Från Open Metadir till Open Metadir 2

Presentation under Swami-dagarna 2006

roland.hedberg@adm.umu.se

Historik

- Började på flygplatsen i Denver Co. USA, juni 2004
- KK1.0 driftsattes maj 2005
- KK2.0 driftsattes mars 2006
- KK2.1 driftsattes november 2006
- KK3.0 i drift hösten 2007

Open Metadir

Några av de problem vi försökt lösa

- Anställda och studenter kommer och går, alla är de potentiella användare av våra system så vi måste hålla bra koll på dem.
- Många källor/applikationer har information om “samma” objekt.
- I samband med olika händelser skapas information som många borde få del av.
- ‘Work-flow’.

Open Metadir

Så..

Vi behövde ett verktyg som tillät oss:

- Skicka information om händelser från en plats till en annan
- Att göra detta beroende på informationsinnehållet och metadata om informationen
- Att dynamiskt ändra “routing” tabellen

Open Metadir

För att detta skulle fungera

- Beskrivningarna av händelser måste kunna ske på ett sätt; med en syntax
- Detta krävde en ‘öppen’ syntax
- Om det fanns en ‘mager’ representation av denna syntax så var det ett stort plus
- Så, vi valde RDF och mer specifikt N-Triples formen

Open Metadir

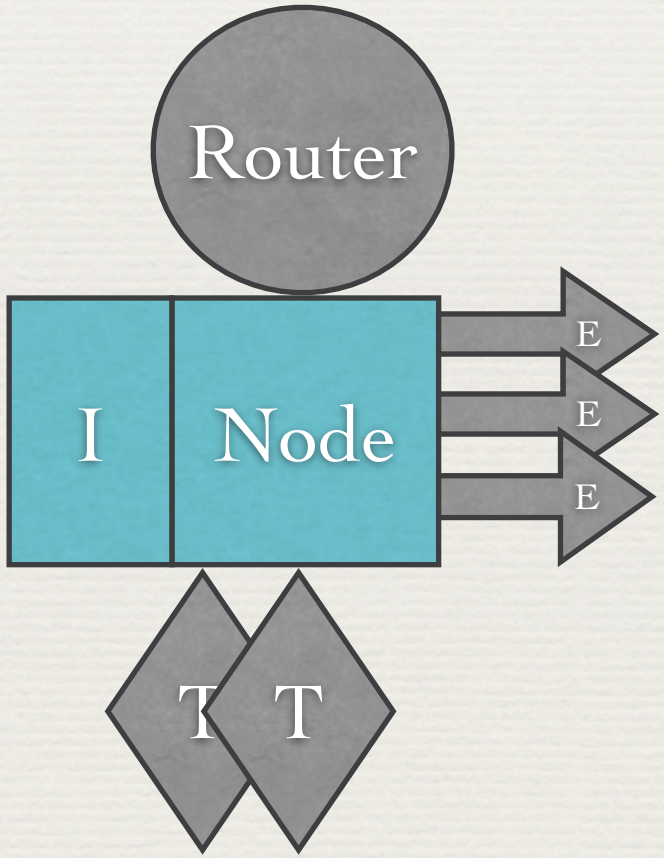
Så vad är RDF ?

- En lättviktig system för formell (maskinläsbar) beskrivning av världen (en avgränsad värld).
- Används för att utbyta kunskap över Webben.
- Baserat på XML och användningen av URI
- RSS använder RDF
- 3 olika serialiserings format N-Triples är ett av dem.

