CodeX

CodeX - eds2003

Enterprise Directory Services Conference 2003

The Swedish Universities IT-Directors Forum with Linköping University, Chalmers University of Technology and Uppsala University invites all Nordic Higher Education institutions to a two day conference aimed at sharing knowledge and experiences of existing enterprise directory services and discussing needs and solutions for future services. The conference focus group is IT-architects, IT-directors and CIOs. Focus will be on overall experience, solutions and future needs for cooperation. In order to be able to have a lively discussion we will limit the number of participants from each university if necessary.
The conference will be held January 29-30th in Uppsala.

Presentations

- LADOK
- Feide, Isabel Barosso Gomes, UiO
- Microsoft
- Integrator, Eddie Hartman, IBM
- Ldap vid LU, Johan Ekman, LU
- Directory related activities at Uppsala University, Pål Axelsson, UU
- SPOCP / PKI / BKS, Torbjörn Wiberg, UmU
- LiU - LUKAS, Mattias Carlsson, LiU
- LiU - LiUPass, Jim Nordlander, LiU
- SUN ONE, Peter Gustafsson, Inserve
- Roskilde, Mads Freek Petersen, Roskilde universitetscenter
- Chalmers
- User administration inside Finnish Higher Education Institutesresults from the KATO project, Barbro Sjöblom, Åbo akademi

CodeX

CodeX

PEPC 2003

EDS 2003

PortalConf 2002

Välkommna!

De svenska universiteten och högskolorna inbjuds att delta i ett nätverk för utbyte av erfarenheter, kod, design med mera. Det som just nu står högt i fokus är arbetet kring portaler.
Det finns en publik maillista: CodeX@listserv.liu.se.
För att komma med på denna maillista så behöver du bara anmäla dig på Ansök sidan.

Styrgruppen har en maillista: codex-styrgrp@unit.liu.se

Medlemmar i styrgruppen:

Joakim Björklund, Linköpings universitet
Magnus Lindqvist, Lunds universitet
Per Wising, Stockholms universitet
Magnus Andersson, Umeå universitet
Dario Lopez-Kästen, Chalmers

Conference list

Name	University
Magnus Andersson	Umeå universitet
Magnar Antonsen	Universitet i Tromsø
Sven Arvidson	Uppsala universitet
Pål Axelsson	Uppsala universitet
Gunnar Backteman	Stockholms universitet
Isabell Barroso Gomez	Universitetet i Oslo
Sören Berglund	Umeå universitet
Magnus Bergroth	Mälardalens högskola
Johan Bergström	Umeå universitet
Joakim Björklund	Linköpings universitet
Björn Brenander	Linköpings universitet
Tor Bu	Univ. i Bergen
Mattias Carlsson	Linköpings universitet
Tobias Ekenstam	Högskolan i Trollhättan/Uddevalla
Johan Ekman	Lunds universitet
Helge Falkenberg-Arell	Universitetet i Oslo
Joel Fredrikson	Uppsala universitet
Alex Gatica	Karolinska Institutet
Ulf Glad	Högskolan i Trollhättan/Uddevalla
Kjell Gullberg	Stockholms universitet
Peter Gustafsson	Inserve Technology AB
Roland Hedberg	Umeå universitet
Lars-Owe Ivarsson	Uppsala universitet
Bengt-Olov Jansson	Luleå tekniska universitet
Carl Jarnling	Karolinska Institutet
Åke Johansson	Uppsala universitet
Leif Johansson	Stockholms universitet
Annette Johansson	Högskolan i Trollhättan/Uddevalla
Magnus Jonsson	Umeå universitet
Leif Lagebrand	Blekinge Tekniska Högskola
Lars-Elve Larsson	Uppsala universitet
Per Lindgren	Uppsala universitet
Stig-Göran Lindqvist	Åbo Akademi
Maximiliano Lubian	Göteborgs universitet
Jens Låås	Sveriges lantbruksuniversitet
Jan-Martin Löwendahl	Chalmers tekniska högskola
Daniel Martinsson	Sveriges lantbruksuniversitet
Gerolf Nauwerck	Uppsala universitet
Anders Nilsson	Luleå tekniska universitet

Jim Nordlander	Linköpings universitet
Hans Nordlöf	Karolinska Institutet
Emil Pedersen	Uppsala universitet
Inge Persson	Sun Microsystems
Mads Freek Petersen	Roskilde universitet
Anders Qvist	Linköpings universitet
Johannes Schmidt	Sveriges lantbruksuniversitet
Börje Sennung	Chalmers tekniska högskola
Barbro Sjöblom	Åbo Akademi
Andora Sjøgren	University of Oslo
Harald Skotnes	Universitetet i Tromsø
Bo Stanley	Örebro universitet
Gunnar Ståhl	Sveriges lantbruksuniversitet
Davor Vusir	Sveriges lantbruksuniversitet
Per Wallentinson	Uppsala universitet
Paul Waserbrot	Chalmers tekniska högskola
Per Wernheim	Karolinska Institutet
Torbjörn Wiberg	Umeå universitet
James Ytterstene	Mälardalens högskola
Kent Åberg	Sun Microsystems
Kuno Öhrman	Svenska handelshögskolan

Number of attendees: 60

UNIVERSITETET
I OSLO

FEIDE

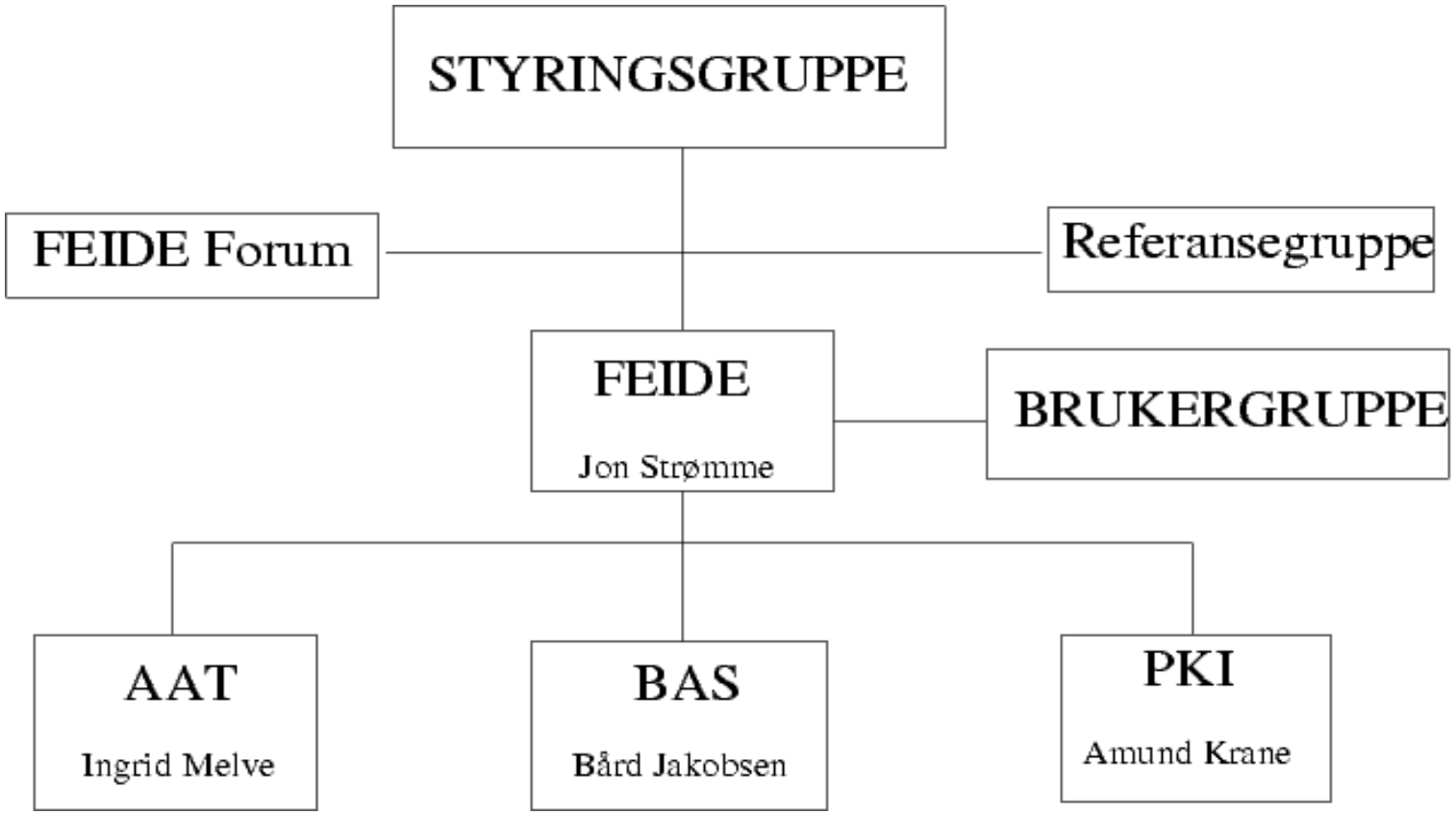
Isabel Barroso Gomez
Overingienør ved USIT

UNIVERSITETET
I OSLO

Om FEIDE-prosjektet

- UNINETT prosjekt på oppdrag av UFD. Hovedaktører USIT og UNINETT, mange deltakere
- Finansieres av UFD og UNINETT + egenfinansiering i sektoren
- Forankret i sektoren ved styringsgruppe hovedsakelig nominert av UHR
- IT-prosjekt, trenger koordinering på administrativ side

UNIVERSITETET
I OSLO



UNIVERSITETET
I OSLO

Hvorfor FEIDE

- Orden på persondata
- Samme person: student ved flere læresteder, ansatt, databruker på mange systemer, låntaker, biblioteksbruker, ...
- Riktige data om personen, fra alle systemer, med personvern
- Riktig tilgang for personen, alle steder, med samme identitet, med sikkerhet

UNIVERSITETET
I OSLO

Grunntanken i FEIDE

- En identitet, et lite antall former
- En kilde for hvert persondata. Samme registre som før, men kun ett er autoritativt for hvert data).
- Ett sett med roller og rettigheter
- En autentisering
- En autorisasjon

- + riktig samspill med resten av verden på PKI

UNIVERSITETET
I OSLO

En identitet

- Personen skal kjennes igjen overalt, en identitet på noen få standardiserte former.
- Tre klassiske former: bruker@domene, fødselsnummer, sertifikatinnehaver.

UNIVERSITETET
I OSLO

En datakilde for hvert data

- Forutsetning for riktige data: Det er bestemt hva som er autorativ kilde for hver eneste type data.
- Kodene for personopplysninger må være fra samme sett.
- Stor jobb for administrasjoner og IT-tjenester.
- FEIDEs tilbud: UREG, autentiseringstjeneste, definisjoner, samarabeid og støtte.

UNIVERSITETET
I OSLO

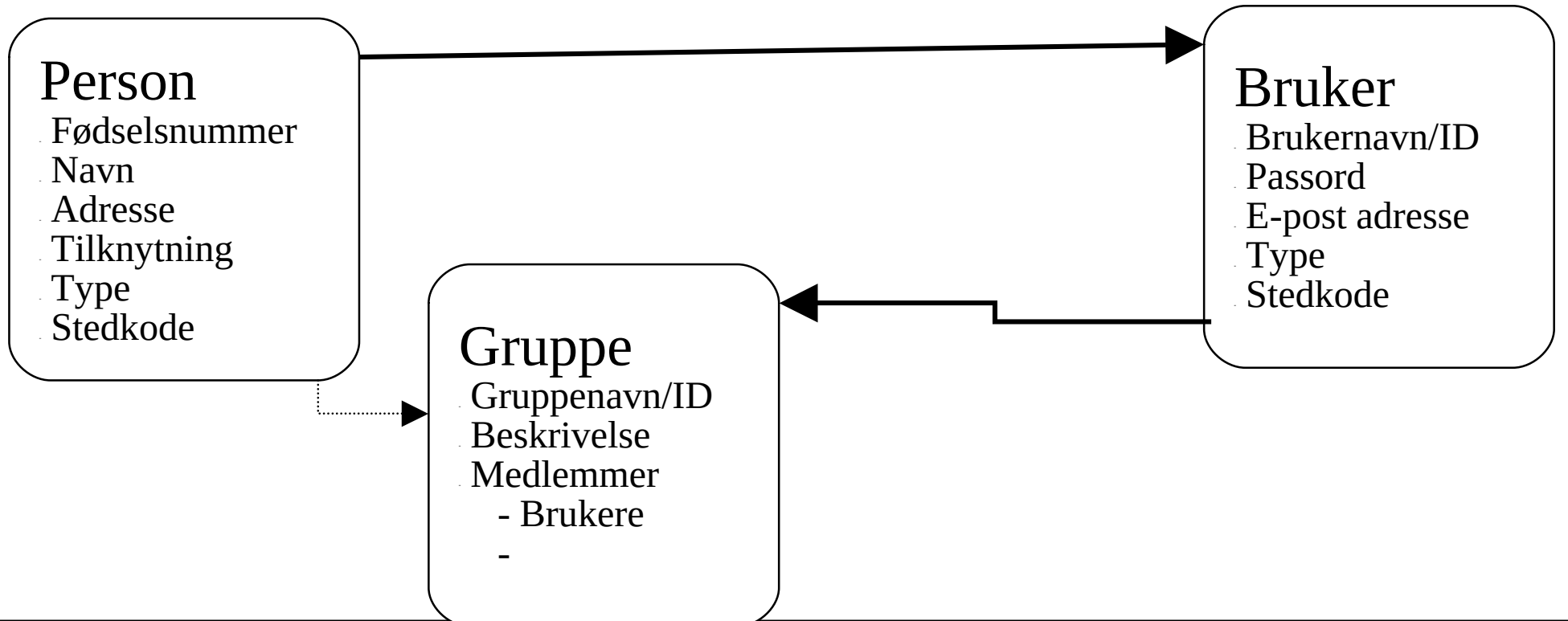
Ett sett med roller og rettigheter

- Personer har roller. Roller gir rettigheter.
- Rollene må være de samme for å fungere over flere institusjoner.
- Dette er en sentral del av kodene, den største jobben for institusjonene er definering og harmonisering av kodene.

UNIVERSITETET
I OSLO

Synsvinkel riktige data

- Personer, roller, rettigheter og attributter
- Datakvalitet og konsistens

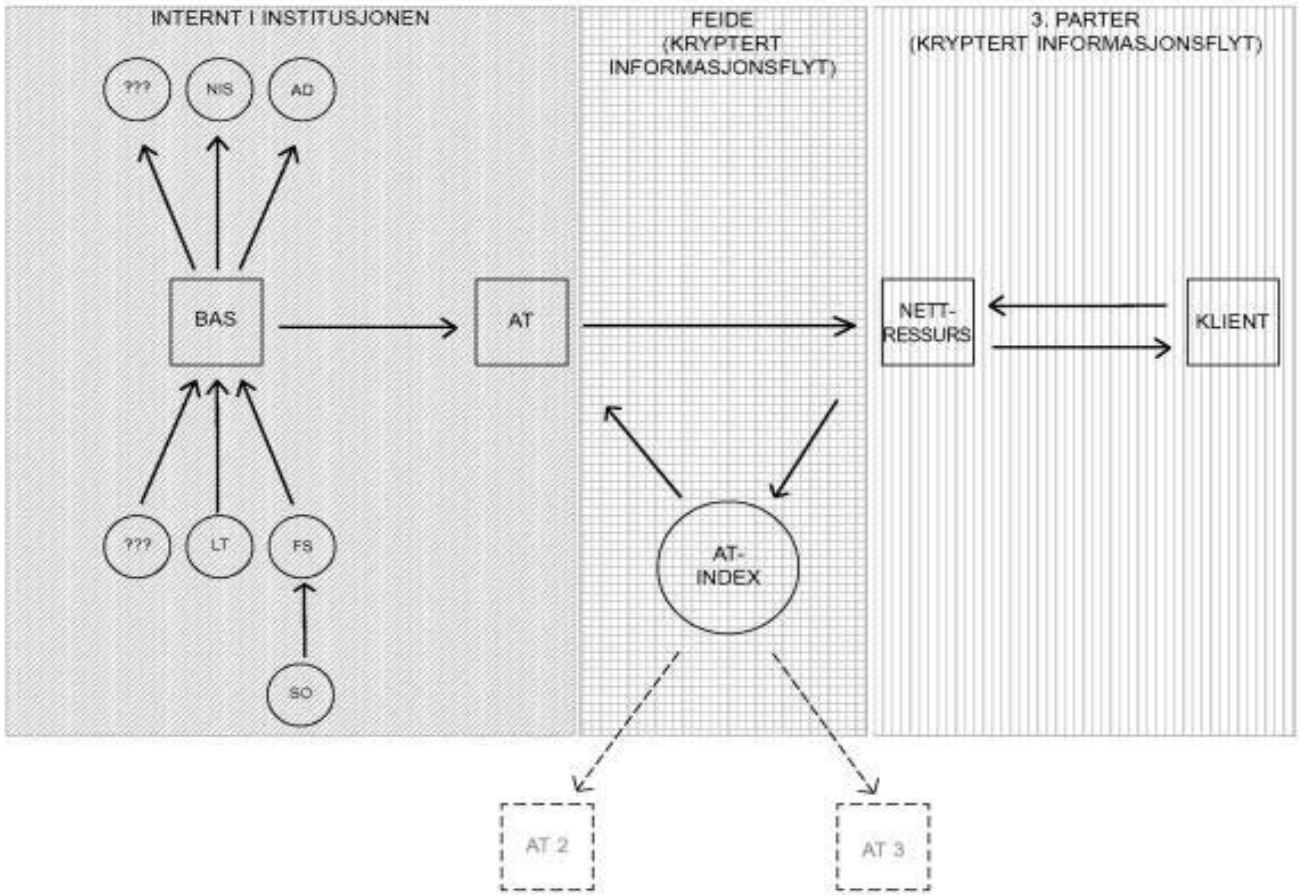


UNIVERSITETET
I OSLO

En autentisering

- Det som FEIDE koker ned til eksternt.
- Autoritative data --> AT
- Tjenester spør AT: Er akkreditivene OK? (Får ikke nye data fra AT, kun bekreftelser).
- Tjenester spør AAT: Javel, vi vet hvem han er. Får han lov?
- Personvern, informasjonslekkasje.
- En demo versjon av FEIDE AT https://stum.uninett.no/demo/

UNIVERSITETET
I OSLO



UNIVERSITETET
I OSLO

PKI

- Valgt teknikk for høy sikkerhet.
- Studentkort/ansattekort med smartkort.
- Koordinering med annen bruk av PKI.
- Griper også inn i institusjonene: kortadministrasjon.

UNIVERSITETET
I OSLO

Hva er et BAS?

- BAS er et brukeradministrativt system. ☺
- Etablering av et BAS er en administrativ prosess, utvikling og omorganisering (og en liten teknisk greie).
- Etablering: Teknologien er 20%, administrasjonen er 80%.

UNIVERSITETET
I OSLO

BAS

- Input
  - Autorative opplysninger om personer tilknyttet institusjonen
  - Krever:
    - Oppdatert studentregister
    - Oppdatert personalsystem
- Output
  - Autentiserings info
    - I dag: Brukernavn (ID) + passord
    - I morgen: ?
      - Synkroniserte grupper på bakgrunn av informasjon fra personalsystemet/studentregisteret

UNIVERSITETET
I OSLO

Hva gjør UREG2000

- Administrerer brukere
  - bygger, endrer og sletter brukere
  - autentisering
    - OS – Unix, NT, W2K, osv
    - Systemer – Lotus Notes, ClassFronter, Oracle, osv
  - rettigheter
  - printerkvoter
- Administrerer grupper
- Skal administrere meta-info om maskiner

UNIVERSITETET
I OSLO

Hvorfor Cerebrum?

- Et nytt BAS omprogramert og i FEIDE tanke
- Ønsker synkronisert brukernavn + passord.
- Ønsker ETT sted å oppdatere informasjon om brukere, uansett hvilket system/OS brukeren bruker.
- Ønsker ETT kontaktpunkt for eksterne som ønsker autentiserings- og autorisasjons-informasjon.
- Ønsker ETT utgangspunkt for levering av informasjon til nye tjenester (Katalog, LMS, osv.)

