
Elmarknadshubben – API-teknik



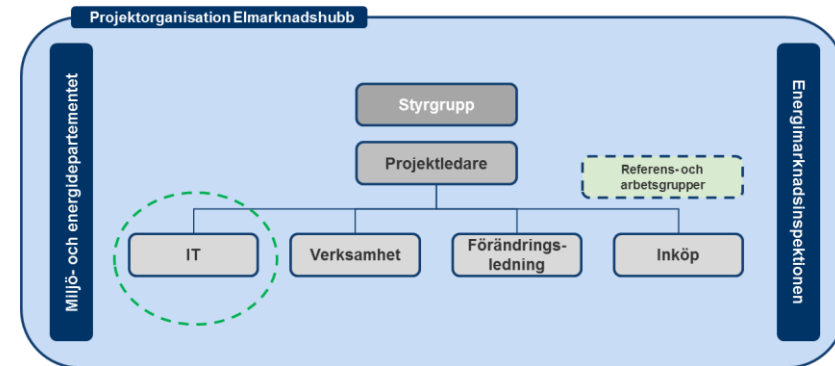
SVENSKA
KRAFTNÄT

Syfte med presentationen?

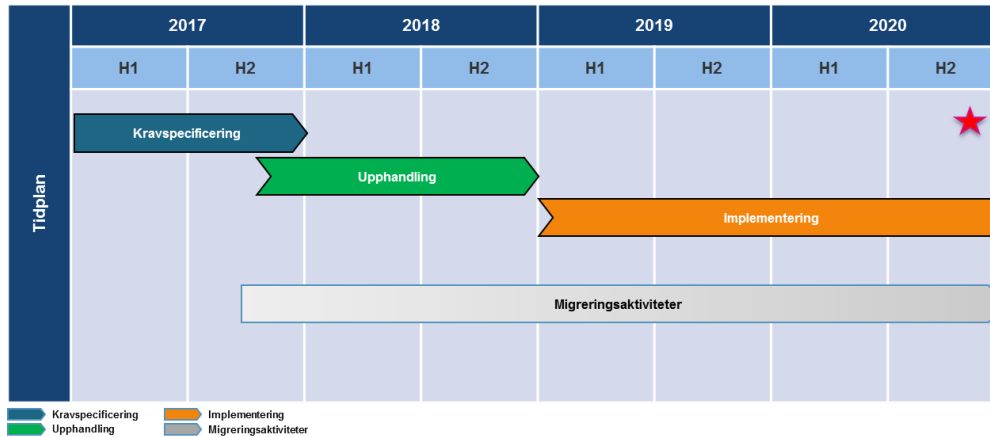
- > Informera om möjlig API-teknik för Elmarknadshubben...
- > ...i syfte att få er feedback
- > Grund för **ett** av många kommande arkitekturbeslut kring Elmarknadshubben

Vi som är här idag...

- > Alexander Gunnerhell, IT-arkitekt
- > Jimmy Petersson, Integrations-/lösningsarkitekt
- > Oscar Ludwigs, Verksamhetsexpert
- > Jan Owe, Verksamhetsexpert



Projektets tekniska status?



- > Etablering av tekniskt team under 2017: pågår
- > Icke-funktionella krav: pågår
- > Software Architecture Description/Systemspekifikation (SAD), ej påbörjad, men diverse förstudier pågår:
 - > Analys av olika arkitekturstrategier
 - > API-teknik
 - > PoC för API
 - > Analys av säkerhetskrav

Förutsättningar för analys av API-teknologi 1/2

- > Diskussioner och utredningar kring Elmarknadshubben har pågått under flera år, men...
- > ...föregående API-utredning pekar på "mainstream"-teknik, snarare än domänspecifik teknik, tex Mades
- > Två teknikstackar analyserade i denna utredning
- > Elmarknadshubben kommer ha karaktär av semi-publikt API => instegströskeln för att använda den måste vara låg

Förutsättningar för analys av API-teknologi 2/2

> API:et måste ha/vara:

> Transportprotokoll: http(s) 1.1/2

https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol,
<https://en.wikipedia.org/wiki/HTTP/2>



> Meddelandeformat: XML eller JSON

(<https://en.wikipedia.org/wiki/XML>, <https://en.wikipedia.org/wiki/JSON>)

<xml />



> Semantiskt kompatibelt med ebix (www.ebix.org)

> API:et förutsätts skyddas av en API-gateway!