Exempel på händelse graf

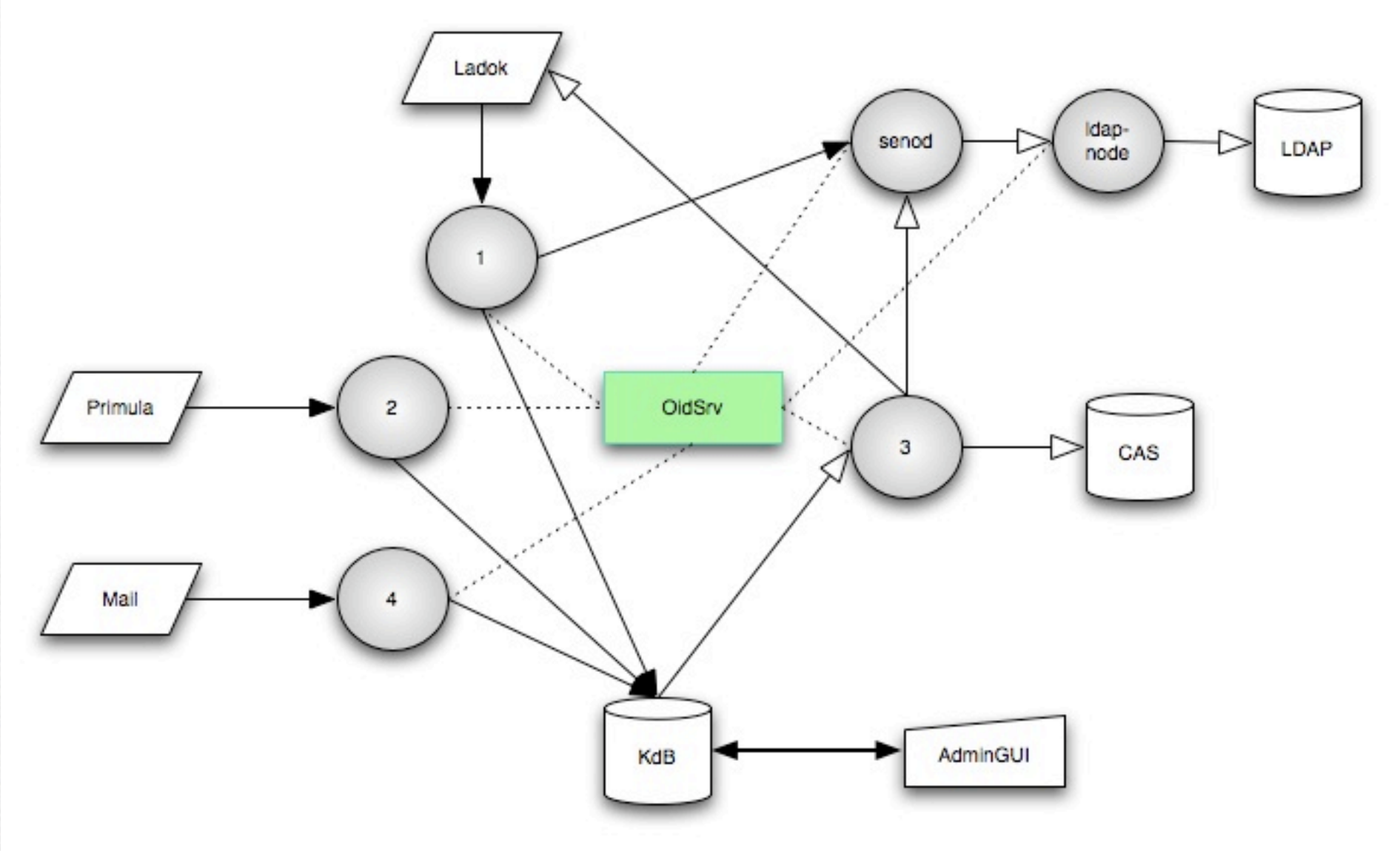
<om:add> <om:src> "primula" .
<om:add> <om:eid> "6648" .
<om:add> <om:oid> "NIN:20051012-8575" .
<om:add> <om:objecttype> "person" .
<om:add> <om:data> _:a .
_:a <omat:norEduPersonNIN> "20051012-8575" .
_:a <omat:givenName> "Per Arne" .
_:a <omat:sn> "Nilsson" .

System arkitektur - Nod = baskomponent



I=Importer
E=Exporter
T=Transformer

KK2.1@UmU


Ladok
senod
ldap-
node
LDAP
1
Primula
2
OldSrv
3
CAS
Mail
4
KdB
AdminGUI

Vad hanterar vi?

- Personer
  - Anställda, studenter, annan personal, gäster
- Organisations enheter (officiella, inofficiella)
- Kurser, kursinstanser
- Grupper (rudimentärt)

Hur många objekt?

- Aktiva studenter: 22054
- Staff: 2444
- Faculty: 1527
- Kurser: 5341
- Kursinstanser: 1966
- Officiella organisationsenheter: 114

Hur många händelser är det frågan om ?

- När vi reinitierade systemet i förra veckan fick Ldap noden ta emot ~220.000 grafer

Vad har vi lärt oss?

- Systemet gör vad vi sagt att det skall göra.
- Att formalisera information får följdeffekter.
- Inget har varit omöjligt att fixa.
- Över tid ökar konsekvenserna av förändringar.
- För- och nackdelar med informell utveckling.
- Vikten av att skilja på KK och OM.
- Utveckling/underhåll är personkrävande.

OM+J-Event => OM^2

Vad vi vill konstruera!

Några kanske hellre vill ha !

Vad det kanske blir!

Who are we ?

- Swami
- Linköpings Universitet
- Stockholms Universitet
- Umeå Universitet

Grundtankarna!