UNIVERSITETET
I OSLO

Autorative datakilder

- Identifiser institusjonenes administrative systemer
- Avgjør hvilke systemer som er autorative
- Avdekk eierskap til disse datakildene
- Det må defineres hvilke datakilder som skal være autorative når en ”bruker” er representert med ulike roller i de ulike datakildene.
- Og hvilke datakilder skal være autorativt i forhold til de ulike attributtene rundt en persons opplysninger

UNIVERSITETET
I OSLO

Datakilder

- Er det fysisk mulig å hente ut data fra de ulike kildene, men en innhentingsfrekvens som gjør at man til hver tid har de nyeste dataene i BAS-et
- Har man tillatelse til å innehente data(personopplysningsloven)

UNIVERSITETET
I OSLO

Eierskap av data

- Eierne av dataene er de som vedlikeholder data og er systemeiere
- Må ha klare regler for ”utlevering” av data.
- Dataeiers rutiner for innlegging av data

UNIVERSITETET
I OSLO

Datakvalitet

- Har dataeier og BAS-eier samme forståelse av hva som er verdien av de ulike attributtene
- Avdekking av spesialtilfeller som institusjonen har laget egne rutiner for
- Eierne av dataene skal til enhver tid sørge for at de data som hentes ut fra deres systemer er korrekte
- Når det blir oppdaget ”feil” i dataene skal det meldes til data eiere, ikke BAS-eier

UNIVERSITETET
I OSLO

Endringshåndtering

- Det må defineres klare rutiner for hvordan man skal melde feil i opplysninger
- Det er viktig at man skiller mellom hva som er systemendringer og hva som er personopplysningsendringer
- Man må ta hensyn til individets behov for endring kontra institusjonen behov for endring av personopplysninger
- Personopplysningsendringer skal til systemeier
- Systemfeil skal meldes til BAS-eiere

UNIVERSITETET
I OSLO

Prosjektorganisering

- Hvem bør være med i denne prosessen?
- Hvilke aktører må man forholde seg til og hvilke ”roller” bør dekkes av prosjektdeltagerne?
- Systemeiere må avdekkes og ta en aktiv rolle i innføringen
- Systemeiere er ofte den administrative og økonomisk enhetene på institusjonen
- Personer fra disse gruppene bør være med i prosjektet for å opplyses i forhold til hva de må gi for å få ”rene” data igjen

UNIVERSITETET
I OSLO

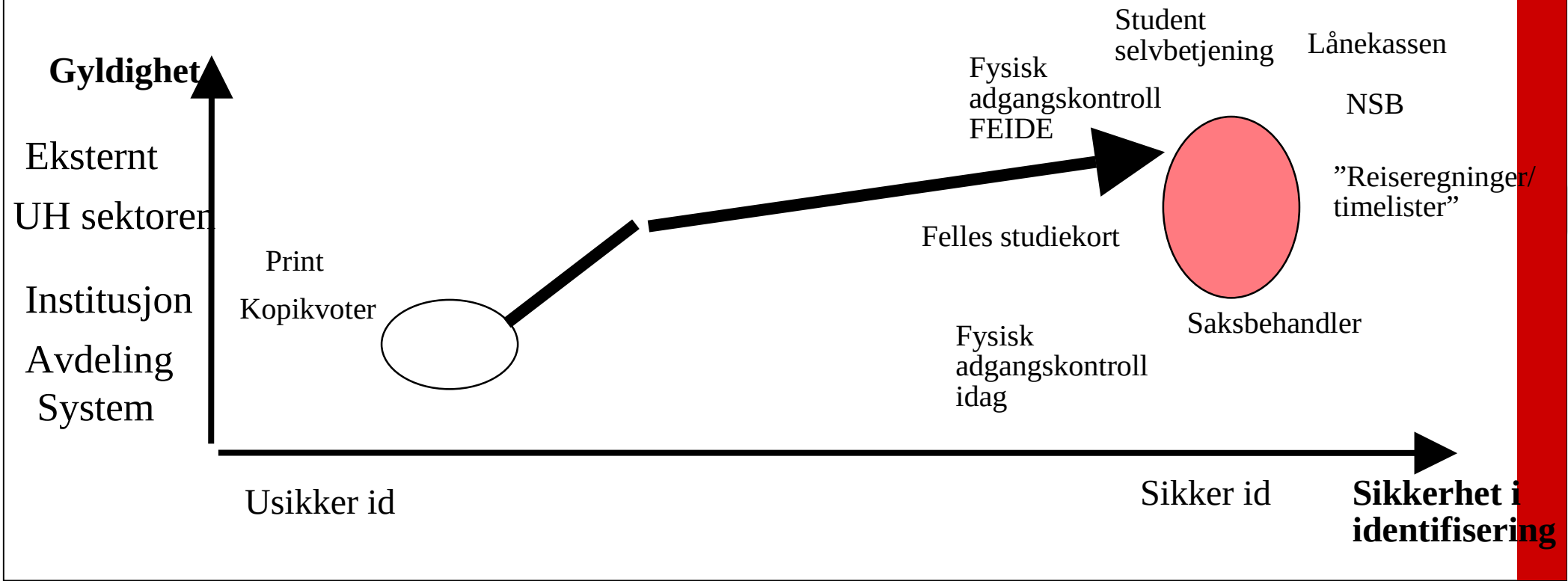
Veien videre

- Det vil komme en mer detaljert veiviser på hvordan man kan etablere et BAS
- Samt råd og tips på hvordan vi ser at man kan dra nytte av andres erfaringer

UNIVERSITETET
I OSLO

Synsvinkel sikker ID

- Fra usikker identifikasjon, per system
- Til sikker identifikasjon for hele sektoren



UNIVERSITETET
I OSLO

Hvor står FEIDE nå?

- Arkitektur etablert
- Standarder og standard programvare ferdigstilles
- Samarbeid i sektoren etablert
- Piloter definert og igangsatt
- Største utfordringer: avtaler, organisasjon og framdrift hos institusjonene.

UNIVERSITETET
I OSLO

PDA(Personlige Deltagere i avspark)

- Det er plukket ut i 5 ulike institusjoner som USIT og Uninett skal håndholde gjennom deres deltagere i FEIDE
- Vi skal holde kurs på Uninett konferanse til sommeren slik at institusjoner som vil bli FEIDE deltagere kan få full opplæring i BAS,AAT og FEIDE tankegangen.

UNIVERSITETET
I OSLO

Fiberskoler

- Første eksterne utrulling av BAS/Cerebrum blir gjennomført på Grunn og videregående skoler
- FEIDE tankegangen vil følges
- Drøm: Samme ID fra Barneskole til Universitet/Høyskole

UNIVERSITETET
I OSLO

Helt til slutt

- Se www.feide.no

Enterprise Directory Services Conference 2003

IBM Directory Integrator

IBM

DoD : Directory of Directory Higher Education – www.internet2.edu/dodhe

- A project of **MACE** (Middleware Architecture Committee for Education)
- Investigating technology to support inter-institutional directory searching.
- An application of:
  - the eduPerson object class
  - **LDAP Recipe** and the **good practices** presented by the Internet2 Middleware Initiative, MACE, and the MACE-Dir, MACE-Shibboleth and HEPKI working groups.

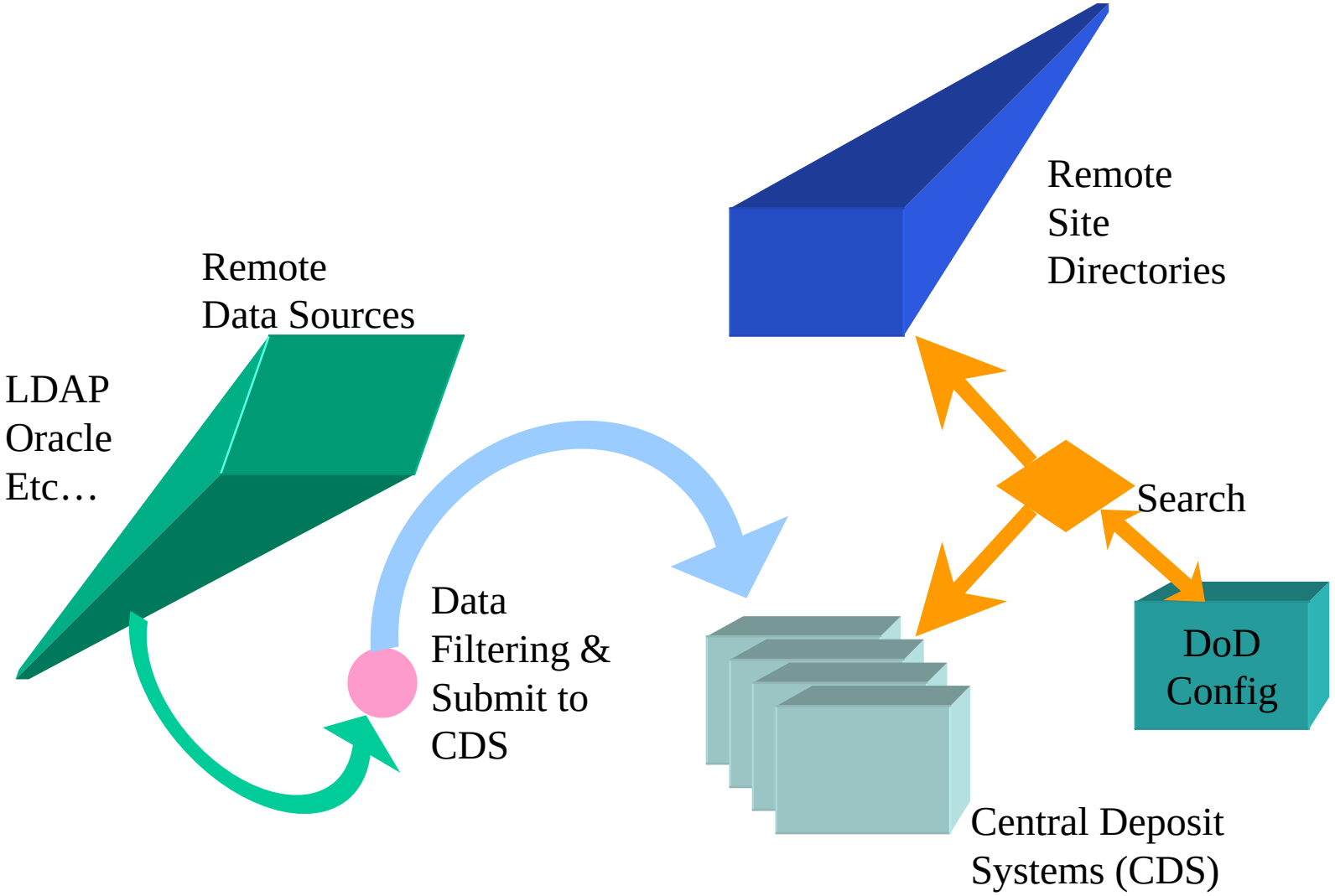
IBM

DoD : Goals

- Investigate and develop a service for directory searching ("Web of People“)
- Employ the use of other MACE-related activities and infrastructures:
  - the Internet2 Middleware Initiative and its work regarding the understanding and definitions of institutional identifiers, authentication schemes and registries of people data.
  - the eduPerson object class definition
  - the LDAP Recipe (more uniform configuration and deployment of institutional directory services)
- Allow each institution complete control over the data and the privacy issues
- Document thoughts, perspectives, progress, methodologies, failures, discoveries...
- Work with appropriate international experts to foster interoperability and alignment with other similar initiatives.
- (Hopefully) make such a service real for the worldwide academic community and possibly influence how such services would be deployed in the commercial sector.

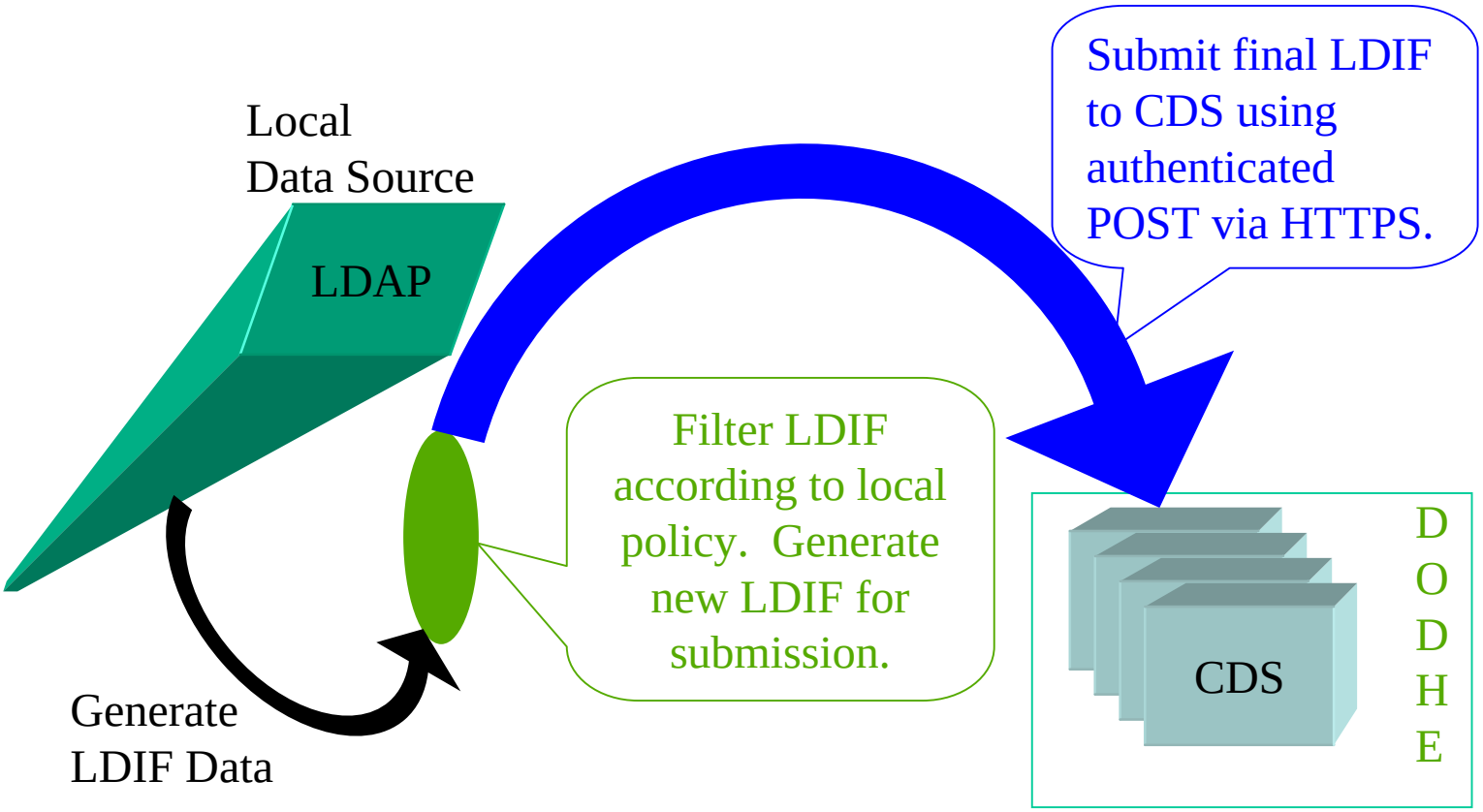
IBM

DoD : Inputs



IBM

DoD : Inputs (Local View)



IBM

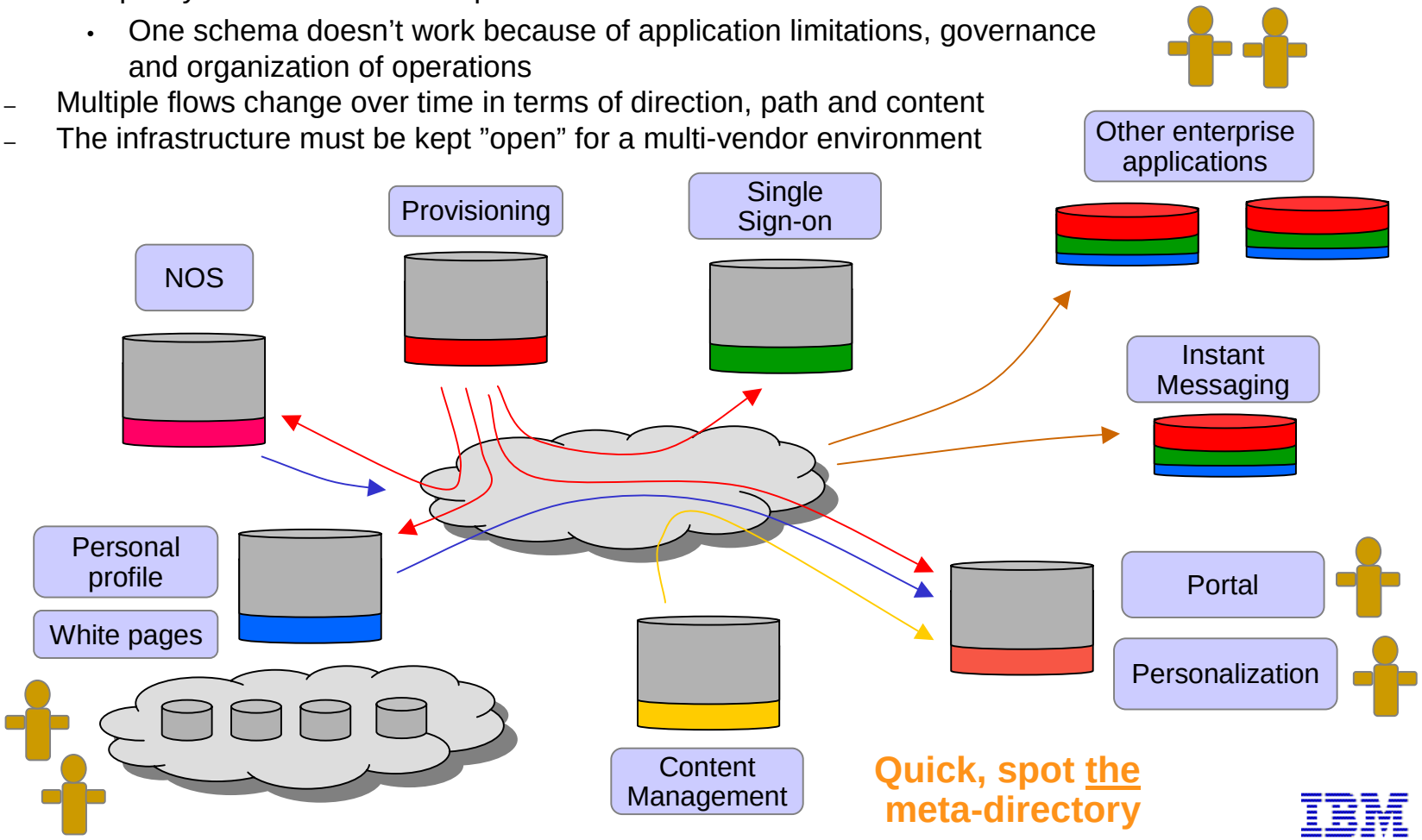
DoD : Inputs – *Why this way?*

- Standardized input is LDIF
  - Could be XML but few products generate XML now (01/2001)
- Could use Directory Integrator as filter and submission mechanism
- Site always submits full dataset. No worry of reconciling. Easier site participation in the DoDHE service.
- CDS handles reconciliation and controls data processing. Can provide feedback.