(https://en.wikipedia.org/wiki/API_management)

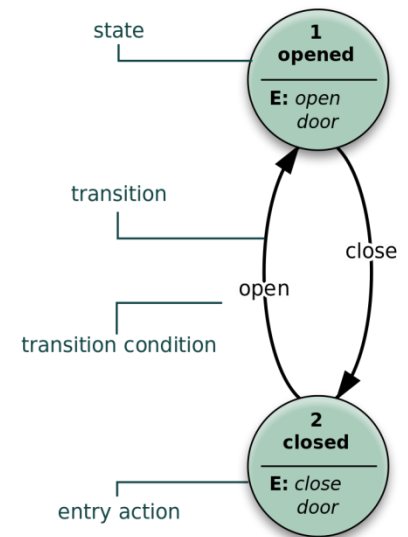
Vilka är teknikstackarna 1/2 ?

> JSON et al:

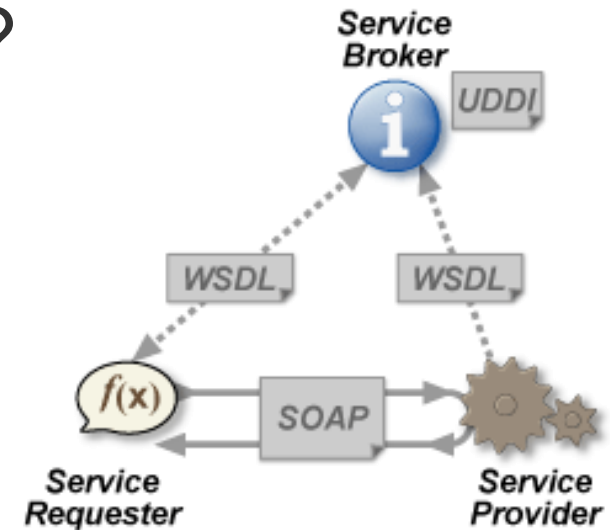
> Nån som känner att de inte har koll på JSON?

> JSON och I-JSON som meddelandeformat
(<https://www.rfc-editor.org/info/rfc7493>)

> REST och HATEOAS som designmönster
(https://en.wikipedia.org/wiki/Representational_state_transfer,
<https://en.wikipedia.org/wiki/HATEOAS>)



Vilka är teknikstackarna 2/2 ?



> Web Services:

- > Nån som känner att de inte har koll på Web Services?
 - > XML och SOAP (<https://en.wikipedia.org/wiki/SOAP>) som meddelandeformat
 - > WSDL (https://en.wikipedia.org/wiki/Web_Services_Description_Language)
 - > UDDI (om behov finns)
(https://en.wikipedia.org/wiki/Web_Services_Discovery#Universal_Description_Discovery_and_Integration)
 - > REST och HATEOAS som designmönster
(https://en.wikipedia.org/wiki/Representational_state_transfer,
<https://en.wikipedia.org/wiki/HATEOAS>)
-
- > WS-I Basic Profile Version 2.0 (https://en.wikipedia.org/wiki/WS-I_Basic_Profile)

Analyskriterier 1/7 – Standardkonformans

> JSON et al standardiserat av:

> IETF 
I E T F

> ECMA 

> => **Hög standardkonformans**

> Web Services standardiserat av:

> IETF 
I E T F

> W3C 

> OASIS 

> => **Hög standardkonformans**

Analyskriterier 2/7 – Mognadsgrad

> JSON et al:

- > enklare produktionstillämpningar runt 2005
- > sedan dess en mycket stor spridning
- > mer eller mindre "best-practice" för publika API:er
- > => **Medel mognadsgrad**