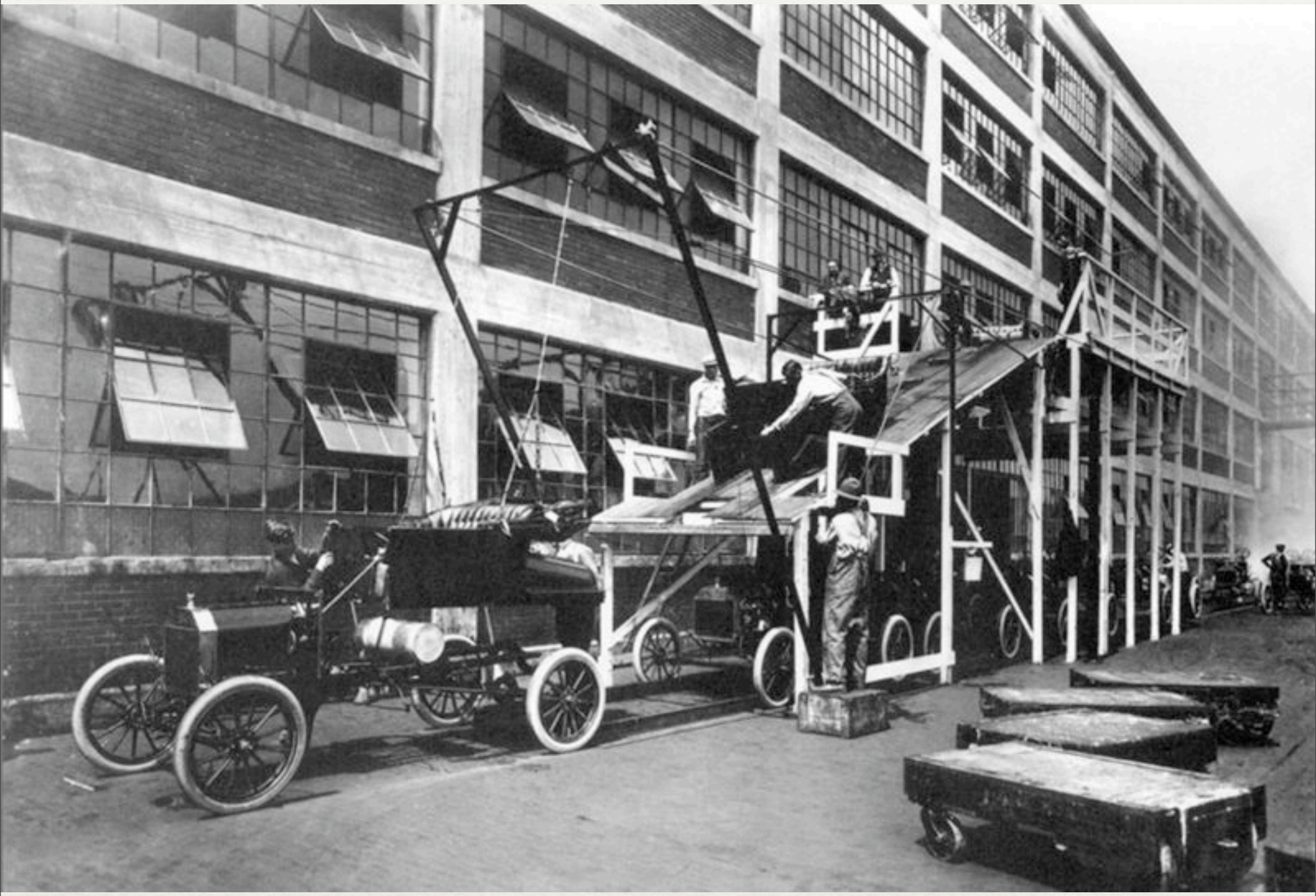
- Cecily. Do you suggest, Miss Fairfax, that I entrapped Ernest into an engagement? How dare you? This is no time for wearing the shallow mask of manners. When I see a spade I call it a spade.

- Gwendolen. [Satirically.] I am glad to say that I have never seen a spade. It is obvious that our social spheres have been widely different.

1. This is no time for wearing the shallow mask of manners
2. Dit is geen tijd voor het dragen van het ondiepe masker van manieren
3. Ce n'est pas un temps pour le port du masque superficiel des manières
4. Δεν είναι ένας χρόνος για το λιμάνι της superficiel μάσκας των τρόπων
5. It is not one year for the harbour of superficiel mask of ways

-Babels fish

- Gemensamt set av ontologier
- Vi måste vara överens om hur saker skall representeras



- Löpandeband teknik

“Neither rain, nor sleet nor snow nor dark of night shall stay this courier from his appointed rounds.”