IBM

Identity data in the enterprise

- Applications conform to standards, but they mostly perform their best with repository control. Not least because of governance and security issues
  - ”Personalization will not happen in SSO directory”
- Multiple systems consume and publish attributes
  - One schema doesn’t work because of application limitations, governance and organization of operations
- Multiple flows change over time in terms of direction, path and content
- The infrastructure must be kept ”open” for a multi-vendor environment



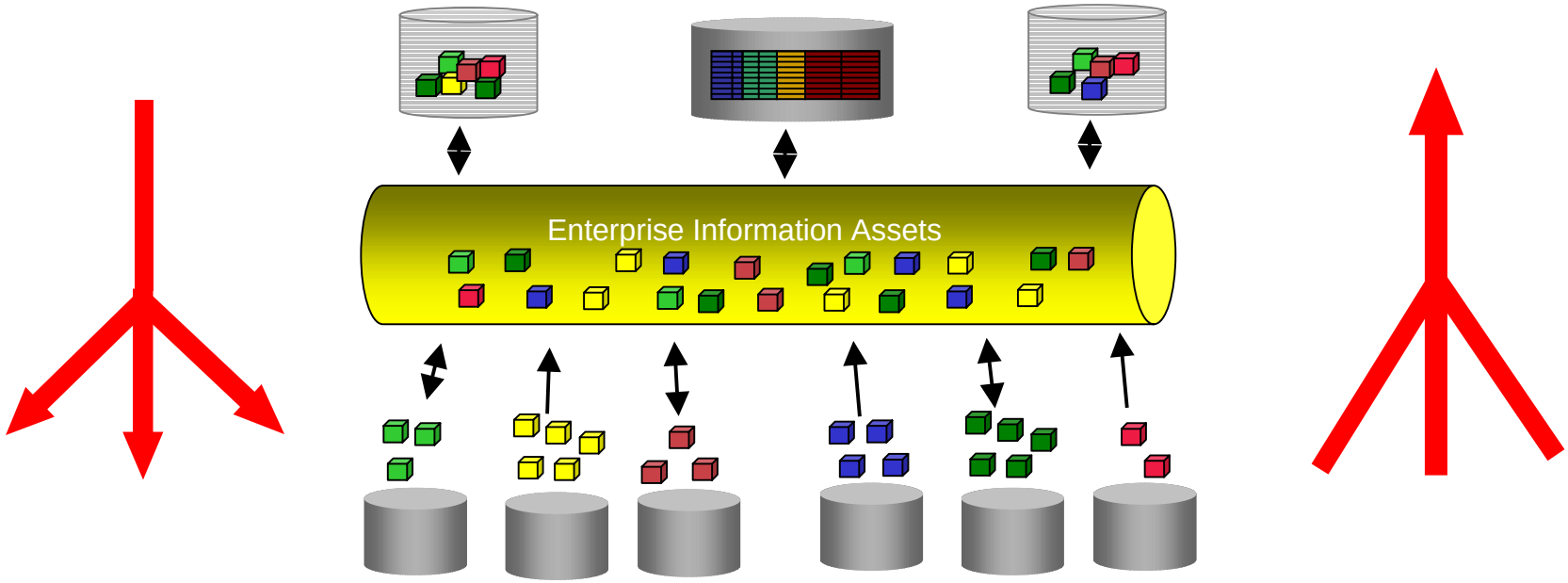
Introducing “PeopleSpace”

- The sum of people-centric information in the enterprise
- Permeates the enterprise and spans systems, geography, lines of business, processes and governance
- The pieces are owned by everybody, but the system by nobody
  - Not managed by one, but by many (all) systems: Human Resources, ERP, email, groupware, security, self-service, workflow and others
- Characteristics
  - Unique in the enterprise, like no other data structure
  - Data structure is object centric, not “set” or relations
  - Distributed, duplicated, replicated
  - Heterogeneous and distributed, across platforms, systems, protocols
- Distributed, duplicated, replicated – and will stay that way.
  - We are not well equipped to deal with it.

IBM

Information flows in the enterprise

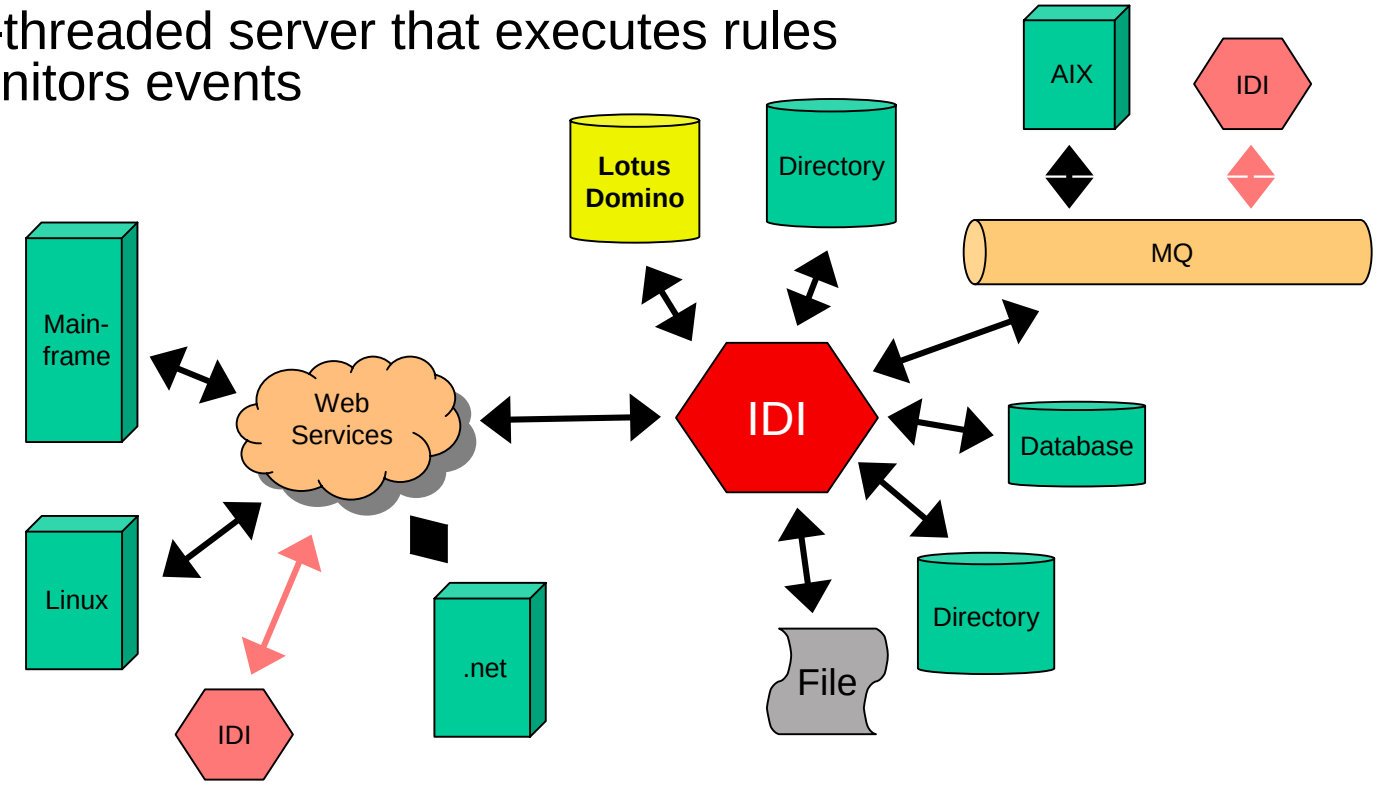
- Centralization vs distribution
  - Today's enterprise requires leveraging diversity and striking a balance between data centralization and provisioning.
  - Some data needs to be centralized to provide scalable services, while other must be duplicated in a controlled manner to support the deliver the same.
- Both approaches require middleware
  - that can move, transform and coordinate data, and data relationships across a multitude of systems, api's, protocols and formats.



IBM

IBM Directory Integrator

- A real-time, event driven, general-purpose, data integration environment consisting of:
- A rapid development GUI for building and maintaining transformation and synchronization rules
- A multi-threaded server that executes rules and monitors events



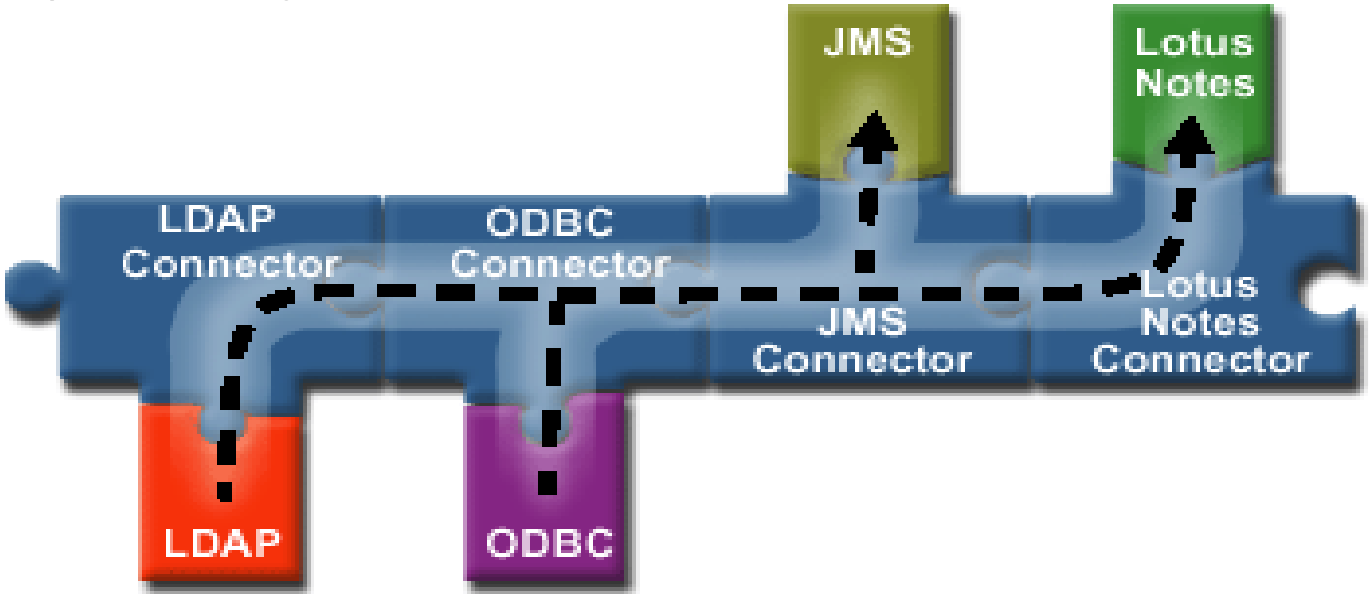
Design goals

- "No" assumptions
  - About platforms, protocols, schema, source and target systems
- Enable any-to-any, many-to-many integration
  - Manage multi-source, multi-location, multi-direction and multi-master information structures
- Provide default behavior and fundamental primitives
  - Designed for exceptions, hooks, triggers and extensions
- Striking a balance between solution and framework
  - Flexibility often loses out to out-of-the-box functionality. Cards leave the table for every assumption that is made
  - Products tend to gravitate towards solutions rather than frameworks. There is always one more feature to make it a little more simple to use. Thereby making it a little less flexible
- Leverage key technologies
  - Java, IP/XML/SOAP/XSL, Message Queuing, Web Services
- System and repository abstraction layer
  - Canonical power
  - Unique methodology and deeply configurable data workflow
- General purpose transformation engine
  - "Open" abstraction layer to external environment through parsers, connectors and event handlers, including XSL transformation
  - External logic, programs and SDKs can be brought to bear on the information flow

IBM

Connecting data in systems

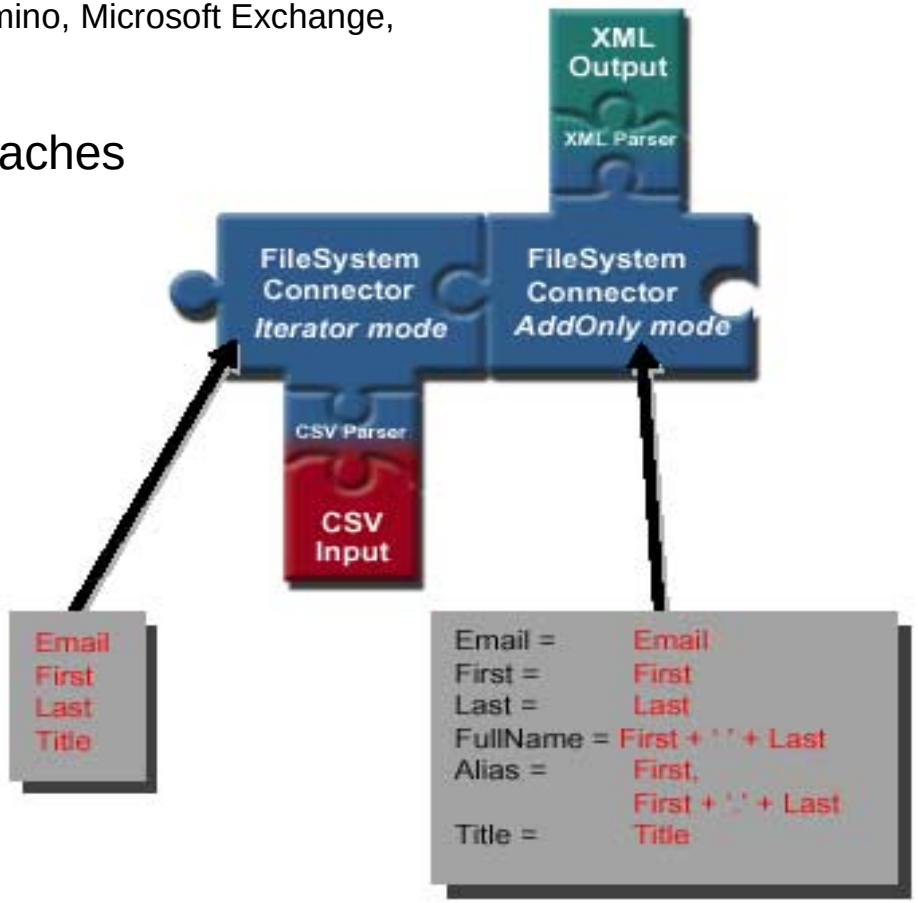
- Moves, copies and transforms data between systems
  - Unique AssemblyLine methodology provides unparalleled speed of deployment, development and maintenance
  - Maps between schemas and attributes of the connected systems
  - The combined attribute flow and transformation rules create output for the target systems
  - Supports JavaScript and VBScript as scripting languages for business logic and exception handling



IBM

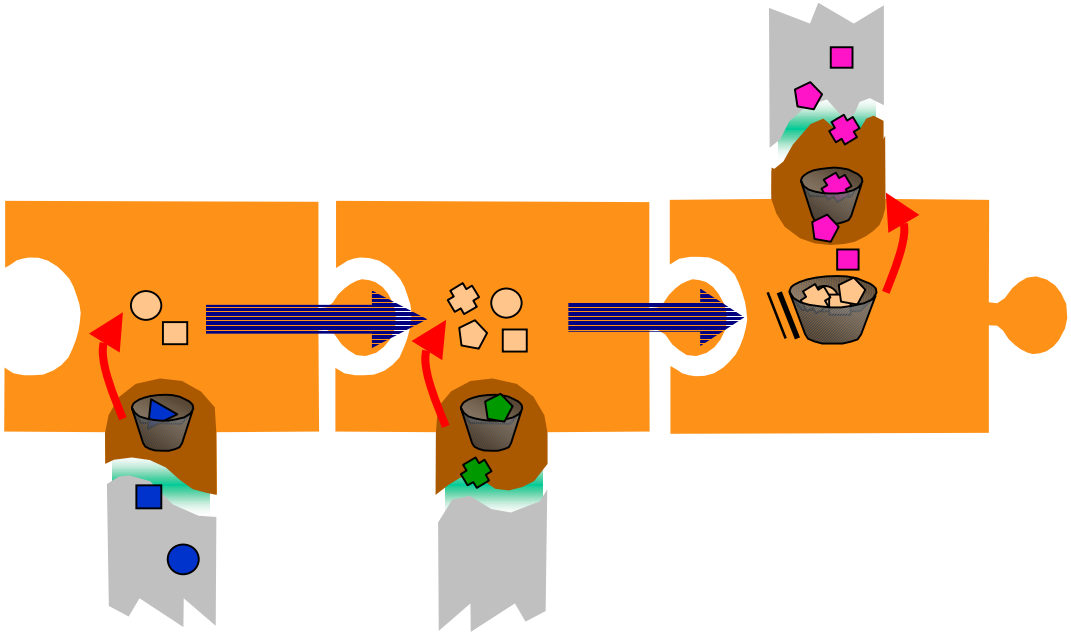
Uniquely suited for identity infrastructure

- Specific integration with systems that contain authoritative identity data
  - Microsoft Active Directory, Lotus Domino, Microsoft Exchange, detailed LDAP support, LDIF, DSML
  - As well as other enterprise systems
- Supports vendor specific approaches
  - Directory event notification
  - Changelog access



AssemblyLine concept

- Unique methodology to describe relationship of external systems and automate higher level flow of information
  - Connectors are strung together
  - Data is piped to canonical Java format
  - Business logic is added in this framework and connectors (next)



Another presentation provides an in-depth walk-through of this concept (Edward Hartman)

IBM

Connector *mode* and *hooks* concept

- Connector *modes* drive behavior and workflow
  - The modes describes the type of operation that the connector will perform on the target system
    - Iterator, lookup, update, delete, addonly, call/reply
  - Dramatically reduces development, testing, maintenance and future iterations
  - Mode behavior is easily overridden
- *Hooks* are specific to mode
  - Logical insertion points in the workflow of the connector that allows insertion of specific business logic
  - Examples: “after delete”, “on failure”, “before lookup”
  - Usage example: In update mode, a specific hook would fire if a record in the target system does not already exist and will be added – vs. modified/updated if it did exist

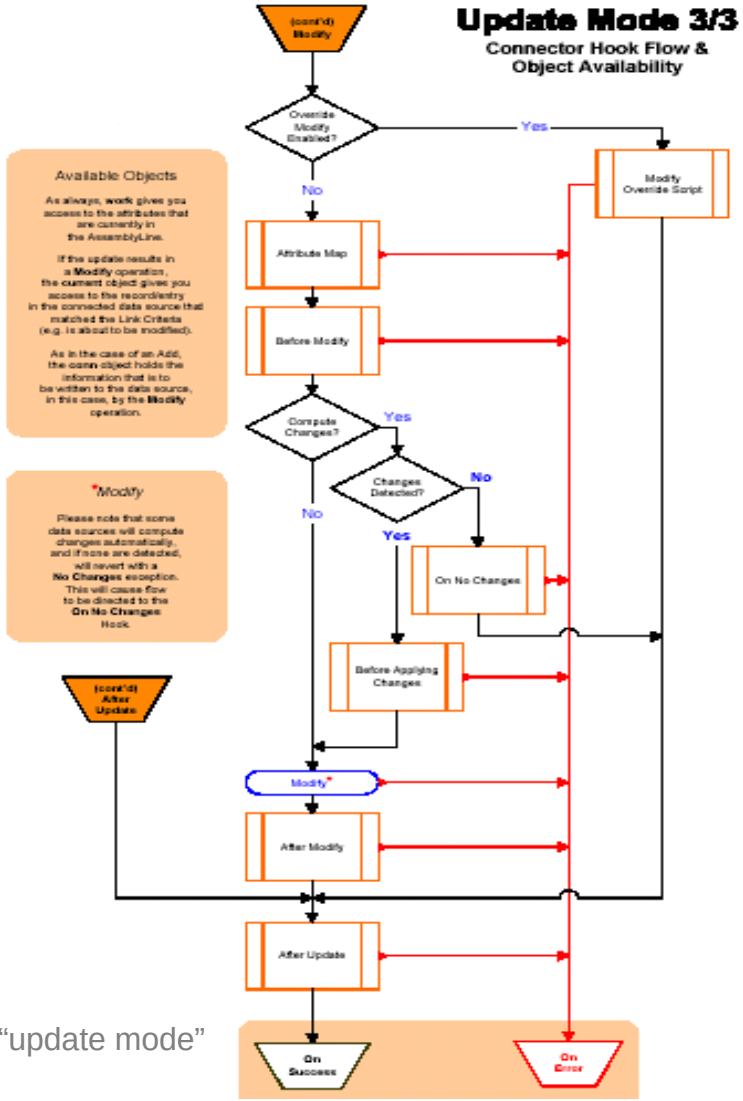
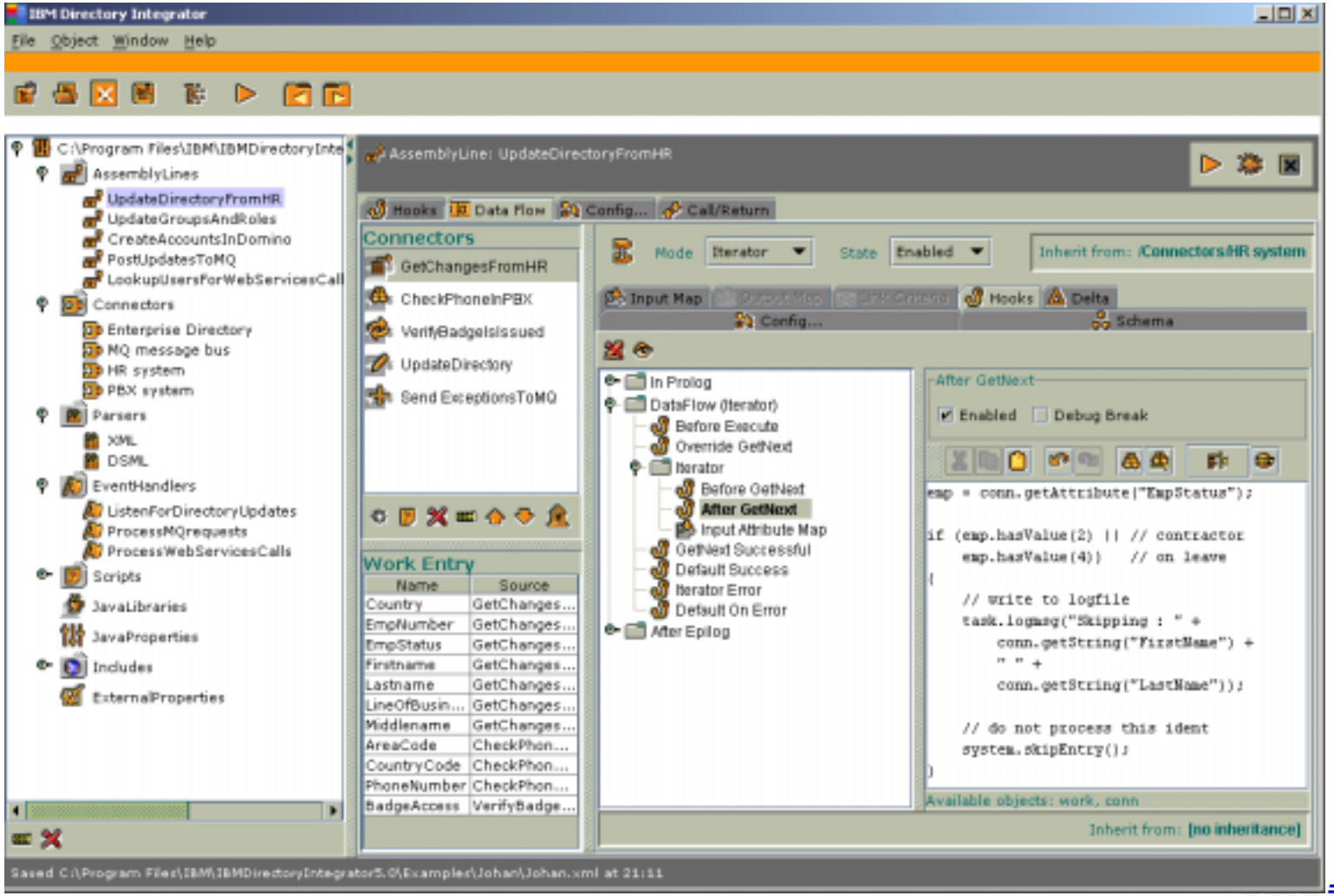