> Web Services

- > enklare produktionstillämpningar runt 2000
- > sedan dess en mycket stor spridning
- > mer eller mindre "best-practice" för EAI- och B2B-tillämpningar
- > => **Hög mognadsgrad**

Analyskriterier 3/7 – Integrationskomplexitet

> JSON et al:

- > text skickas fram och tillbaka över http!
- > kräver inga formella kontraktsdefinitioner, **men**
- > meddelanden måste **ändå** valideras syntaktiskt och semantiskt någon annanstans i Elmarknadshubben
- > => **Låg integrationskomplexitet**

> Web Services:

- > Strukturerad och typad text (XML) skickas fram och tillbaka över http!
- > kräver formella kontraktsdefinitioner (WSDL:er)
- > => **Medel integrationskomplexitet**

Analyskriterier 4/7 – Featureset

- > JSON et al:
 - > separerar ”kontrakt” och implementation genom [URL:er](#)
 - > i övrigt finns inga features alls standardiserade
 - > däremot finns mängder av features utanför standarden
 - > => **Litet featureset**

 - > Web Services:
 - > servicediscovery/katalogtjänst finns
 - > separering av kontrakt och implementation finns
 - > maskinbaserad validering av kontrakt finns
 - > => **Stort featureset**
-

Analyskriterier 5/7 – Säkerhetsegenskaper

- > JSON et al:

- > Basal autentisering och god kryptering på transportnivå (https/TLS), men:
- > PKI-infrastruktur och auktorisering saknas i stacken
- > => **Låg säkerhet**

- > Web Services:

- > Basal autentisering och god kryptering på transportnivå (https/TLS), men:
- > PKI-infrastruktur och auktorisering saknas i stacken (ingår ej i Basic Profile 2.0)
- > => **Låg säkerhet**

Analyskriterier 6/7 – Prestanda

> JSON et al:

- > Mycket litet overhead och därmed hög prestanda, skalbarhet beror av design snarare än teknologin
- > => **Hög prestanda**

> Web Services:

- > Relativt hög overhead och lägre teoretisk maxprestanda, skalbarhet beror av design snarare än teknologin
- > I praktiken är prestandan sannolikt tillräckligt god för Elmarknadshubben
- > => **(Tillräckligt) hög prestanda**

Analyskriterier 7/7 – Framtidssäkring

> JSON et al:

- > Fortsatt kraftig ökning av marknadspenetration mellan 2020-2030
- > Fortsatt kraftig vidareutveckling av teknikstacken mellan 2020-2030
- > Hög tillgång och hög kvalitet på kompetenser mellan 2020-2030
- > => **Hög framtidssäkring**

> Web Services:

- > Marknadspenetration konstant eller minskande mellan 2020-2030
 - > Fullmogen teknikstack, ingen signifikant vidareutveckling mellan 2020-2030
 - > Hög tillgång på kompetenser men med avtagande kvalitet mellan 2020-2030
 - > => **Medel framtidssäkring**
-

Slutsatser

Analyskriterie	Web Services	"JSON et al"
Standardkonformans	HÖG	HÖG
Mognadsgrad	HÖG	MEDEL
Integrationskomplexitet	MEDEL	LÅG
Featureset	HÖG	LÅG
Säkerhetsegenskaper	LÅG	LÅG
Prestanda, skalbarhet och last	HÖG	HÖG
Framtidssäkring	MEDEL	HÖG

Att fundera på tills nästa gång...

- > Finns det någon annan teknologi ni hellre skulle se i Elmarknadshubbens API? Varför är den lämpligare för er?
- > Är analyskriterierna relevanta? Saknas några kriterier som ni tycker borde vara med?
- > Givet de två kandidatteknologierna, vilken skulle passa er organisation bättre? Varför då?
- > Tänk på att era system skall inte bara anropa hubbens API-tjänster utan ni måste implementera API-tjänster som hubben skall anropa (främst notifieringar)!

Appendix – Crash course JSON och XML

- > JSON: <https://www.youtube.com/watch?v=sSL2to7Jg5g&t=6s>
- > Web Services:
https://www.youtube.com/watch?v=ip7_X2DQB54