OM2

- Har *Ontologier* för att beskriva möjliga utseenden på meddelande.
- Är *händelsestyrt* och det handlar om relativt små meddelanden
- Använder *regler* för att styra vem som får publisera vad, vem som kan få tillgång till informationen och hur.
- Kan transformera meddelanden *on-the-fly*.
- Garanterar *säker* transport.
- Transportprotokoll *agnostiskt*

Ontologies

(wikipedia)

Ontologies are used in artificial intelligence, the semantic web, software engineering and information architecture as a form of knowledge representation about the world or some part of it. Ontologies generally describe:

- **Individer**: basala eller "grundnivå" objekt
- **Klasser**: set, samlingar, eller typer av objekt
- **Attribut**: egenskaper, drag, karakteristika, eller parametrar som objekt kan ha och dela.
- **Relationer**: sätt som olika objekt kan vara relaterade till varandra

Varje källsystem sitt eget pastorat

- En ontologi per källsystem
- Om nödvändigt kan man skriva en ‘översättnings’ ontologi till en mer generell ontologi, som i så fall tillämpas i periferin.

OM ontologi (1)

- OMMessage
  - begin
  - dependentOn
  - end
  - errorTo
  - *event*
  - mid
  - receiver
  - replyTo
  - source

OM ontologi (2)

- OMEvent
  - OMAddEvent
    - (timestamp)
    - object
  - OMChangeEvent
  - OMDeleteEvent
  - OMJoinEvent

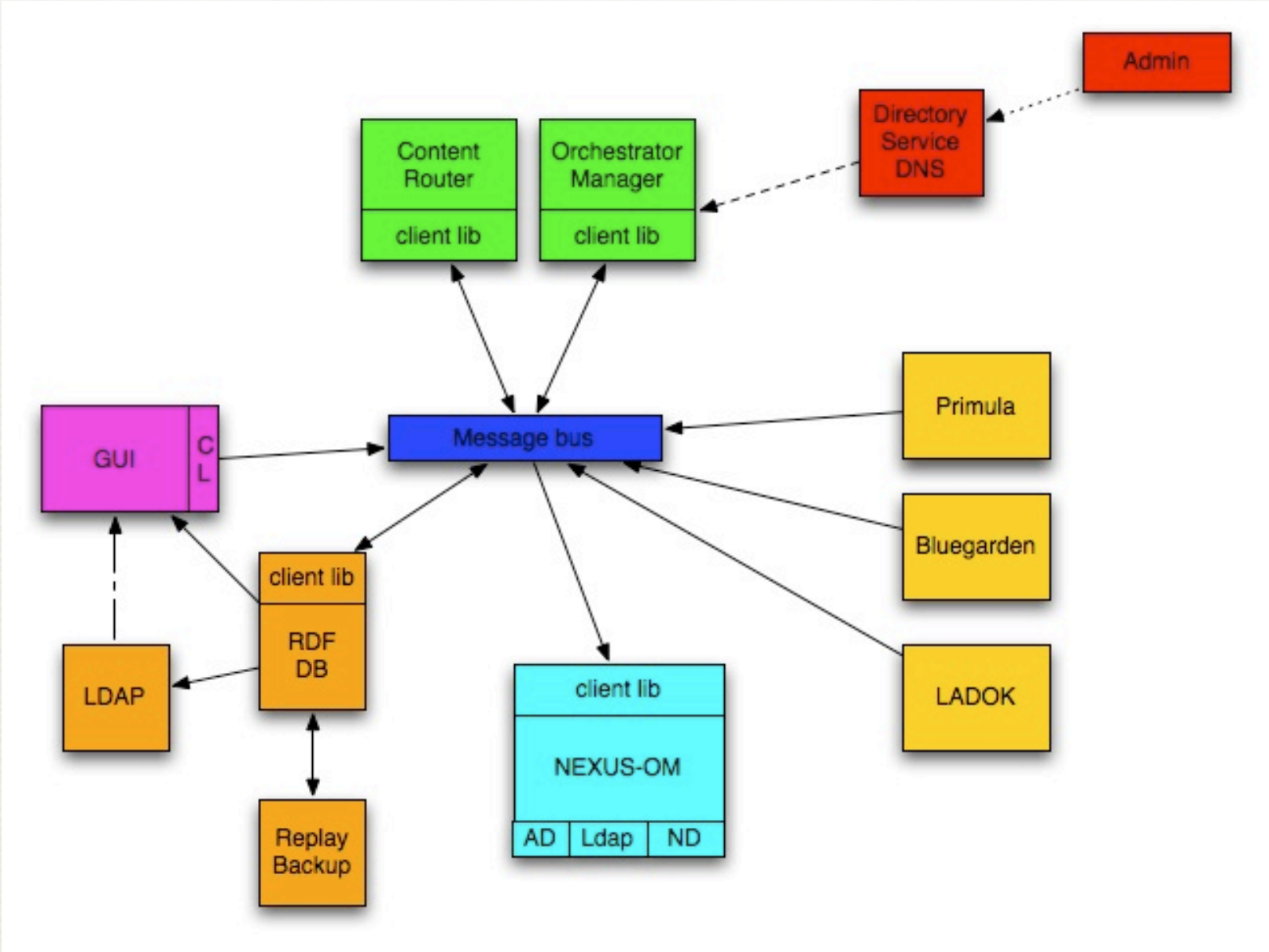
Provisioning Registry Information Model (PRIM)