Illustration of third part of “update mode”

Graphical Development Environment

- Enables rapid development, testing and maintenance cycles



Highlights

- Event driven, general-purpose, data integration environment
  - Rapid development GUI for the transformation and synchronization rules
  - Multi-threaded execution server for rules and event-handling
- Not dependent on a repository or centralized data model
- Connects to a large number of protocols, API's and formats
  - Build information objects based on related attributes in any number of source systems and delivers them to target systems
  - JDBC, ODBC, LDAP, HTTP, IP, FTP, JMS, Web Services, Files, XML, XSLT, Lotus Domino, Microsoft Exchange/Directory, NT4, email, command-line
- Particularly suited for integrating identity data across the enterprise
- Highly extensible with JavaScript, VBScript and compiled Java
  - Configurable: attribute mapping, business logic
  - Exception handling:
  - Connectors: Adapt and override existing connectors for new systems
- "AssemblyLine" and "connector mode" workflow methodology
  - Automates the flow of attributes across systems, leaving the developer to map schemas, modify attributes and deal with exceptions
- JMX management framework exposes engine in real-time
- Publish and consume Web Services
- Multiple IDI servers use connectors to communicate to transform data across the enterprise
- Most Java platforms supported
  - Windows, Linux (Intel, z-series), AIX, Solaris

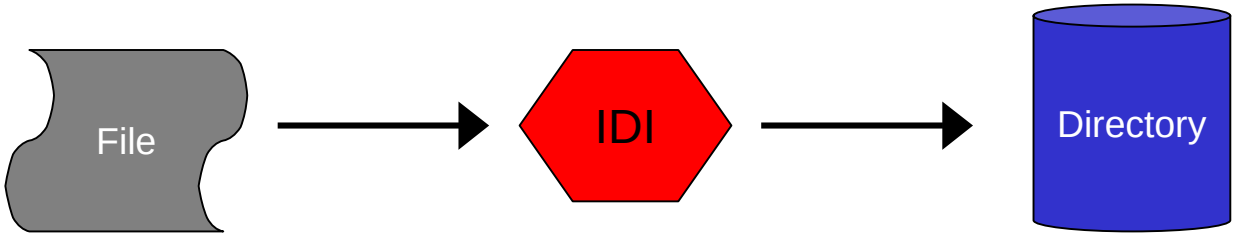
IBM

Scenarios – architecture

- IBM Directory Integrator presented through a number of ”use cases” that illustrate technical capabilities and some of the solutions that can be architected. While not a comprehensive list, it provides the creative mind with some mental structures for further development.

”One to one”

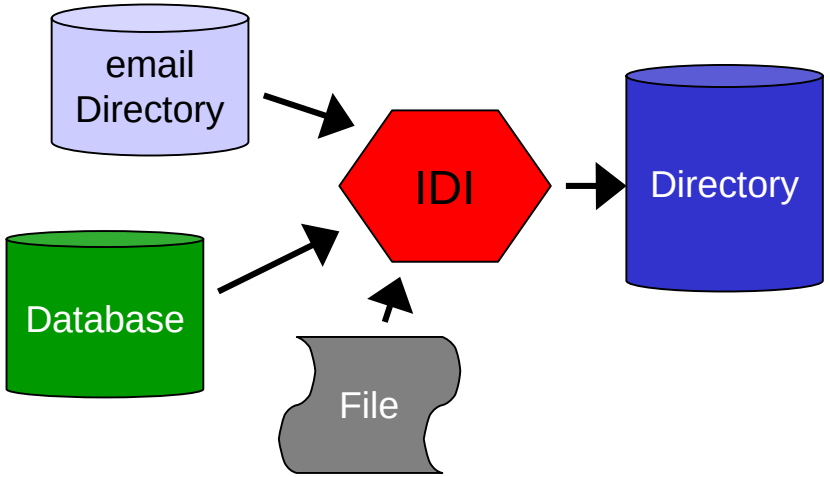
- Scenario
  - Data exists in a file that needs to be synchronized, transformed and maintained in a directory. This file could be updated regularly by a HR application or other enterprise system. A wide range of formats can be accomodated.
- Use
  - Separate connections are established to the file and the directory. After discovering the attributes in the file, IDI allows mapping to attributes in the directory as well as applying transformation rules to modify the content of the incoming data.
- Options
  - The file can be read at regular intervals, or read whenever IDI discovers that it’s available. The outside application may also trigger IDI to read the file at its own leasure.



IBM

”Many to one”

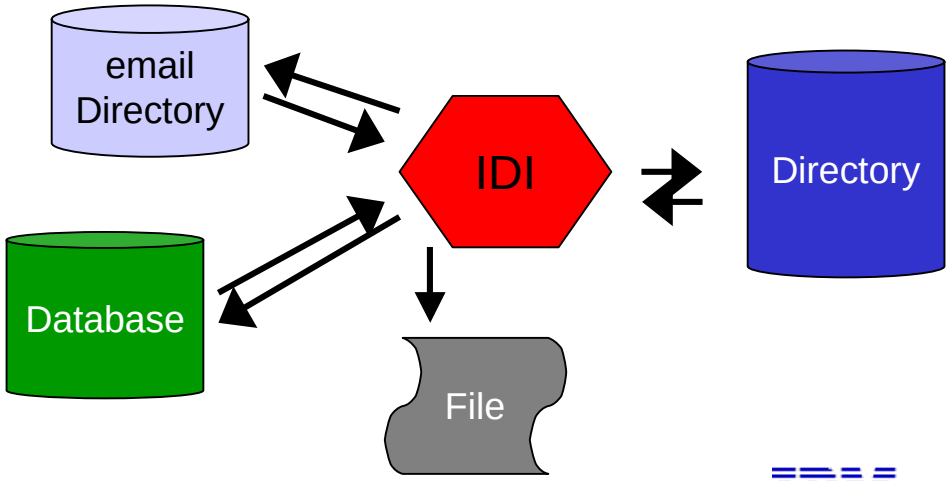
- Scenario
  - Data exists in one or multiple related systems that needs to be synchronized, transformed and maintained in a directory. The information needs to be ”joined” together before it’s updated in the directory.
- Use
  - Connections are established to each external system. Schemas in databases are automatically detected. Rules may be created that describes how attributes from one source is used with attributes from the other systems to create the desired results. Data from one system may be used to lookup information in another.
- Options
  - IDI can detect changes in real-time within certain directories, allowing immediate update of other connected systems. Connections may be configured to lookup only data that has been modified within a certain timeframe, or data sets that conform to a specific search criteria.



IBM

”One to many”

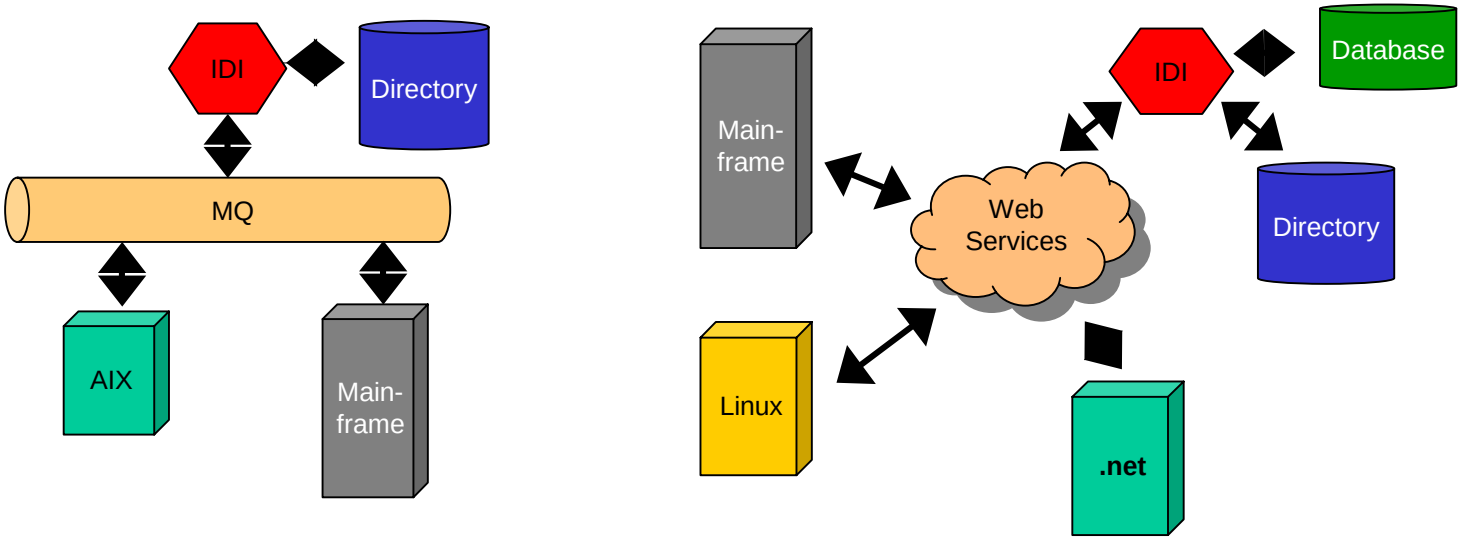
- Scenario
  - The data in the directory is a result of authoratitive data from multiple connected systems as well as possible modifications done by applications connected to the directory. The connected systems could have great interest in this data, especially when IDI ensures that they always operate on the correct information by updating them whenever the authoratitive data changes.
- Use
  - The previous examples have illustrated how IDI reads from external systems. However, IDI can perform exactly the same write/update/delete/create modifications on all connected systems as it does for directories. The rules are simply turned the other way. Now IDI will drive changes into the connected systems. Now all systems can share the common authoratitive data set.
- Options
  - This work can be performed as part of the previous examples, or as separate rules that performed on another schedule or based on other business events.



IBM

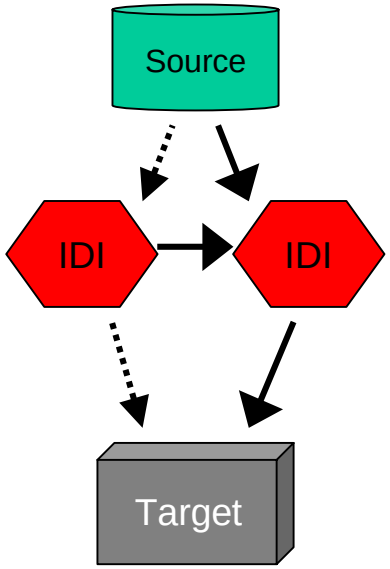
”Other data resources”

- Scenario
  - There are many reasons why data flows through channels such as message queuing, HTTP, email, FTP and Web Services. Data might need to pass through firewalls that block protocols like LDAP and database access. Security, high-availability, transactions control and desire for asynchronous or synchronous data transfer are other reasons.
- Use
  - It’s importants to understand that IDI can both send and receive with these mechanisms. This creates a wide scope of solution opportunities, too wide to describe in simple use cases. Some examples are illustrated below.

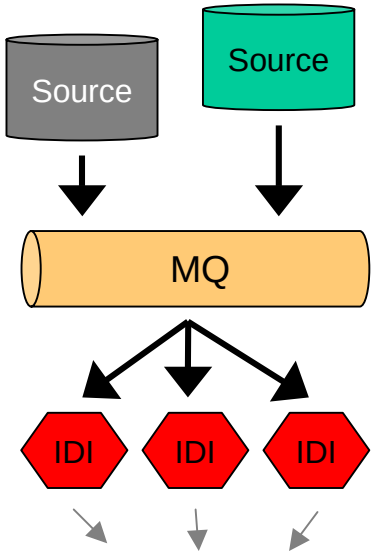


”Distributed”

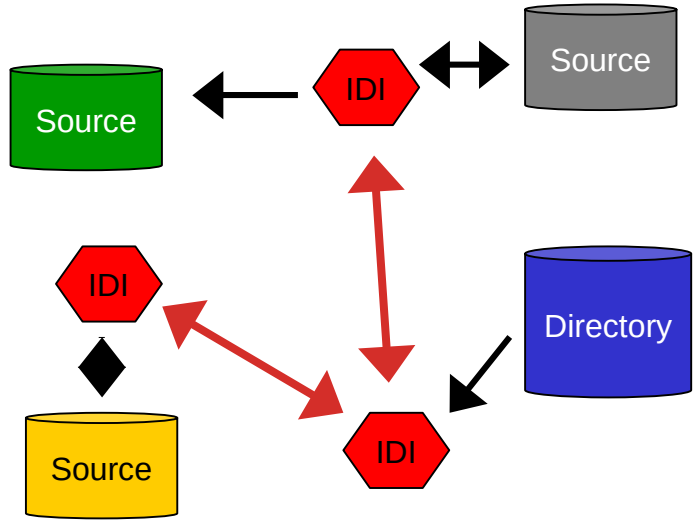
- Scenario
  - In a distributed architecture, a single point of integration is often undesireable, for reasons such as distance, financial, security, availability or governance
- Use
  - All of the mechanisms described previously, such as IP, HTTP, Web Services, email, MQ and others can be used to communicate between instances of IDI. Below, the colored arrows indicate use of such communications mechanisms



One IDI pings the other and takes over if it fails to respond



The input stream is too fast compared to the business rules that IDI has to execute. Multiple IDIs can operate of a queue

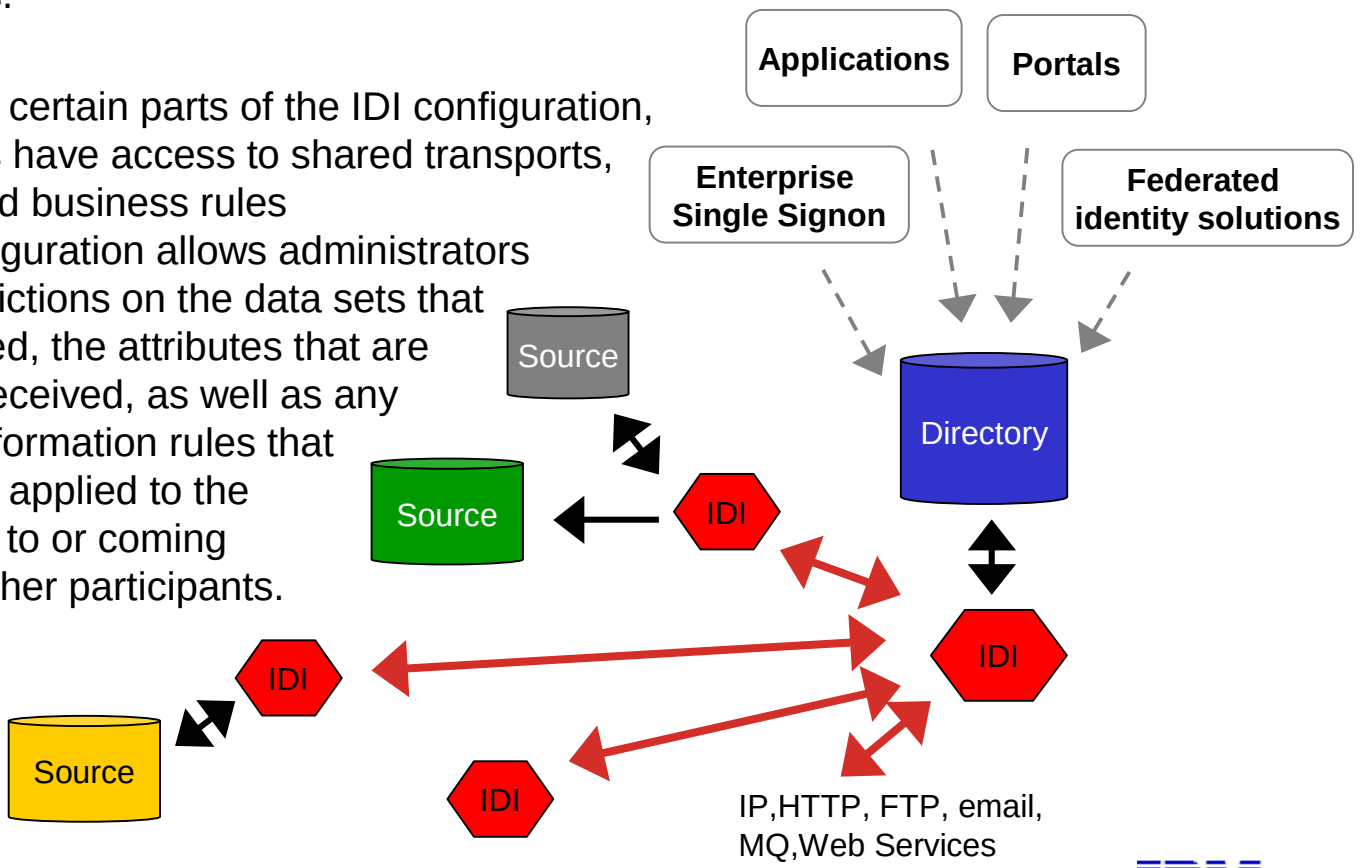


This two-way architectures propagates updates in the directory to the rest of the enterprise and consolidates local modifications back to the central directory

IBM

”Federated”

- Scenario
  - While similar to the distributed scenario, federated implies that control and management is not entirely centralized. This could be business units or entities that cooperate, but want to retain local control over how and what information is shared with others.
- Use
  - By sharing certain parts of the IDI configuration, IDI servers have access to shared transports, formats and business rules
  - Local configuration allows administrators to set restrictions on the data sets that are exposed, the attributes that are sent and received, as well as any local transformation rules that need to be applied to the data going to or coming from the other participants.



