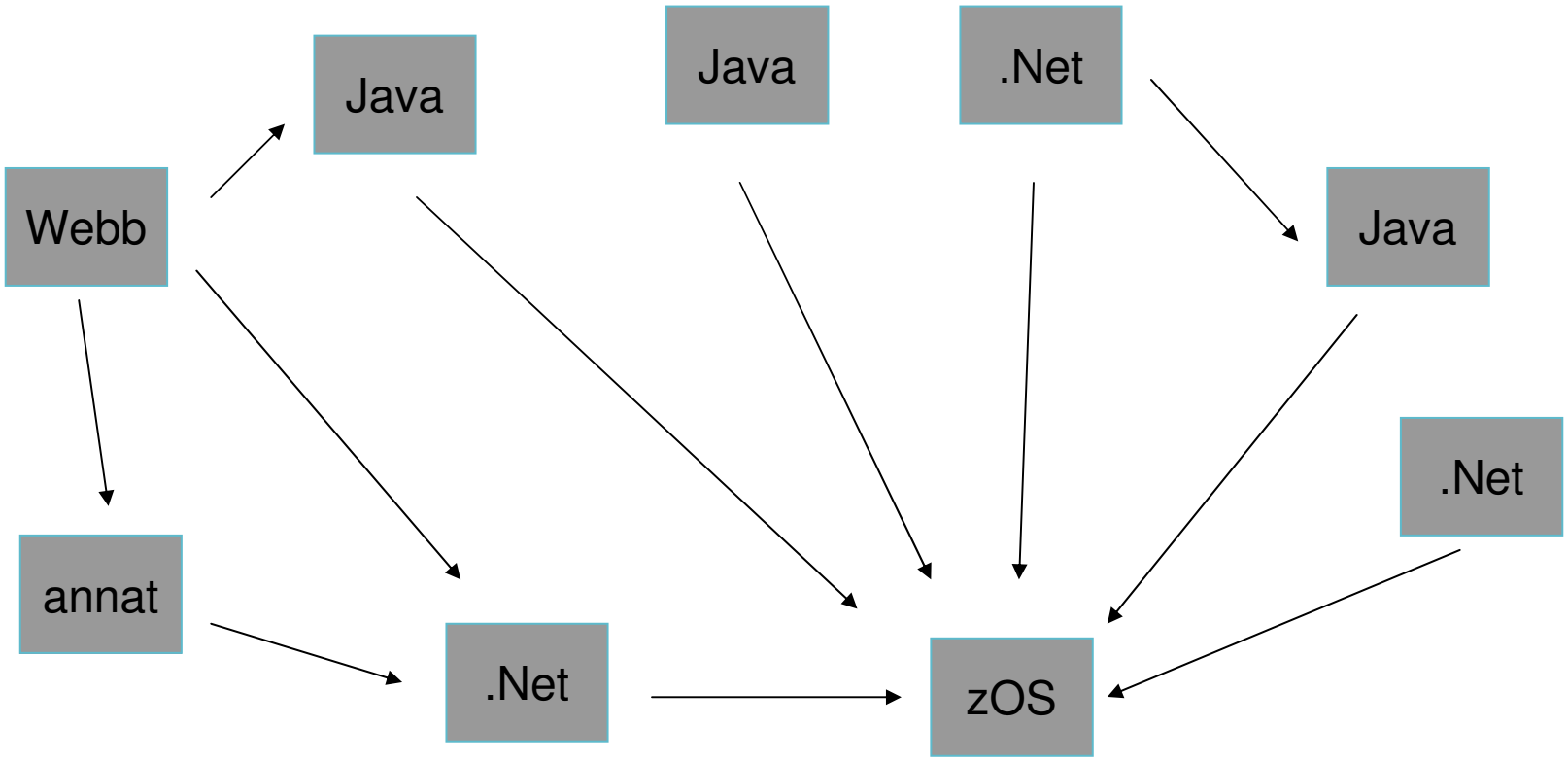
- SSEK i webbläsare
- SSEK för enstaka tjänst
- SSEK för extern kommunikation
- SSEK för extern och intern kommunikation

SSEK-konsument i webbläsare

- 'Upload' med SSEK-funktionalitet. Signering sker i webbläsare. Krav:
  - Certifikat med privata nycklar installerade på klient
  - Webbläsare som kan exekvera komponenter
- Alternativt skapa XML från webbformulär som skickas på samma sätt.
- Asynkront meddelandeflöde med hämtning ger full funktionalitet