- Relation
  - Owner
  - Role
- Collection
  - Group
  - Organization
- Thing
  - Person
  - User
  - Machine
  - Service
  - Course

OM2 transporterar objekt!

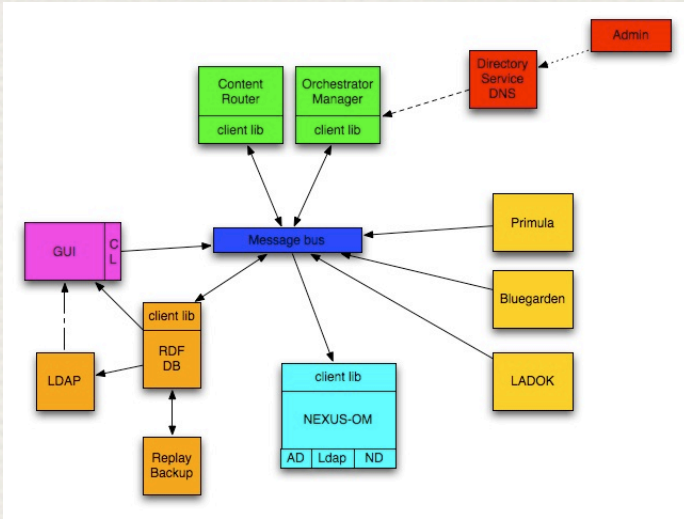
1. Källsystem A populerar ett objekt med information (java,python,perl(?)). Objektet är en språkberoende representation av en ontologi.
2. Lämnar objektet till OM2s klient bibliotek
3. Magi händer
4. Mottagare X får ett objekt

Under huven


Admin
Directory Service DNS
Content Router
client lib
Orchestrator Manager
client lib
Primula
GUI
C L
Message bus
Bluegarden
client lib
RDF DB
LDAP
client lib
NEXUS-OM
AD
Ldap
ND
LADOK
Replay Backup