IBM

IBM Directory Integrator Connectors (I)

LDAP, changelog connectors, event notification	LDAP and X.500 directories: IBM Directory Server, iPlanet/Sun ONE, Microsoft Active Directory and Exchange 5.5/2000, Novell NDS/edirectory, many other
JDBC	JDBC/ODBC databases: DB2, Oracle, SQL-Server, Informix, MySQL
NT4	NT4 Domain SAM registries
Domino, Domino Users	Domino databases and Domino User Directory
FileSystem	Any file on a mounted file system (uses parsers)
HTTP Server & Client	Web servers, other HTTP applications
JMS	Message queuing systems, e.g. IBM MQ Series
Web Services	Both publish and consume Web Services

IBM

IBM Directory Integrator Connectors (II)

Script	Custom-written with JavaScript or VBScript
Email	POP, IMAP message stores (also can send SMTP email with system method)
SNMP	Accesses external SNMP resources
FTPClient	Retrieves and parses remote files
MemoryStream	Temporary storage buffer
Command Line	Executes shell programs and parses results
BTree DB	Fast local object storage; also used by delta mechanism

IBM

Unique capabilities & value proposition

Capabilities	Business Benefit
Not dependent on a centralized repository	Mitigate risk by deploying on existing infrastructure and not introducing any vendor repository dependence. Enables distributed deployment because coordinating logic does not have to be centralized
Integration broker architecture	Well suited to support the myriad of technical requirements in business infrastructures. Event-based architecture provides near real-time integration. Robustness and resilience
Independent of directory vendor	Both technical and business independence mitigates risk of the implementation lifespan. Customers retains power to choose best-of-breed solutions
Extreme flexibility	Addresses the specific needs of customer. The design allows for arbitrary deep level of tailoring through scripting, Java extensions or integration with other tools and systems
Java platform	Risk mitigation by platform independence. Can start with Windows2000 and continue through massive Sun server deployment
Scope of usability	Highly suitable for a number of integration needs with the enterprise even though tailored for Directory specific integration in regards to attribute multi-master&direction and source mapping

IBM

Rapid Integration Development

Two parts: IDE & run-time server

- **IDE** for creating, testing & enhancing solutions
- **Server** for deploying
- Both written in **Java** (platform agnostic)
- Both work with the **same configuration** data

IBM

Modular architecture:

- **Framework** - handles housekeeping:
  - log files, error handling, startup parameters…
- **Components** - each abstracts away a particular:
  - transport
  - API
  - protocol
  - data/file format

IBM

Methodology

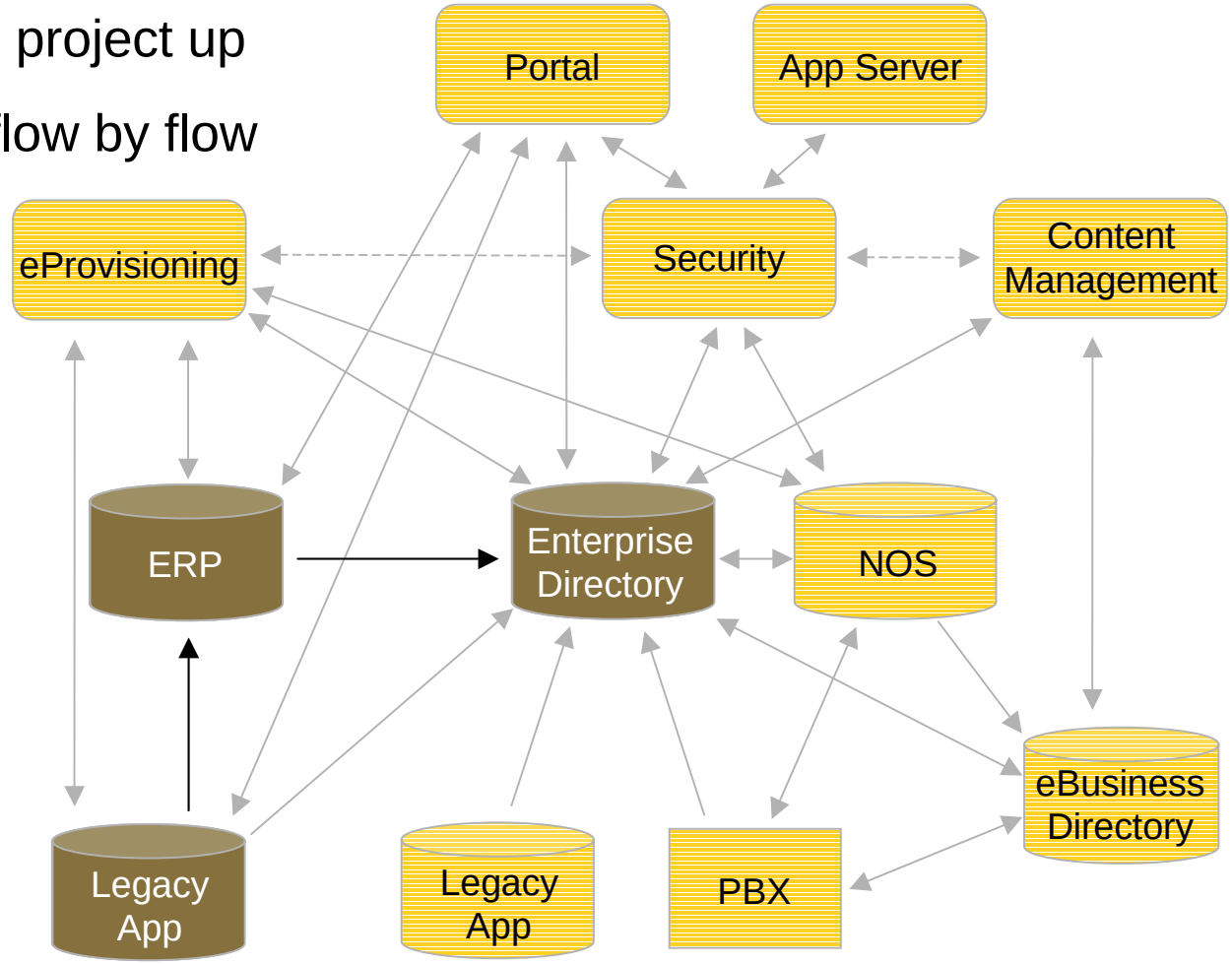
Simplify and Solve

Breaking down the problem set into its constituent parts (the data flows) and then building and deploying these rapidly

IBM

Rapid Integration Development

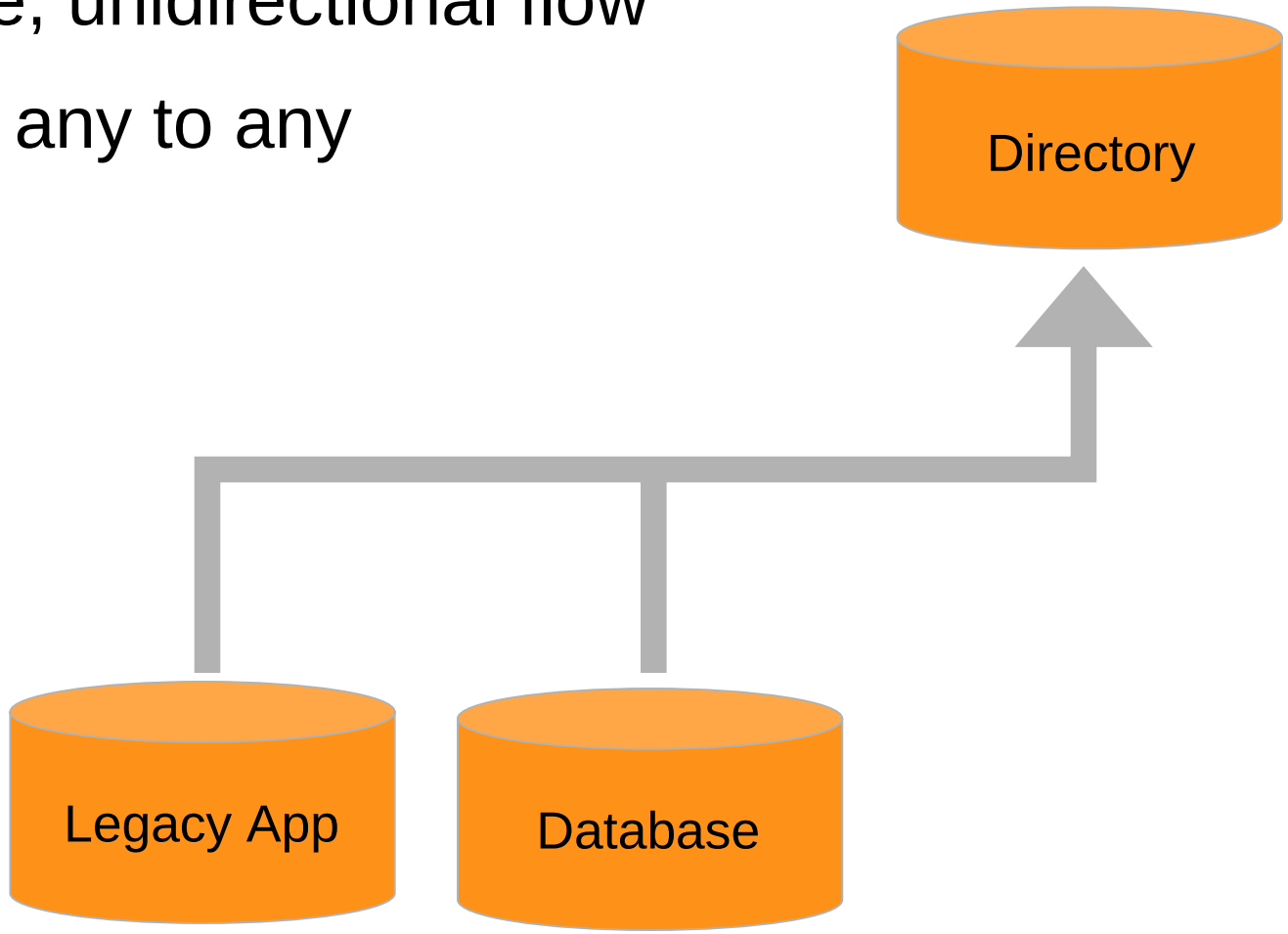
- Divide project up
- Build flow by flow



IBM

Isolate a single flow

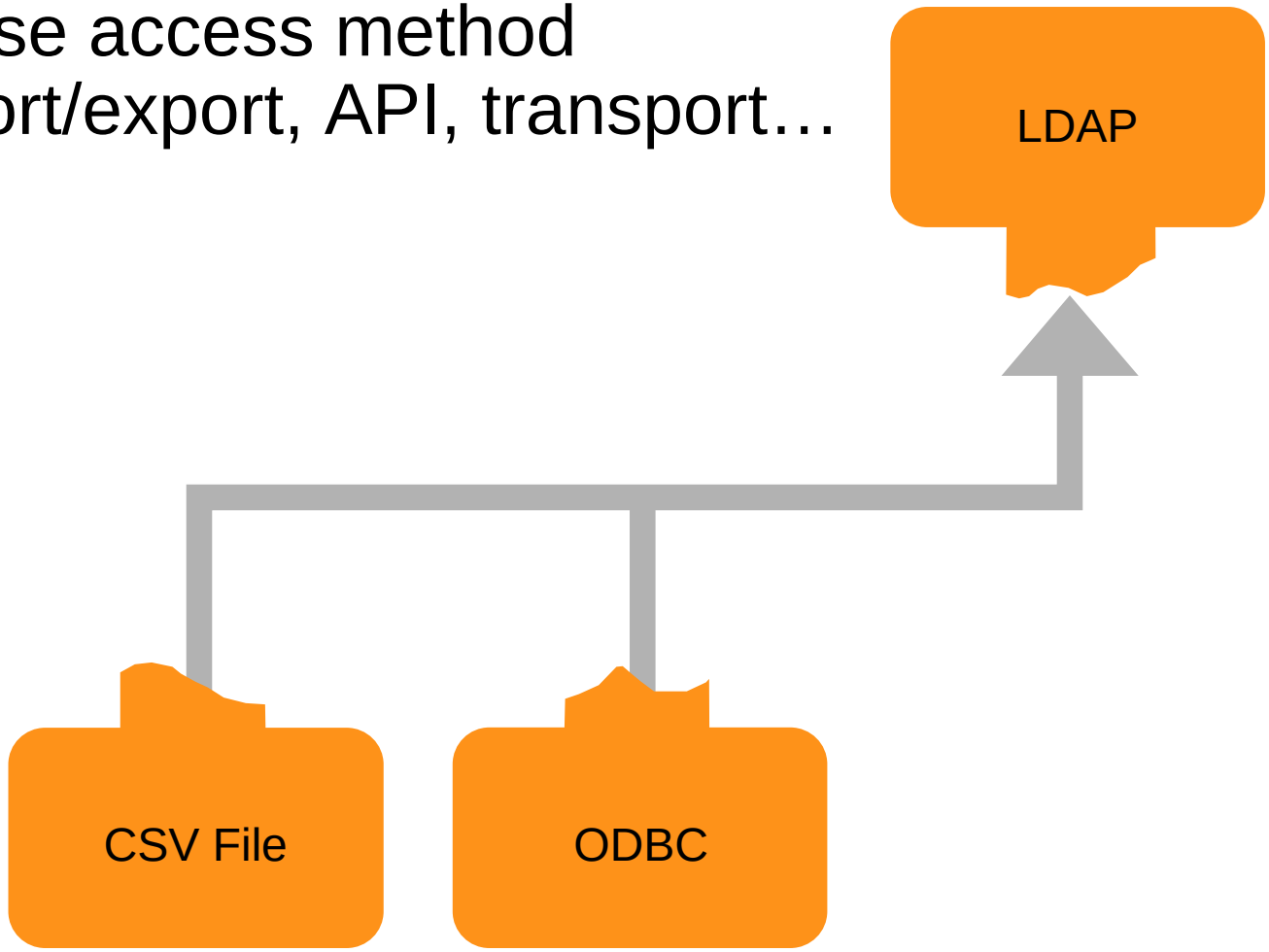
- Single, unidirectional flow
- From any to any



IBM

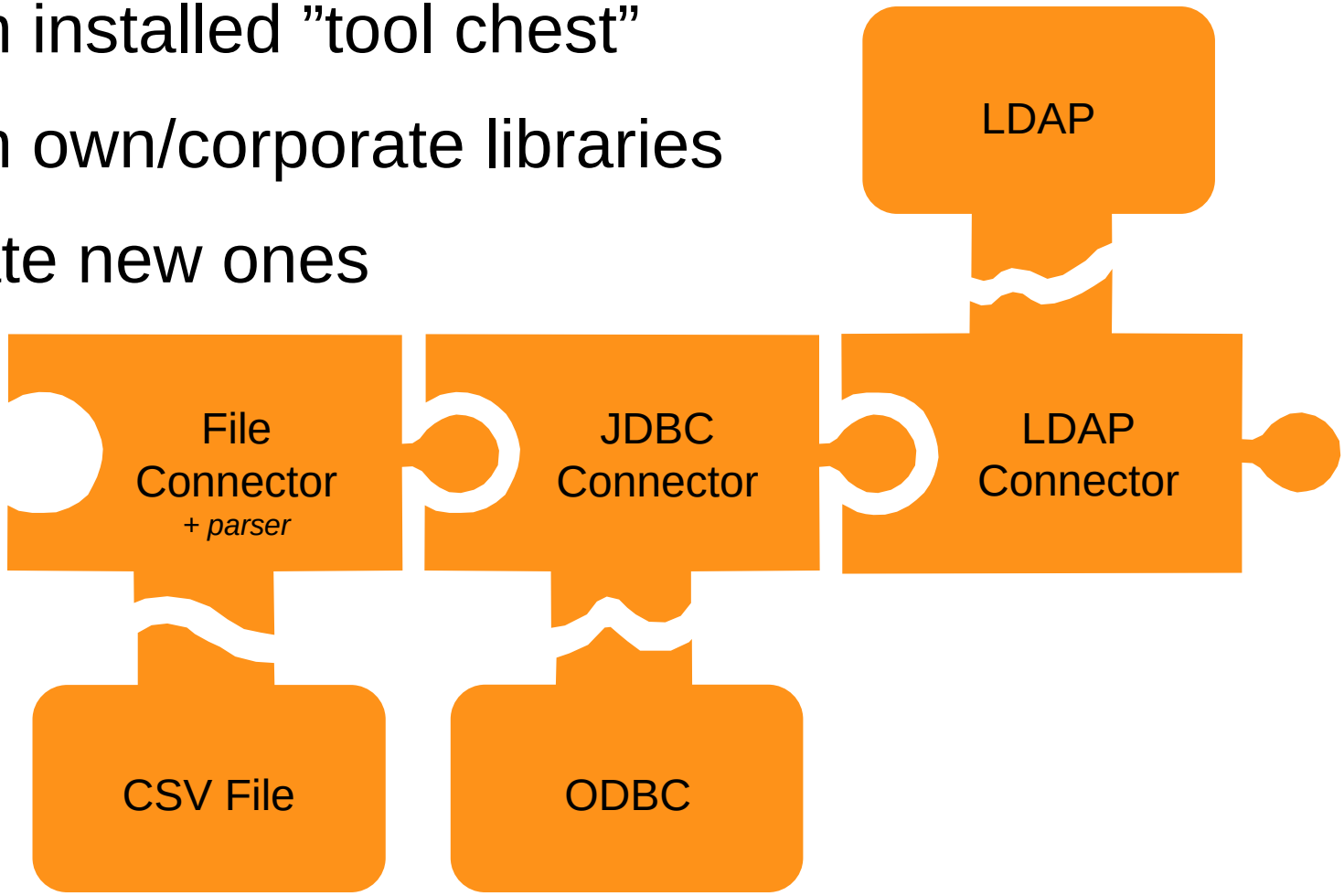
Identify Data Source Types

- Choose access method
  import/export, API, transport…



Click Together Components

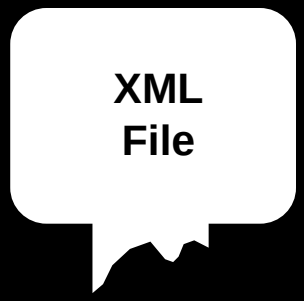
- From installed ”tool chest”
- From own/corporate libraries
- Create new ones



IBM

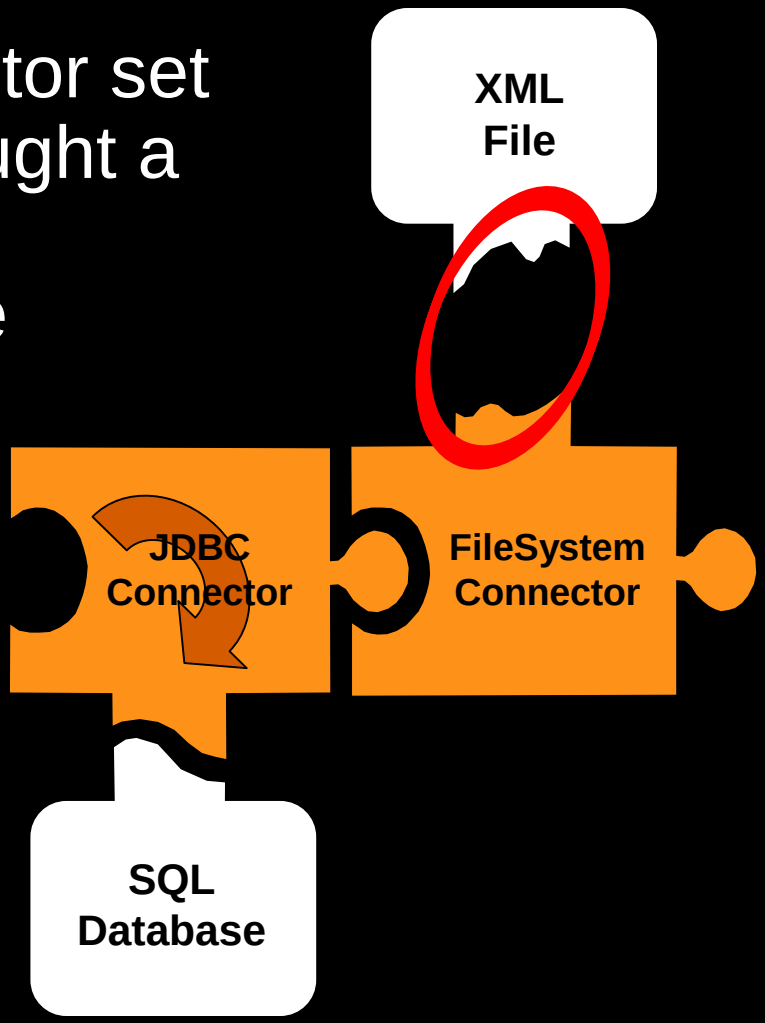
Simple Example

- Input SQL database (access through JDBC/ ODBC)
- Output XML document (write the structured bytestream to a file)