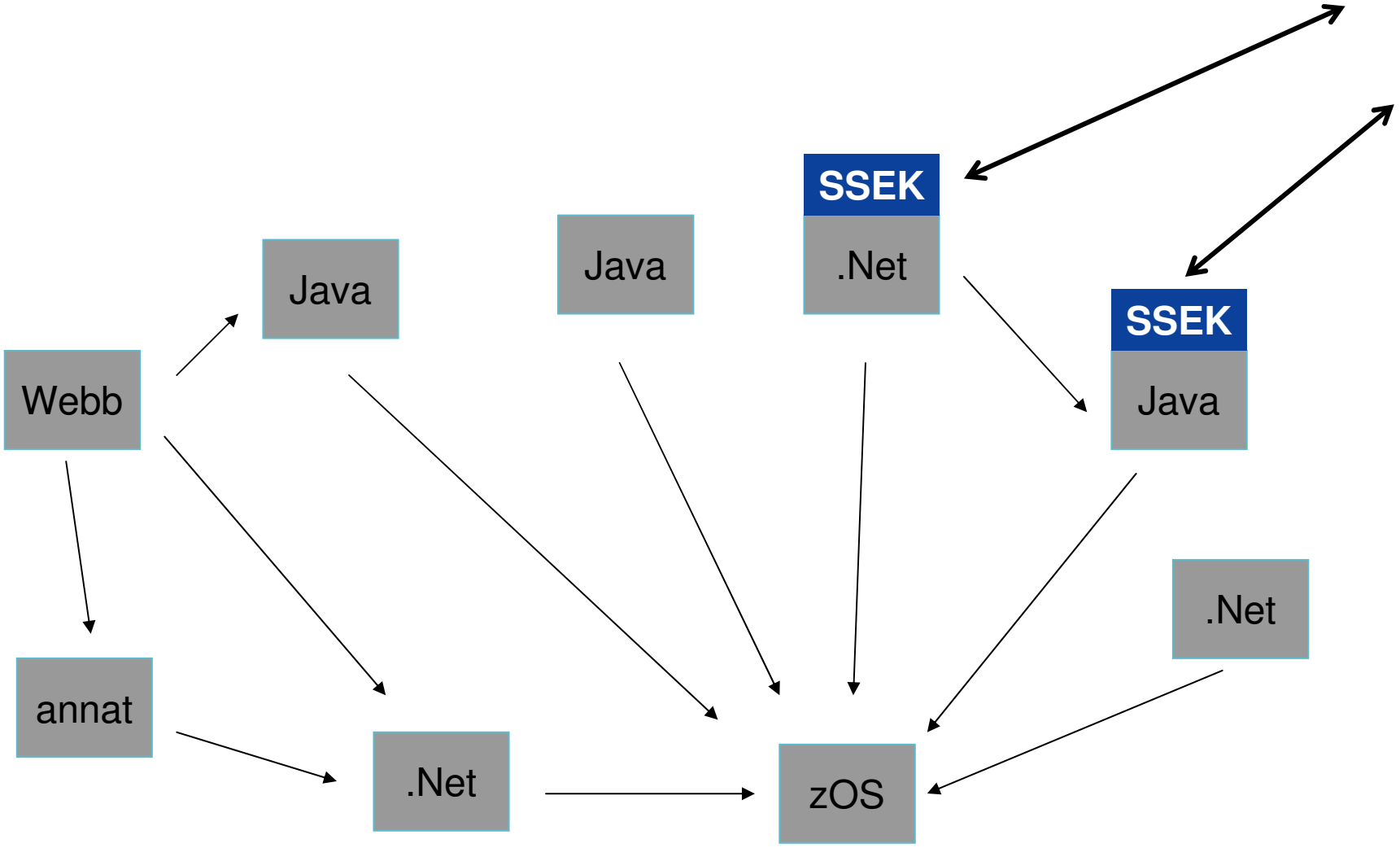


Typisk arkitektur



Enklare SSEK-arkitektur

SSEK 2.0



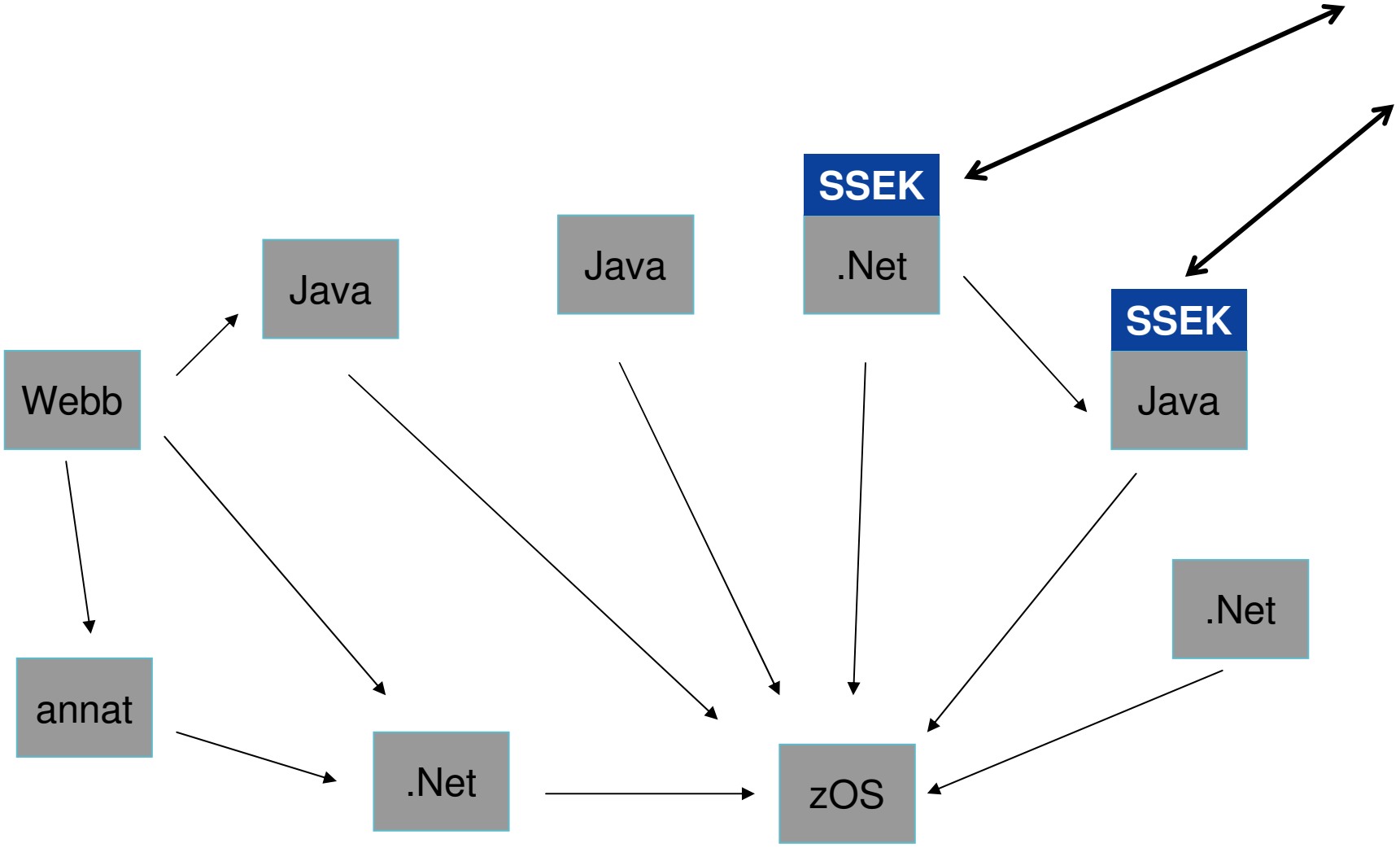
SSEK för enstaka tjänst

- Utveckla klienten eller tjänsten i ditt favoritverktyg
- Använd SSEK-toolkit för exempelvis Axis, WCF eller WSE3.0.

- Fördelar:
  - Snabbt komma igång med SSEK för användning mot affärspartners utan större investeringar.
- Nackdelar
  - Inget generellt stöd för arkivering, felsökning etc.
  - Enskilda system måste ha kunskap om mottagares certifikat, urler etc.
  - Inga synergieffekter av SSEK.