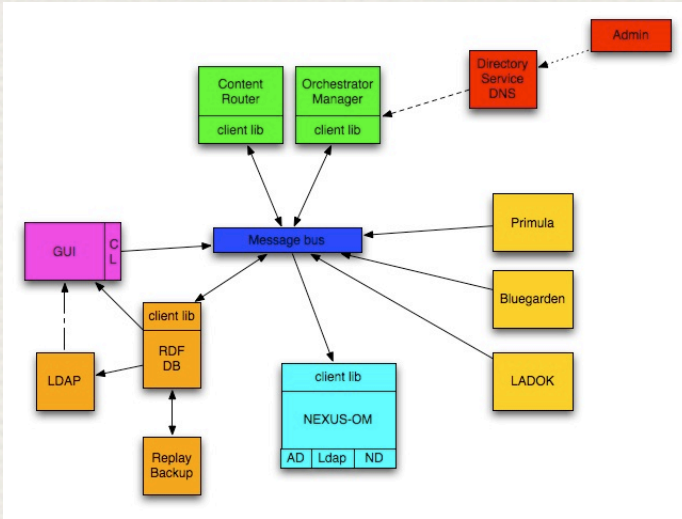
Meddelande bussen

- Kommer att stödja minst
  - SOAP/WS-RM
  - REST/HTTP
  - XMPP



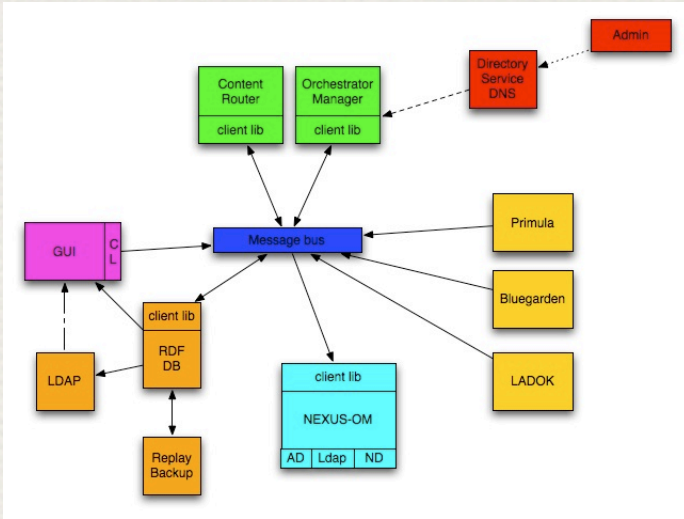
Katalog tjänst

- NAPTR baserad DNS uppslagning
- HTTP GET för att få tag på nodbeskrivning



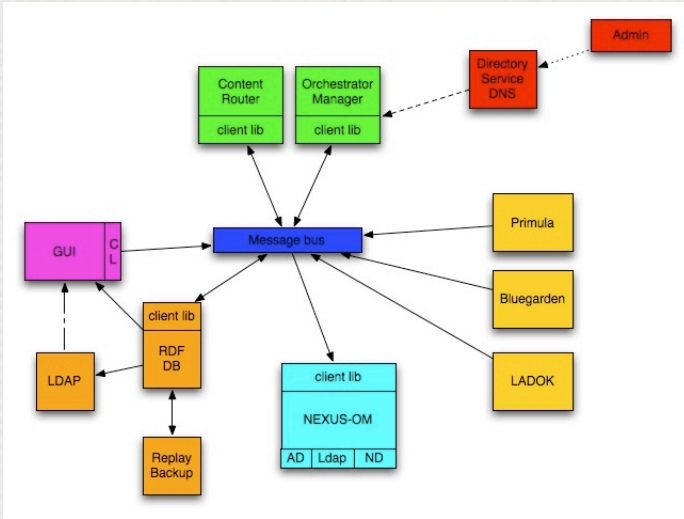
Löpande bandet

- Meddelande-baserad routing
- Operationell logik
- Transformations logik



The registry

- RDF databas
- SPARQL som frågespråk
- WS och LDAP read-only interface



RDF store

Vad är ett RDF store?

- En databas som lagrar triplar/grafer
- Har bara add och delete av tripler
- RDF's ability to assign any predicate/object pair to any subject makes it an ideal format for a freeform database
- Take two files of RDF triples written in the n-triples syntax, or in the related Notation 3, append one file to the other, and you've just merged two databases.

SPARQL Protocol and RDF Query Language (SPARQL)

A SPARQL query is a tuple (GP, DS, SM, R) where:

* GP is a graph pattern
* DS is an RDF Dataset
* SM is a set of solution modifiers
* R is a result form

```
@prefix foaf:       <http://xmlns.com/foaf/0.1/> .
@prefix rdf:        <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:       <http://www.w3.org/2000/01/rdf-schema#> .

_:a  foaf:name      "Alice".
_:a  foaf:mbox      <mailto:alice@work.example> .

_:b  foaf:name      "Ms A.".
_:b  foaf:mbox      <mailto:alice@work.example> .
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name
WHERE { ?x foaf:name  ?name }
```

---

name
Alice
Ms A.

```
@prefix foaf:       <http://xmlns.com/foaf/0.1/> .
@prefix rdf:        <http://www.w3.org/1999/02/22-rdf-syntax-ns#> .
@prefix rdfs:       <http://www.w3.org/2000/01/rdf-schema#> .

_:a  foaf:name      "Alice".
_:a  foaf:mbox      <mailto:alice@work.example> .

_:b  foaf:name      "Ms A.".
_:b  foaf:mbox      <mailto:alice@work.example> .
```

```
PREFIX foaf: <http://xmlns.com/foaf/0.1/>
SELECT ?name1 ?name2
WHERE { ?x foaf:name  ?name1 ;
           foaf:mbox  ?mbox1 .
         ?y foaf:name  ?name2 ;
           foaf:mbox  ?mbox2 .
       FILTER ( ?mbox1 = ?mbox2 && ?name1 != ?name2 )
}
```

---

name1	name2
Ms A.	Alice
Alice	Ms A.

```
@prefix  foaf:  <http://xmlns.com/foaf/0.1/> .

_:a    foaf:name   "Alice" .
_:a    foaf:knows  _:b .
_:a    foaf:knows  _:c .

_:b    foaf:name   "Bob" .

_:c    foaf:name   "Clare" .
_:c    foaf:nick   "CT" .
```