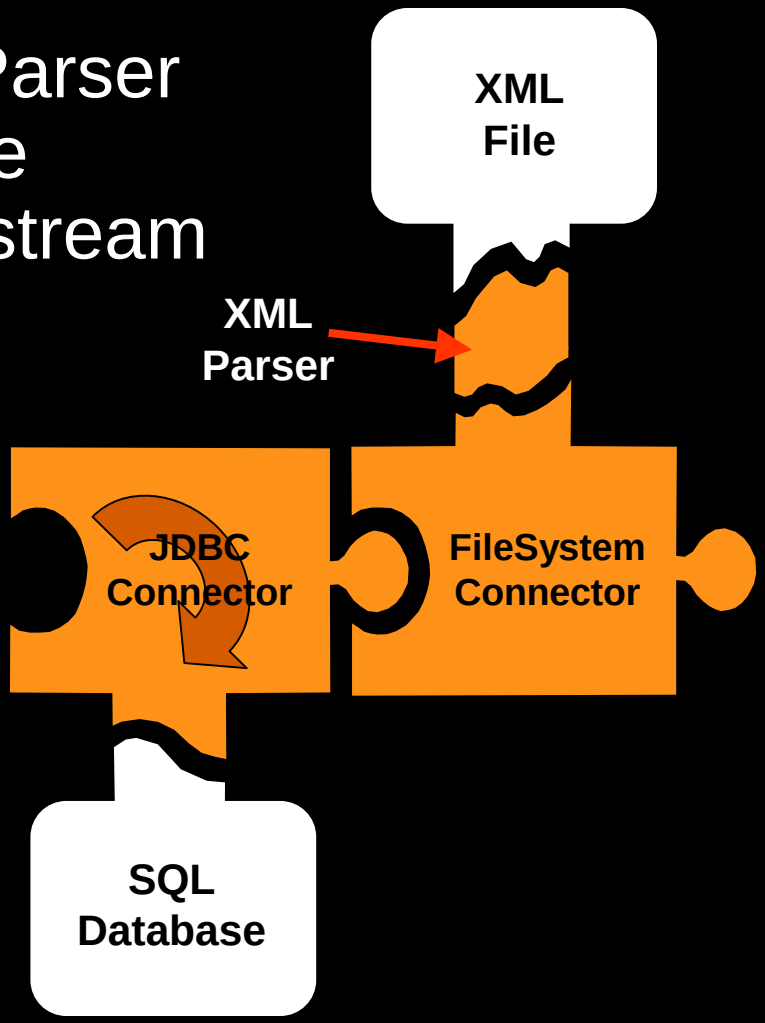
Simple Example

- JDBC Connector set to iterate throught a view in the SQL database
- FileSystem Connector for writing the XML file



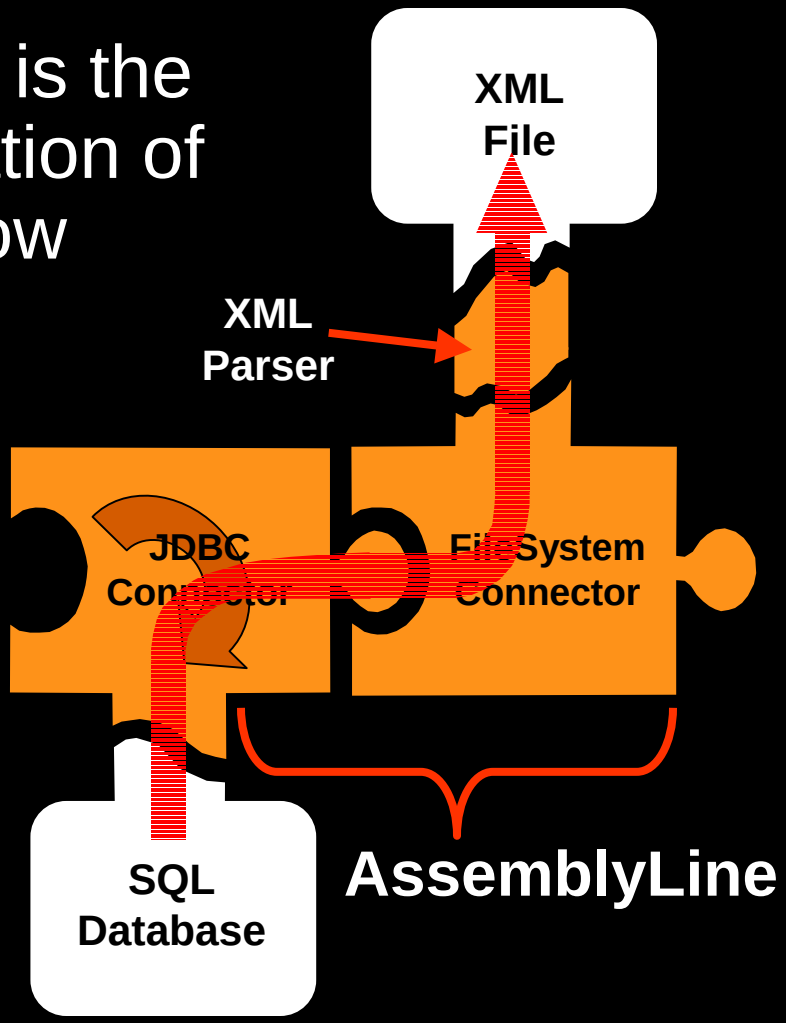
Simple Example

- Add an XML Parser to structure the outgoing bytestream



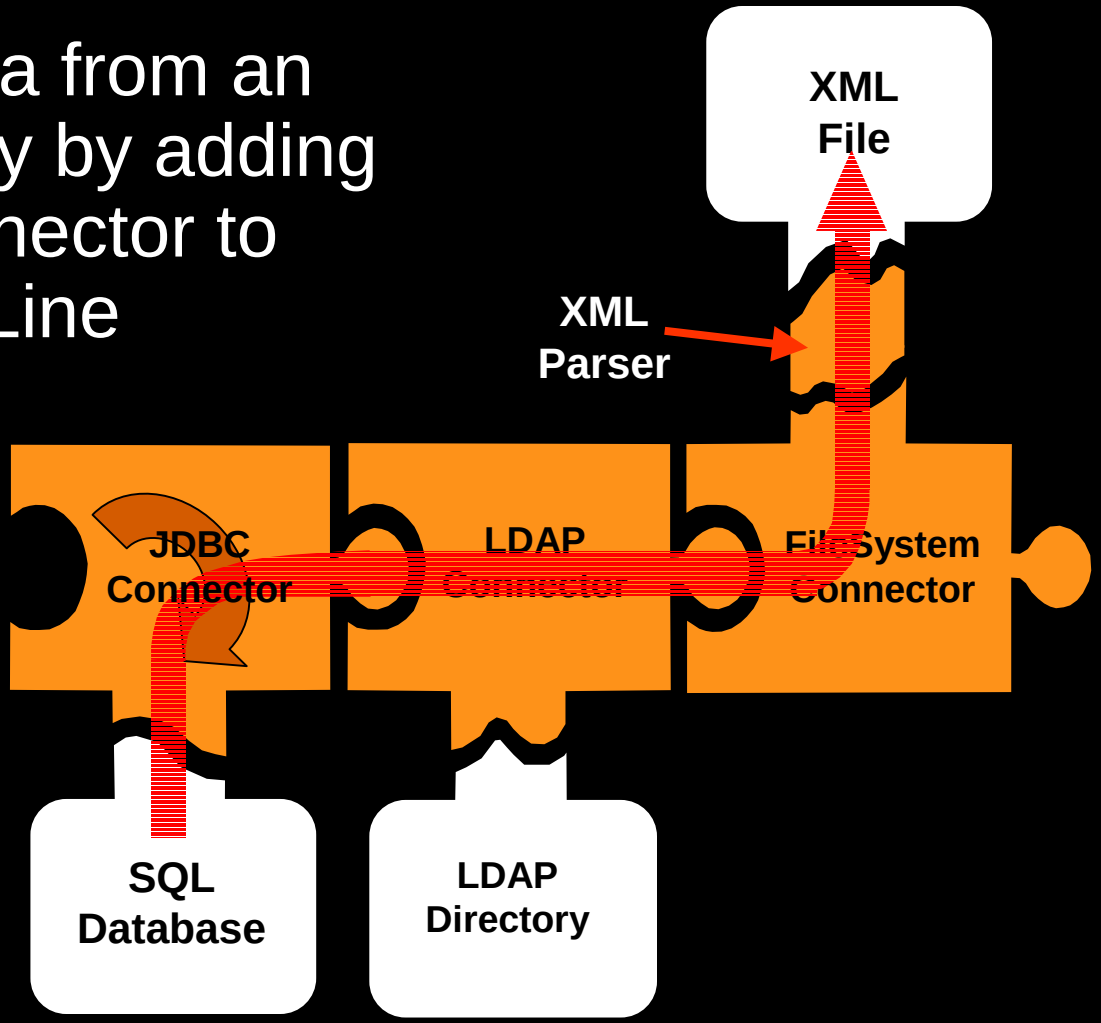
Simple Example

- AssemblyLine is the the implementation of a single data flow

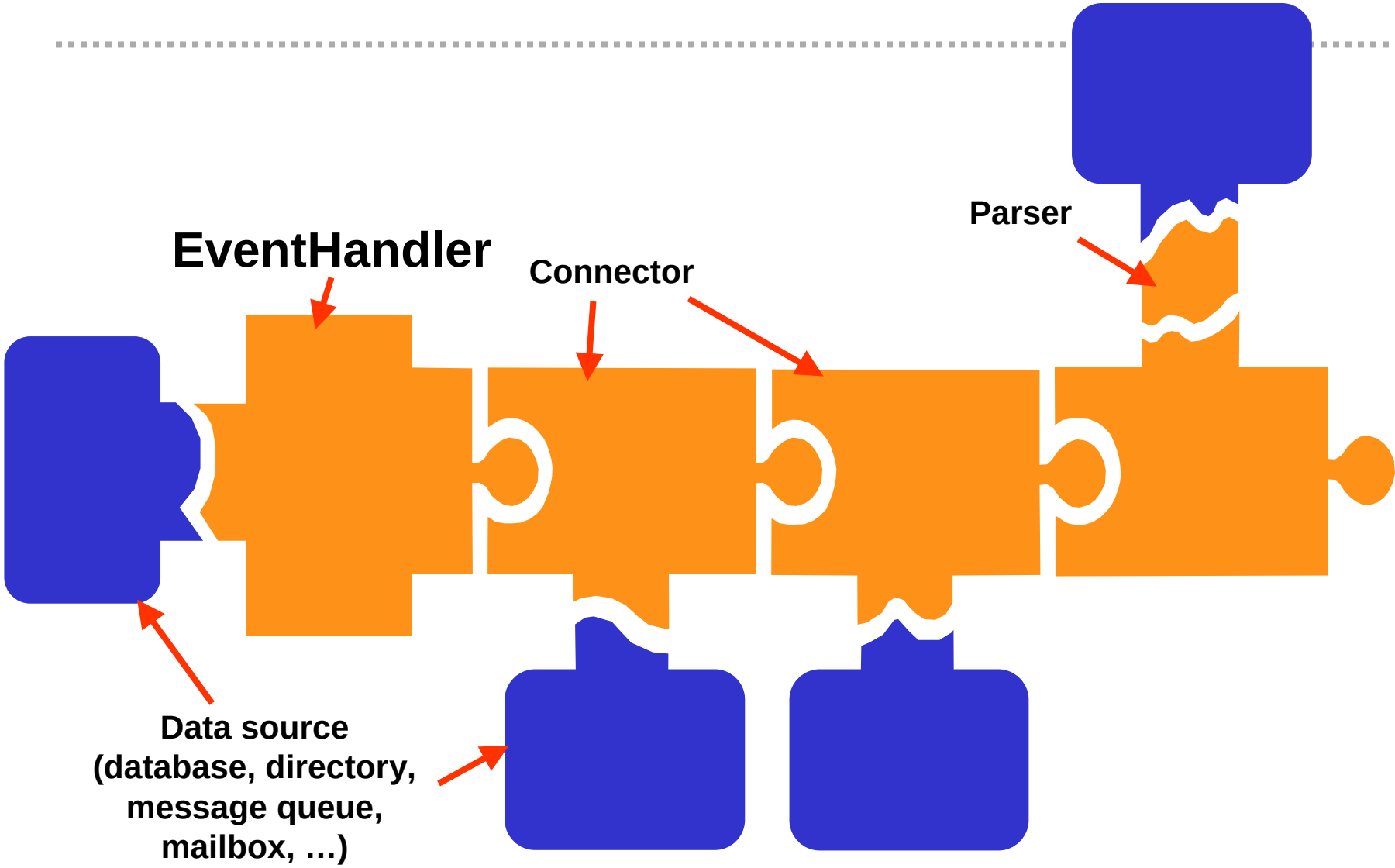


Simple Example (enhanced)

- Aggregate data from an LDAP directory by adding an LDAP Connector to the AssemblyLine



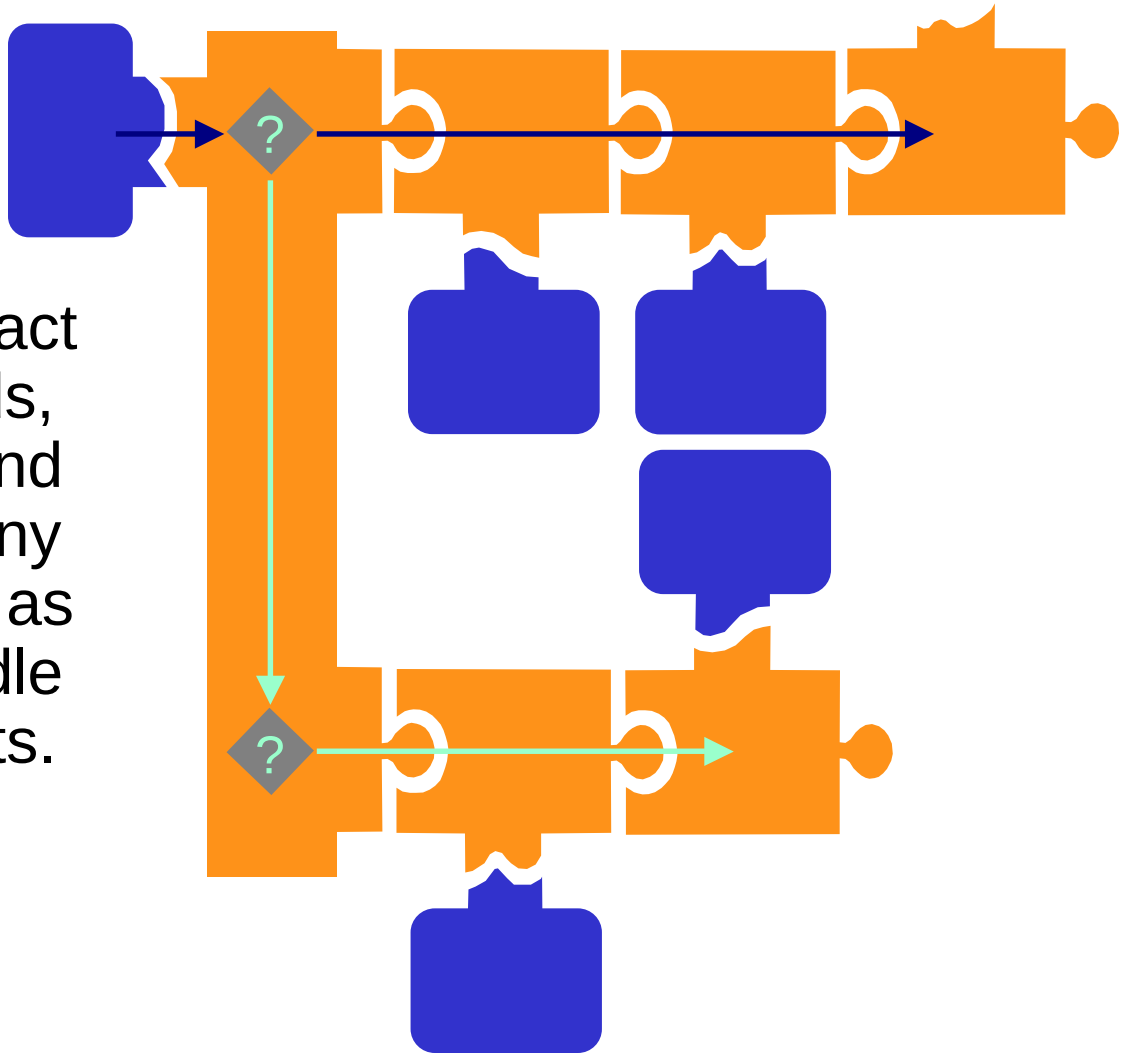
Event Handlers



IBM

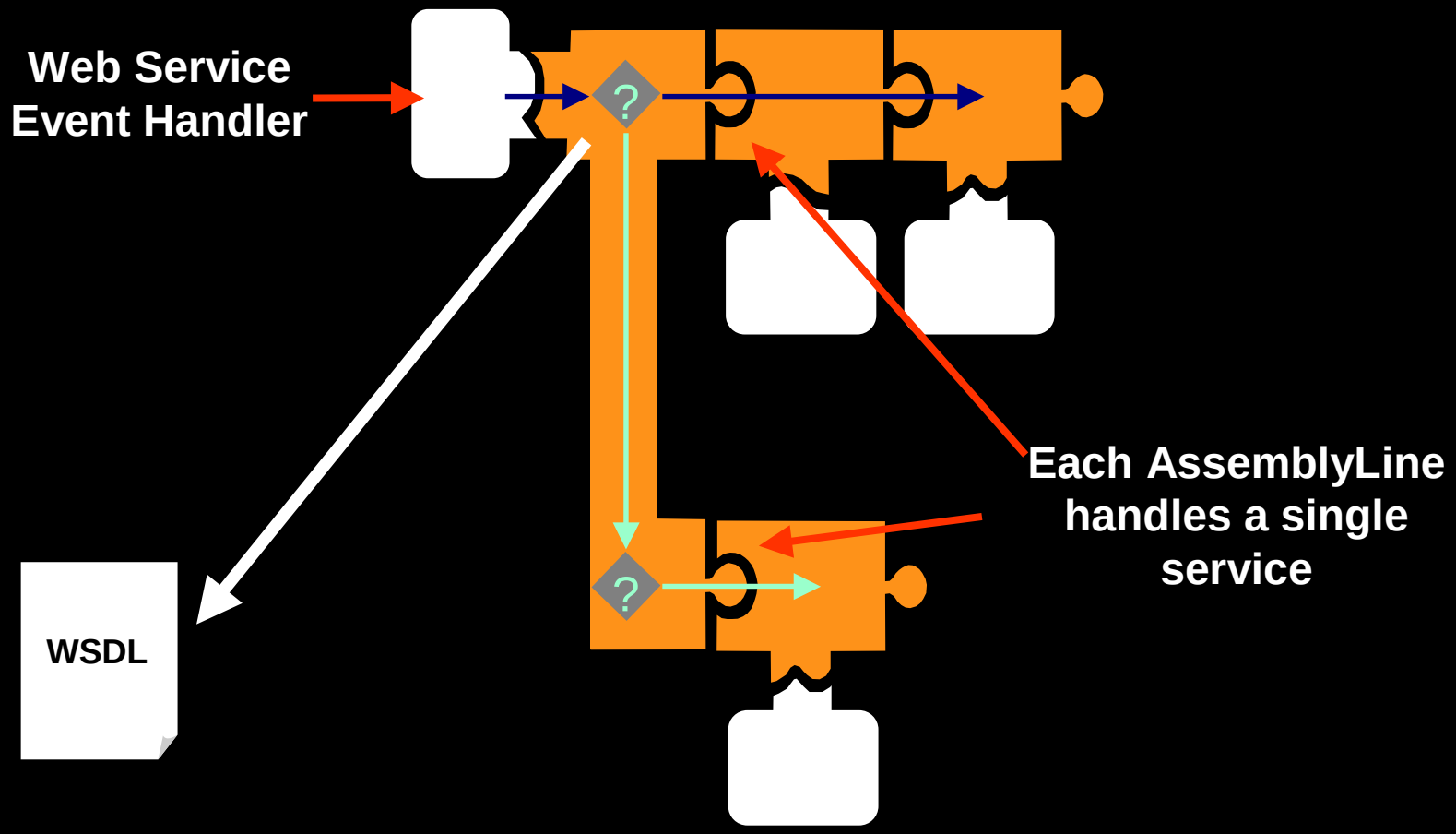
Event Handlers

EventHandlers act as switchboards, and can start and manage as many AssemblyLines as needed to handle incoming events.