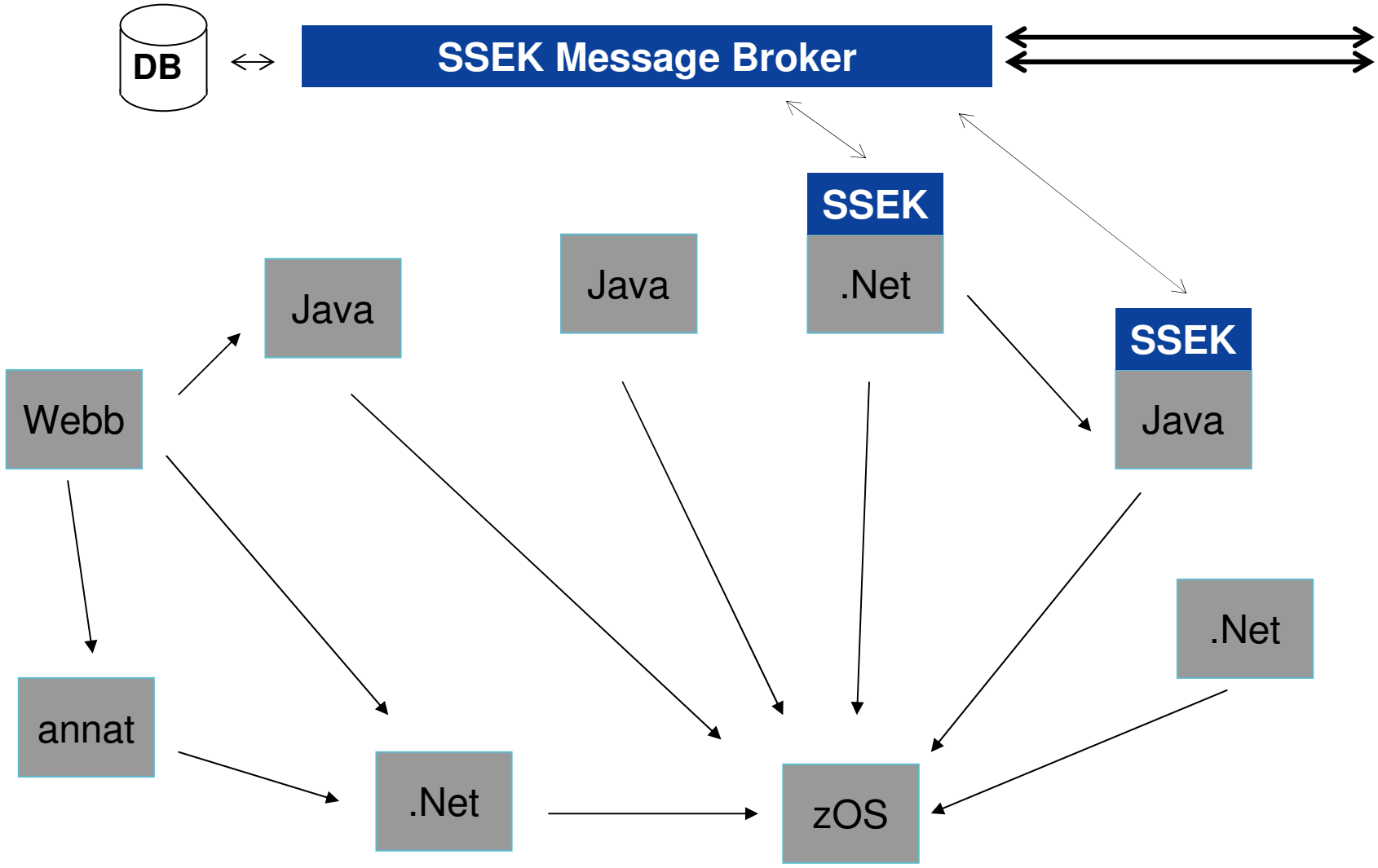
Enklare SSEK-arkitektur

SSEK 2.0



Extern SSEK-arkitektur

SSEK 2.0



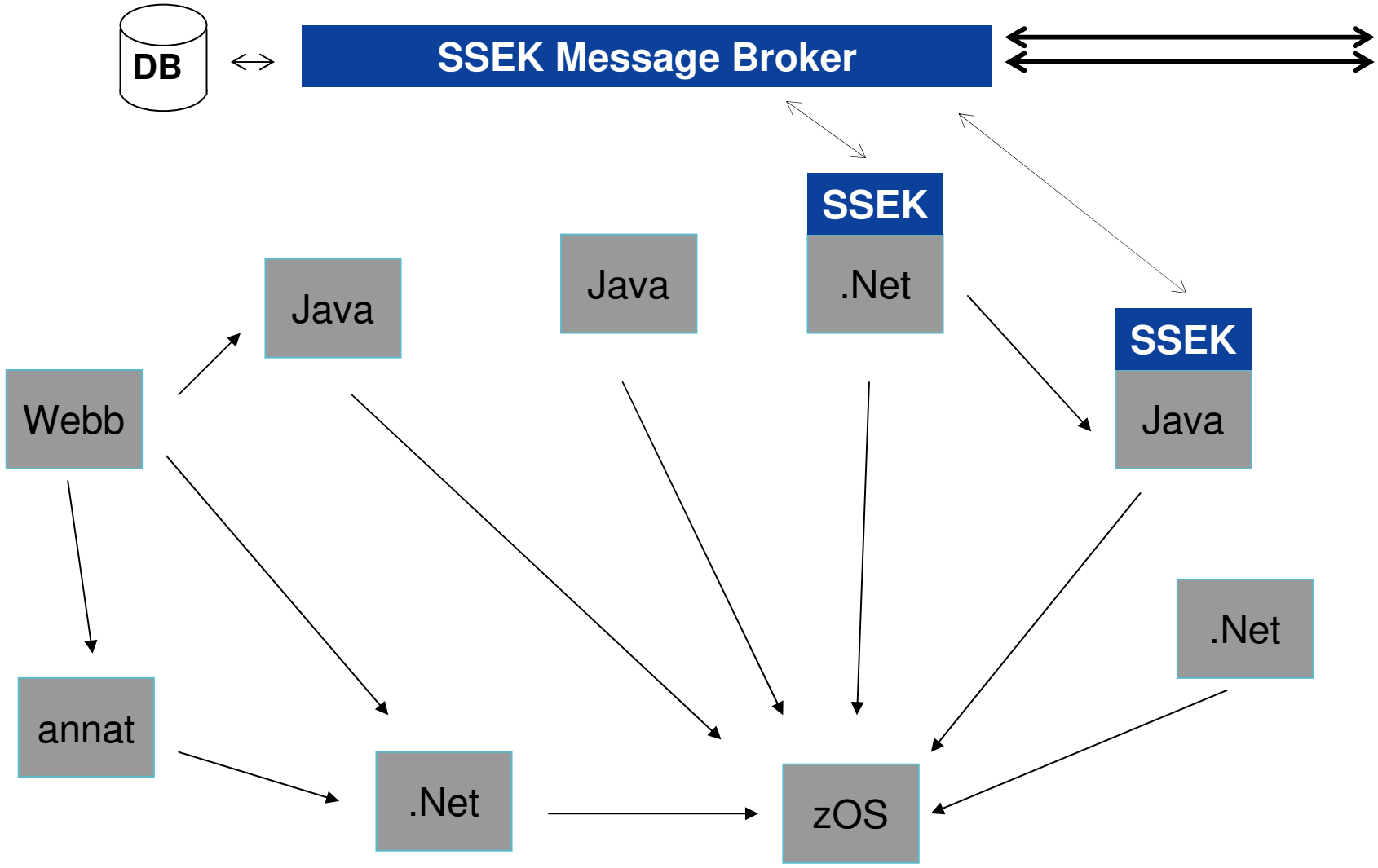
Använda SSEK externt

- En ingång för all SSEK-kommunikation.
- Signering, arkivering och loggning sker på ett ställe
- Kan outsourcas.

- Fördelar:
  - Komplett stöd för SSEK.
  - Enskilda system behöver endast ha kunskap om mottagande organisations identitet.
- Nackdelar
  - Initialt större investering