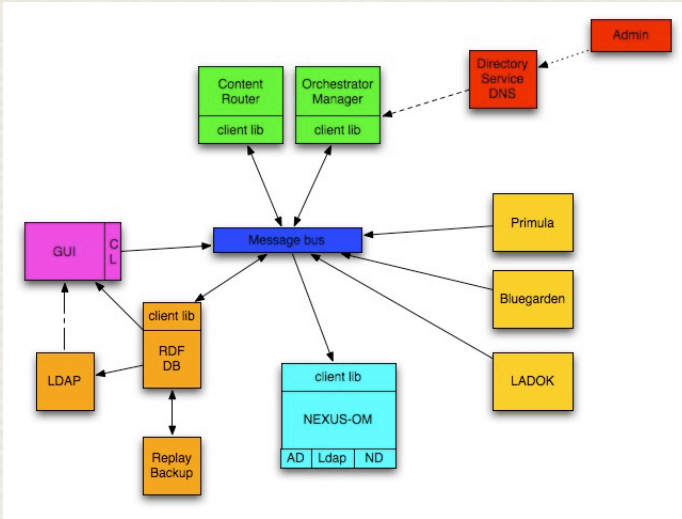
```
PREFIX foaf:    <http://xmlns.com/foaf/0.1/>
SELECT ?nameX ?nameY ?nickY
WHERE
 { ?x foaf:knows ?y ;
      foaf:name ?nameX .
   ?y foaf:name ?nameY .
   OPTIONAL { ?y foaf:nick ?nickY }
 }
```

---

nameX	nameY	nickY
"Alice"	"Bob"	
"Alice"	"Clare"	"CT"