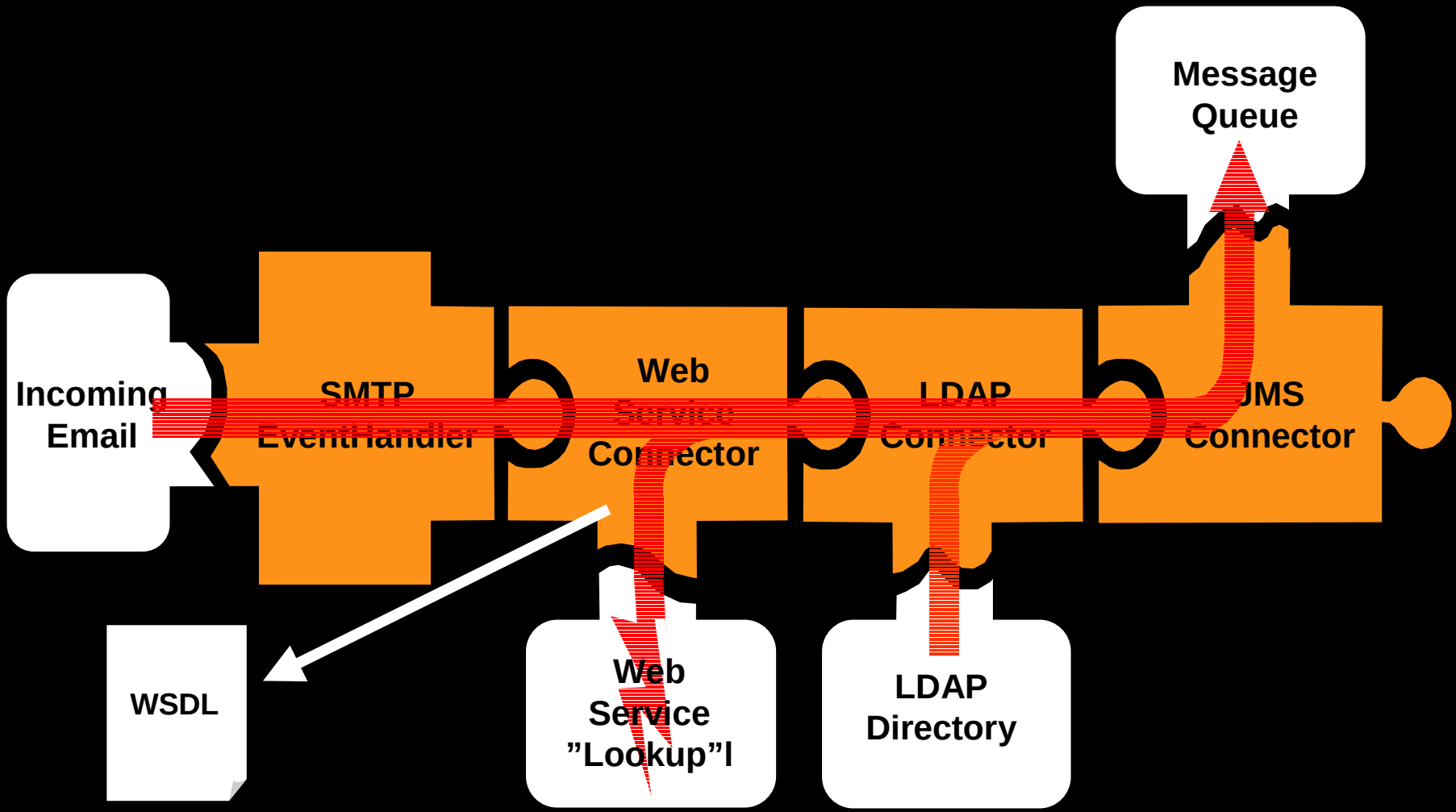


IBM

Web Services: Publishing



Web Services: Consuming



LUNDS
UNIVERSITET

Ldap vid LU

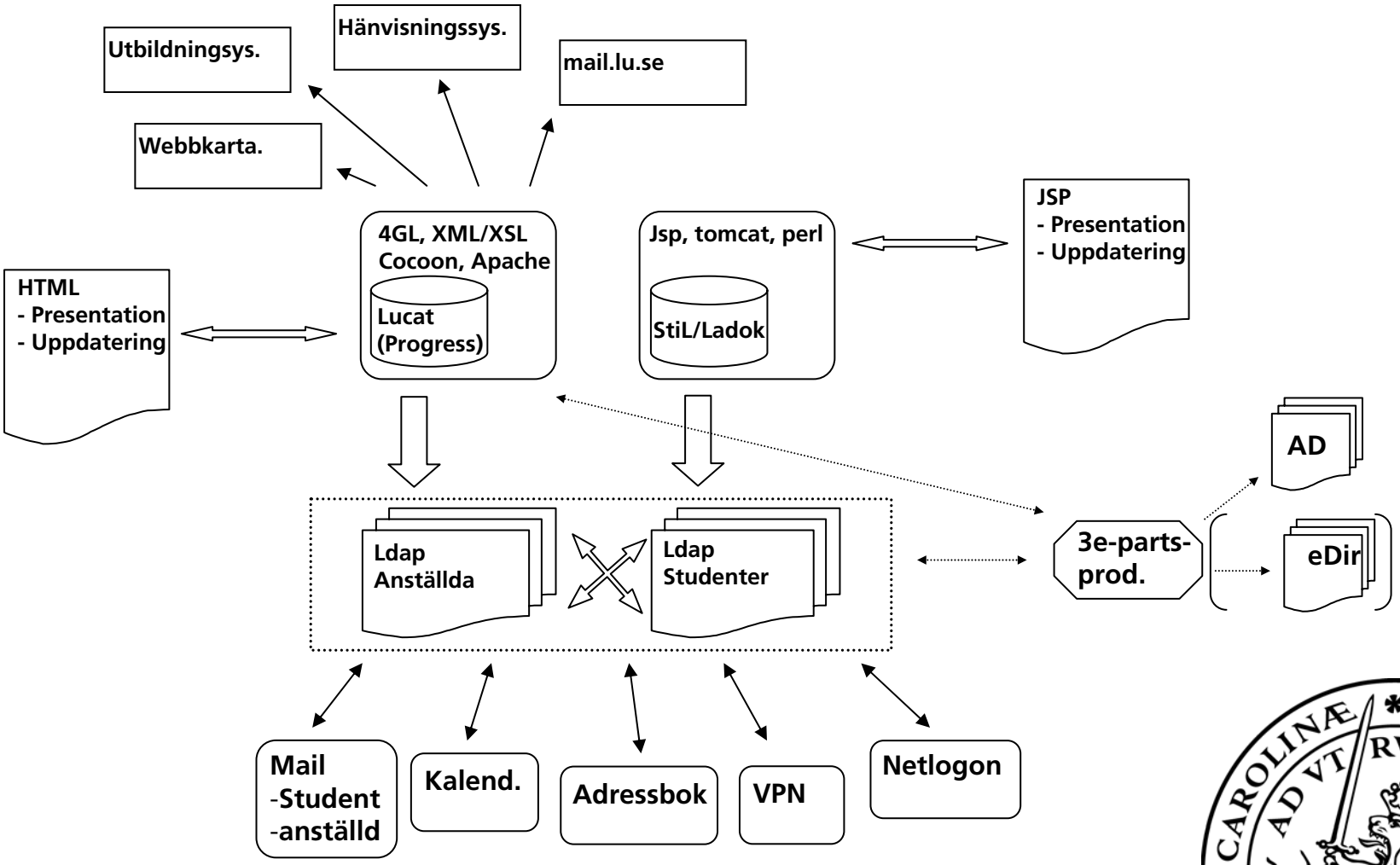
- Ldapstrukturen idag
- Kommande Ldapstruktur
- Informationsträdet
- Objektklasserna – eduPerson/luEduPerson

Informationsflödet med iPlanets Directory Server

- Anställda
  - Progress/4GL
  - Koppling mot:
    - Externa system 1 (Utbildningsys., webbkarta, växelsys.)
  - Ldap A
  - Mailsystem A
  - Kalender A

- Studenter
  - StiL/Ladok
  - Koppling mot:
    - Externa system 2 (EkH)
  - Ldap B
  - Mailsystem B
  - Kalender A

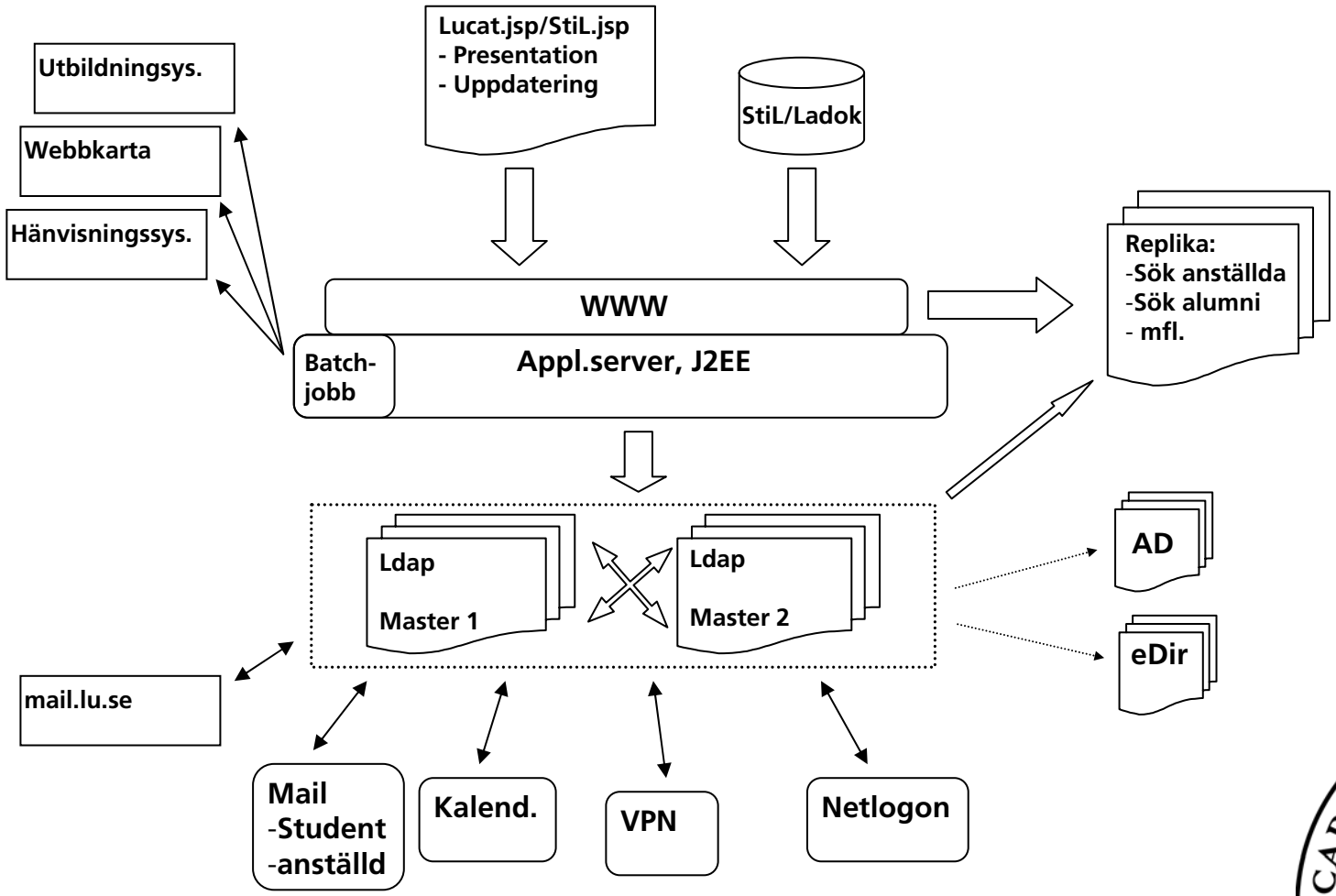
iPlanets Directory Server

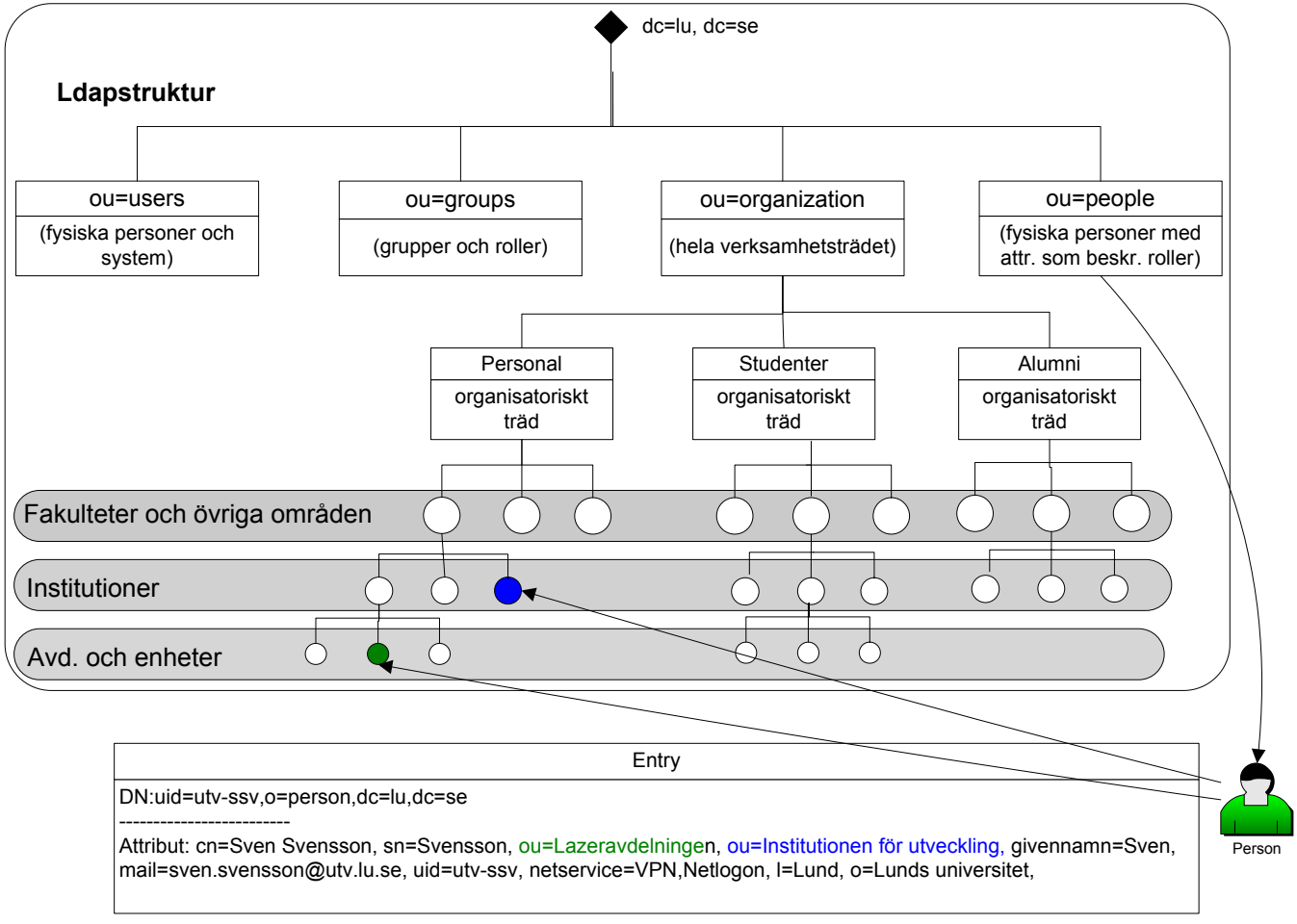


Informationsflödet med Sun ONEs Directory Server

- Ldap
  - Studenter, Anställda, Alumni
  - Javaplattform, J2EE, application server (JBoss)
  - Kopplingar, externa system – endast mot Ldap
  - Kalender
  - Mailsystem A och B
  - Ladok

Sun ONEs Directory Server




dc=lu, dc=se
Ldapstruktur
ou=users
(fysiska personer och system)
ou=groups
(grupper och roller)
ou=organization
(hela verksamhetsträdet)
ou=people
(fysiska personer med attr. som beskr. roller)
Personal
organisatoriskt träd
Studenter
organisatoriskt träd
Alumni
organisatoriskt träd
Fakulteter och övriga områden
Institutioner
Avd. och enheter
Entry
DN:uid=utv-ssv,o=person,dc=lu,dc=se

Attribut: cn=Sven Svensson, sn=Svensson, ou=Lazeravdelningen, ou=Institutionen för utveckling, givennamn=Sven, mail=sven.svensson@utv.lu.se, uid=utv-ssv, netservice=VPN,Netlogon, l=Lund, o=Lunds universitet,
Person

luEduPerson

luAcademicDegree
luCareof
luAdminEvent
luHomeCountry
luInstantMessage
luMyMobile
luNetService
luPersonNumber
luKursluProgram
luStudentNation
luPrivacy
luQuota
luLastMailcheck
luSparrbeskrivning
luStudentForening
luStudentInfo
luStudentKar
luTemporaryAddress
luPrivousName

UPPSALA
UNIVERSITET

Directory related activities at Uppsala University

CodeX - EDS2003 2003-01-29

Pål Axelsson, Uppsala University
Pal.Axelsson@its.uu.se

UPPSALA UNIVERSITET

Agenda

Current applications

- Uppsala University Directory
- Authsrv
- Active Directory
- Certificate Authority (UU-CA)