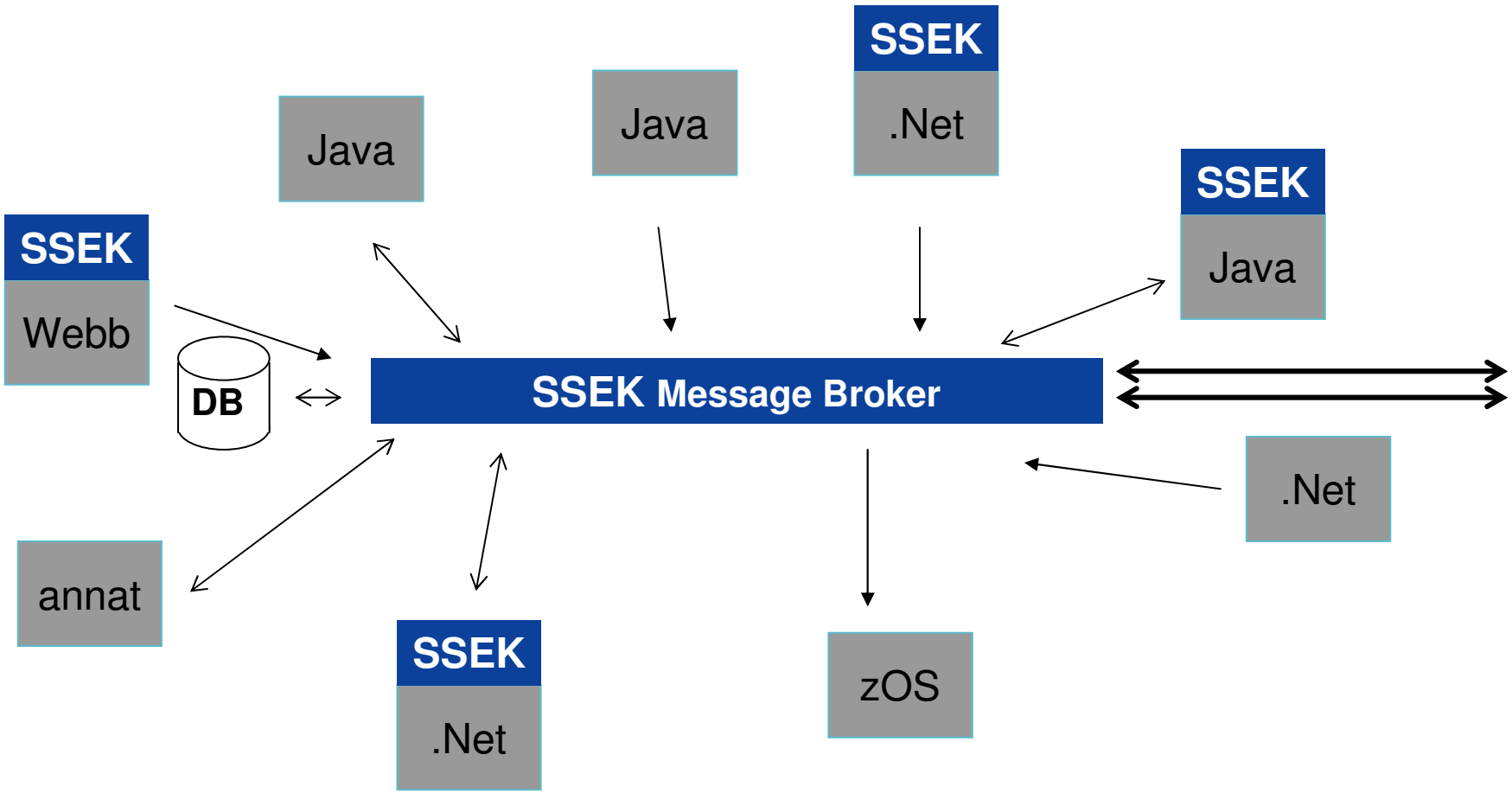
Extern SSEK-arkitektur

SSEK 2.0



Integrerad SSEK-arkitektur

SSEK 2.0



Använda SSEK externt och internt

- Kommunikation mellan både externa och interna system sker enligt SSEK.
- Signering, arkivering och loggning sker på ett ställe

- Fördelar:
  - Maximal nytta av SSEK.
  - Förenklar intern kommunikation mellan affärssystem
  - Bra bas för SOA-arkitektur

SSEK Message Broker

- Integrering av affärssystem
- Katalogfunktion
- Signering/verifiering
- Arkivering
- Loggning
- En administrationspunkt för tjänster, system och organisationer

SSEK Message Broker

- Exempel på administrativa funktioner:
  - Skapa tjänster genom import av WSDL-filer inklusive SSEK-policy.
  - Konfigurering av tillåtna avsändare/mottagare
  - Starta/stoppa tjänster
  - Söka i arkiv efter meddelanden från viss avsändare
  - Granska loggar, tidmätningar

- Mer att tänka på
  - Test- och verifieringsmiljöer med testcertifikat etc
  - Lastbalansering, redundans

Skandia Liv - erfarenheter

- Skandia Liv använder SSEK för extern och intern kommunikation sedan 3 år.
- Alla webservice-anrop (även till stordator) passerar genom SSEK Message Broker.
- Viktigt med låg fördröjning i systemet (5-45 ms)
- Nödvändigt med funktioner för loggning och felsökning
- Administrativt gränssnitt med integrerad behörighet och audit-funktion ger skalbarhet i en stor organisation.

- Slutsats: SSEK mycket bra bas för SOA

SSEK

Frågor om teknik och arkitektur?

Öppen diskussion följer efter 10 minuters paus.