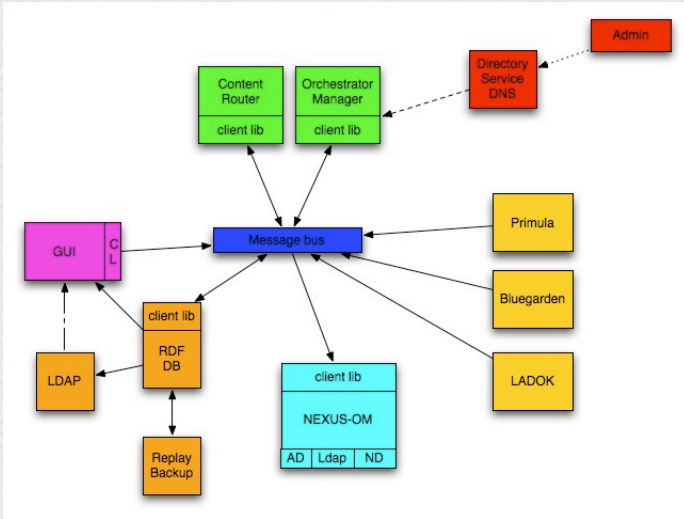
GUI't

- För interactioner mellan adminstratörer och registry't
- Ontologi baserat

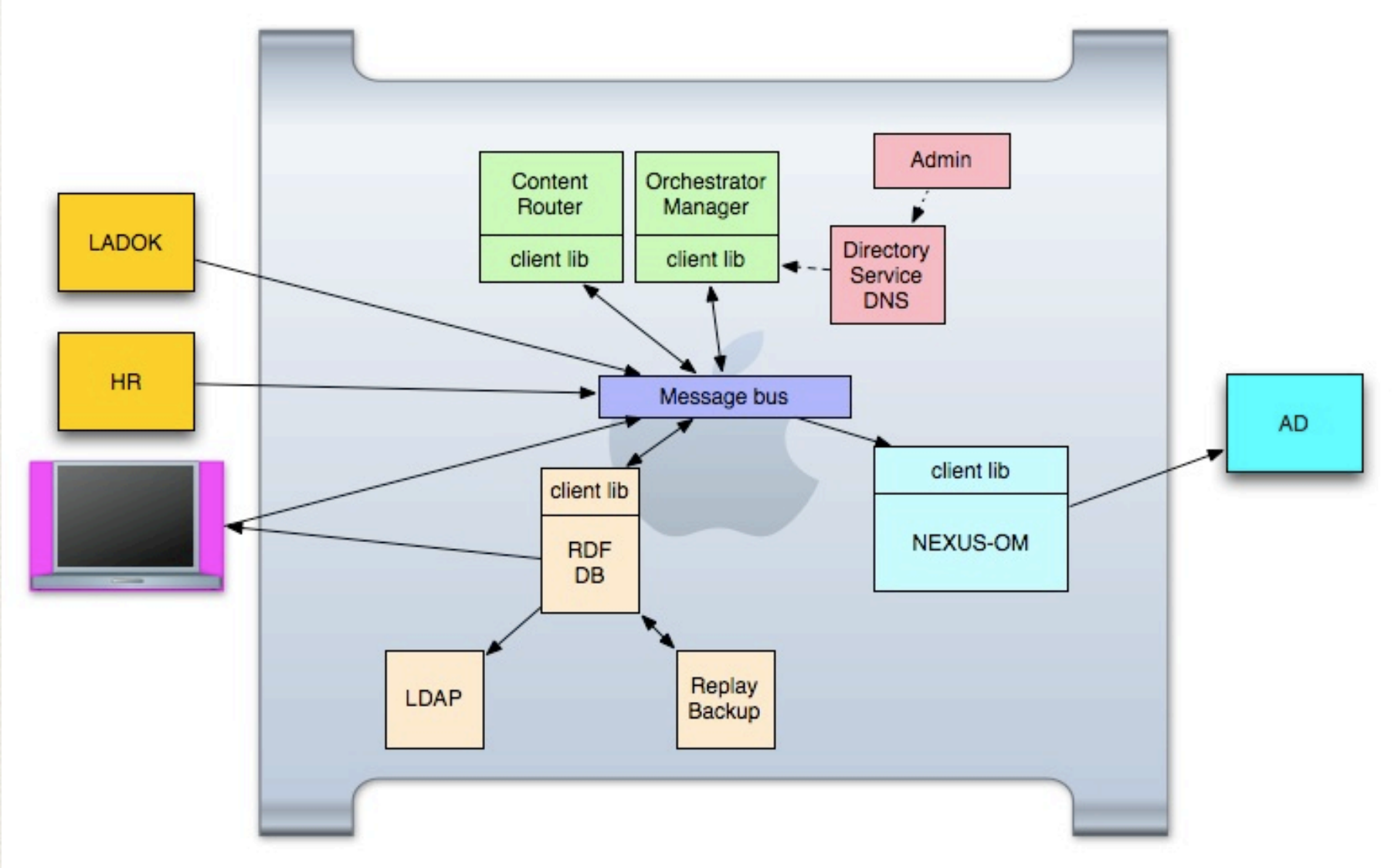


Export till LDAP

- SPML (möjligen/förmodligen) baserad provisionering



In-a-box

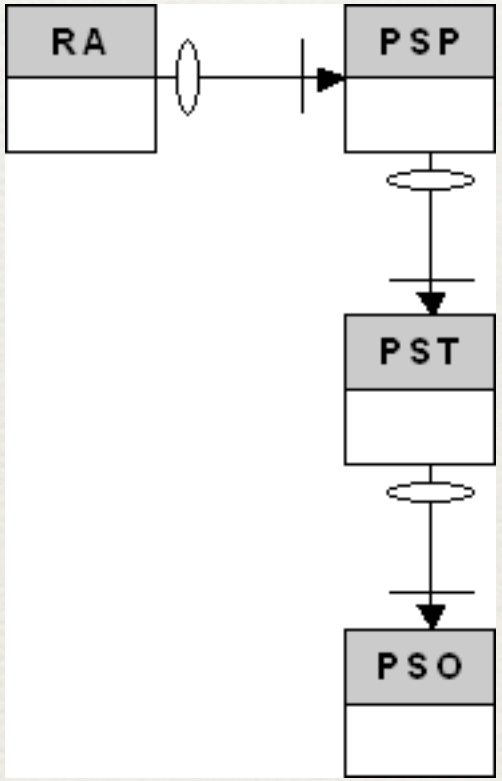

LADOK
HR
Content Router
client lib
Orchestrator Manager
client lib
Admin
Directory Service DNS
Message bus
client lib
RDF DB
LDAP
Replay Backup
client lib
NEXUS-OM
AD

Questions!

Service Provisioning Markup Language (SPML)

SPML domain model

- Requesting Authority (RA) or requestor is a software component that issues well-formed SPML requests
- Provisioning Service Provider (PSP) or provider is a software component that listens for, processes, and returns the results for well-formed SPML requests from a known requestor. For example, an installation of an Identity Management system could serve as a provider.
- Provisioning Service Target (PST) or target represents a destination or endpoint that a provider makes available for provisioning actions.
- A Provisioning Service Object (PSO), sometimes simply called an object, represents a data entity or an information object



The SPMLv2 core operations

- a discovery operation (listTargets) on the provider
- several basic operations (add, lookup, modify, delete) that apply to objects on a target