Current projects

- Personal Certificate Authority (UU-Personal-CA)
- A2K2

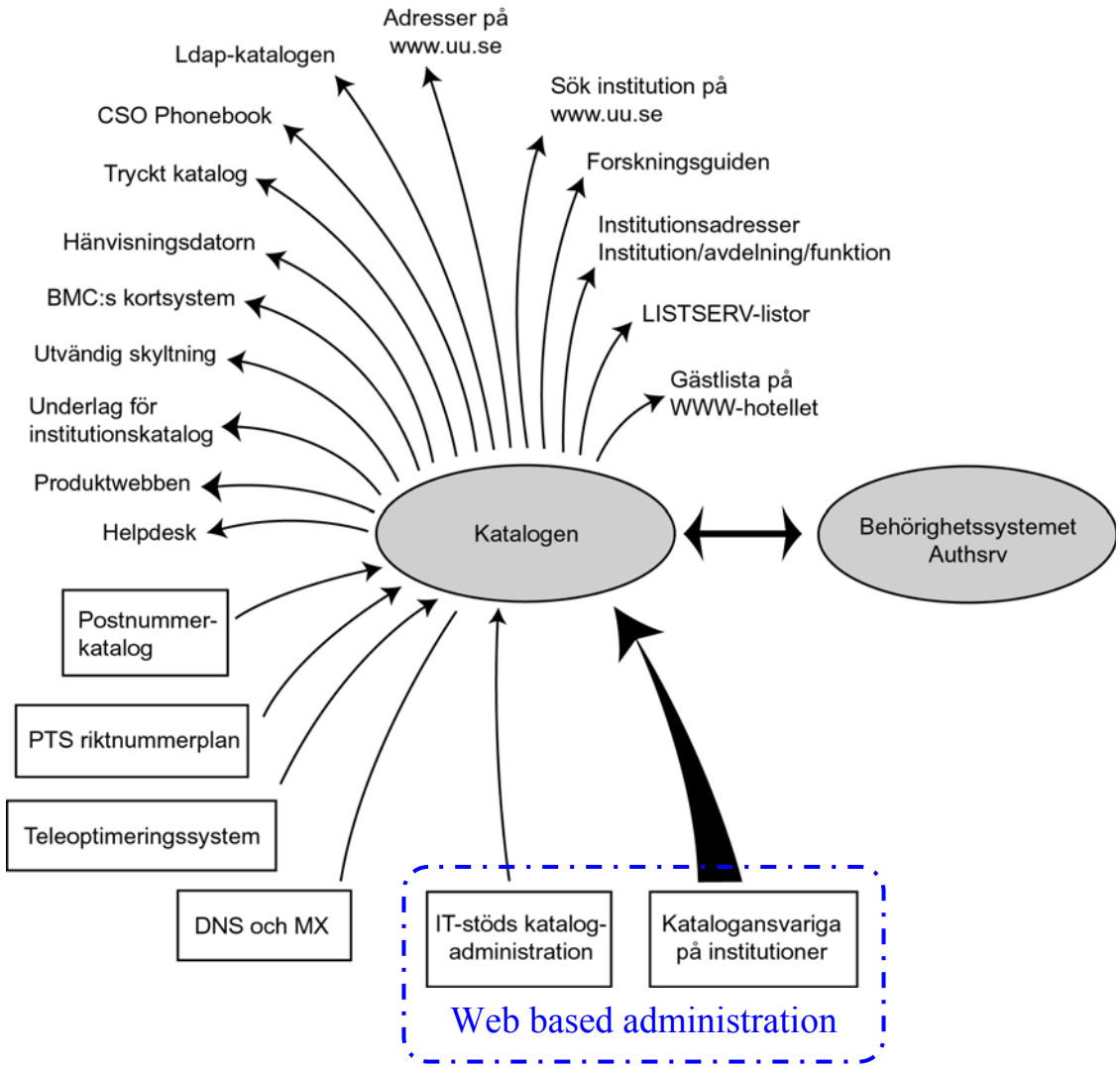
UPPSALA UNIVERSITET

Uppsala University Directory

- Developed in the early 90's and was operational in 1994
- A directory of employees and affiliates
  - 9500 individuals (2003-01-28)
- Includes UU's organizational structure and information for web based information resources and the printed university catalogue
- Decentralized administration
- Vital part of Uppsala University's infrastructure and a information resource for other vital systems
  - Uppsala University Authentication system (Authsrv)
  - Call management systems
  - Electronic directory services (LDAP and CSO Phonebook)

UPPSALA UNIVERSITET

UU Directory external connections



UPPSALA UNIVERSITET

How to populate a university directory

- Human resource system
  - Doesn't include affiliates
  - Correct national identity number
  - Name and home address is only as good as needed
- Call management system
  - Not all persons have a telephone at UU
- E-mail system
  - In 1992 only a small percentage had e-mail
- Printed university catalogue
  - Includes employees and affiliates
  - Is correct at the time of production
- Solution at Uppsala University
  - First populated from the manuscript of the printed catalogue and information was added semi automatically from the human resource system and the e-mail system
  - Web based administration is used during normal operation

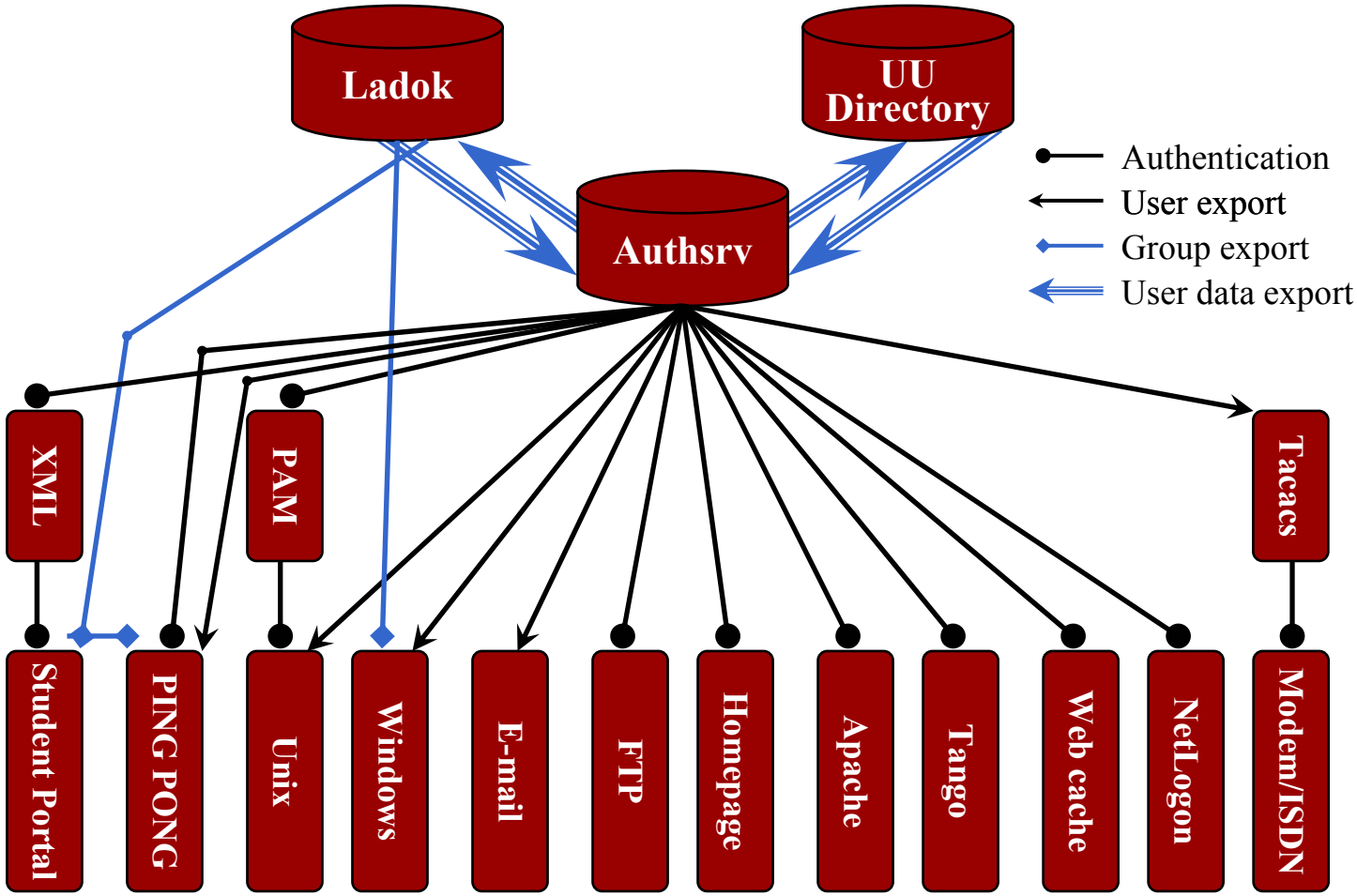
UPPSALA UNIVERSITET

Authsrv – Authentication Service

- Developed in 1995 for the new student network (e-mail and connections)
  - Students are exported daily from Ladok
- Worked so well that it was expanded to include employees and affiliates in 1997
  - Exported daily from UU Directory Service
- Other services than e-mail and connections has been added on demand from 1996 and forward
- Authsrv updates it's authentication services twice daily

UPPSALA UNIVERSITET

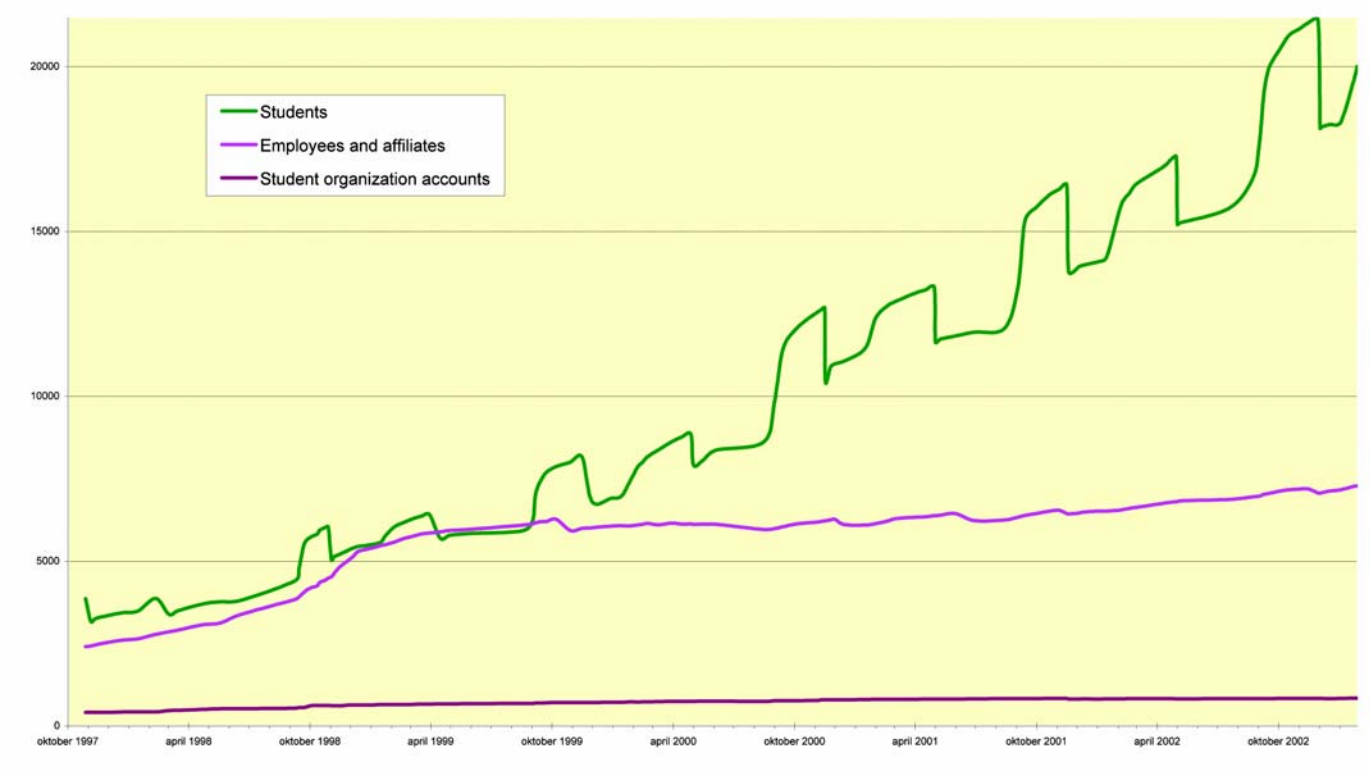
Authsrv Authentication Structure



UPPSALA UNIVERSITET

Authsrv Statistics (2003-01-28)

- 20 000 active student accounts
  (60% of all students; 55% of all accounts used by females)
- 7 200 active employee and affiliate accounts
  (76% of all employees and affiliates)
- 800 active student organization accounts



UPPSALA UNIVERSITET

Uppsala University Active Directory

- Operational in September 2000
- Student accounts imported daily from Authsrv
- Registered students on a course are imported daily as an AD group from Ladok
- Is used with MS Certificate Server for UU-Personal-CA
- The UU-AD is also used as a resource directory for windows based resources at Uppsala University

UPPSALA UNIVERSITET

UU Certificate Authority (UU-CA)

- X.509 based Offline Certificate Authority for server authentication (SSL / TLS)
- 1st approved member of SwUPKI in Q1 2001
- Important building block for a more secure infrastructure
- Statistics
  - 38 active certificates
  - 28 revoked certificates
  - 3 expired certificates

http://uu-ca.uu.se

UPPSALA UNIVERSITET

UU-Personal-CA

- Ongoing development project since spring 2002
- Online Subordinate CA to UU-CA
- Issuing of personal certificates for authentication with smart cards
- Our goal is to start simple and carry on
  - Build the infrastructure
  - Operating system smart card logon
  - Web application smart card authentication
  - Test built-in platform and application support

UPPSALA UNIVERSITET

Automatic certificate issuing

- Automated process with account activation and certificate signing
  - Usage of a public identity smart card
    - Contains the national identity number
  - Identifying the user automatically
  - Allows issuing against student registry records
  - Allows issuing against Uppsala University Directory for employees and affiliates
- A smart card based automated process can make account administration more effective

UPPSALA UNIVERSITET

Import lessons with smart cards

- Write a CPS and get it approved
- Logistics with a subordinate CA's
- CRL-distribution
  - Critical and underestimated
  - On different servers (redundancy)
  - Verified CA and CRL-distribution chain
- Software bugs
  - Smart card hardware drivers
  - Smart card administration utilities
  - Smart card handling in applications
- You need to have an AA system

UPPSALA
UNIVERSITET

A2K2 (development label)

- Ongoing development project in 2 planned phases
  - Authentication, authorization and e-mail administration
    - Delivery for Uppsala University in June 2003
  - University Catalogue
    - Delivery for Uppsala University in March 2004
- Supersedes Authsrv and Uppsala University Directory
- Inherit functionality from the old systems
- Being built with standard tools
  - SQL, LDAP, C, C++, Java, Soap and HTML

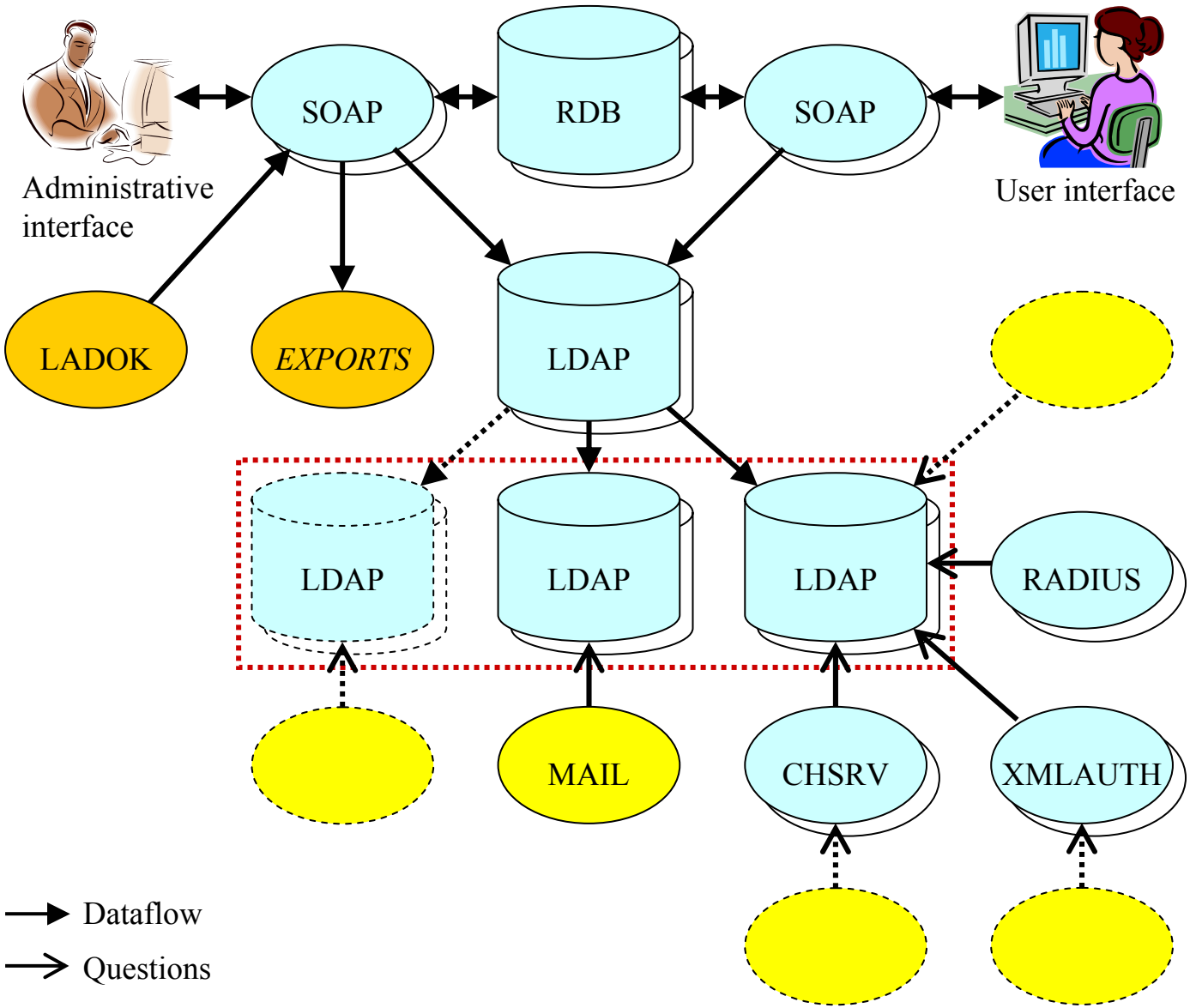
UPPSALA UNIVERSITET

What's the big news with A2K2

- Two systems are going to be one system for employee directory, authentication, authorization and e-mail administration
- Support for one individual one account
- Support for SPOCP
- Support for LDAP based authentication
- Support for PKI
- No unnecessary delay from changing information in the system until it is available for all depending systems.

UPPSALA UNIVERSITET

A2K2 Technical Architecture



UPPSALA UNIVERSITET

Conclusions

- We can build a more secure environment with directories
- We can save money on directories
- We need to have an authentication and authorization system to make wide spread PKI for users feasible
- We haven't found any commercial systems that satisfies all our needs
- It's harder to blend old systems into a new system

Enterprise Directory at Umeå University

www.umu.se/it/umu_internt/entdir/EDC2003.pdf

Torbjörn Wiberg
CIO, UmU

Self-Service in IT Applications

- Some trends are:
  - Increased Self-Service in IT Applications
  - The non-specialist user has web access to the Application
- This tends to make all our students and/or all our personnel (non-specialist) users of more and more of our systems
  - Tur och Retur
  - Ladok på webb
  - Personal portals
  - Ping-Pong

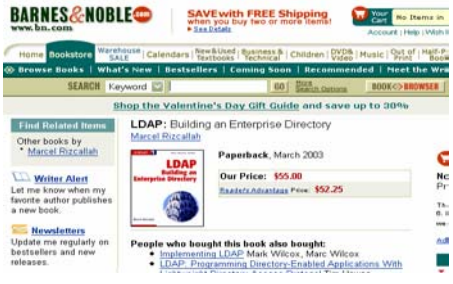
The Role of an Enterprise Directory

- Problem:
  - It is expensive and hard to maintain good quality user account systems separately for each application
  - People become members of and leave (as "friends or enemies") the university community
- Make life easier by using an Enterprise Directory to support the maintenance of user accounts. Giving support for
  - Authentication – identification of a person or other entity
  - Authorisation – decision of what an authenticated entity is allowed to do in an application

What Can You Expect to Find in an Enterprise Directory

- Traditional Directory
  - White Page information about personnel (and perhaps students) accessible through LDAP (an access protocol)
  - Name, phone#, email, organisational unit, position, …
- An Enterprise Directory
  - Information to support Authentication and Authorisation.
  - White page information about each person in the community
  - Information about identifiers used
    - Minimize the number of identifiers
    - Including authentication credentials such as identity certificates (?), and username/"passwords"
  - Delegation and Appointment information
    - Administrative
    - Course responsibilities…

Not Invented Here, There Are Books and White Papers



Michigan Technological University – Identifiers for Persons/Groups

- MTU: Userid/login
  - Generic: account login
- MTU: Unix UID
- MTU: E-mail Address
  - Generic: email address
- MTU: MichNet Modem login
  - Generic: netid
- MTU: Tech ID
  - Generic: publicly visible id (PVI)
- MTU: Card Chip ID
- MTU: TechExpress Id
- MTU: PIDM
  - Generic: Unique Identifier
  - admin. syst. Id
  - person registry id
- MTU: Oracle id
  - Generic: admin. sys. id
- MTU: SS#
  - Generic: SS#
- MTU: Group id
- MTU: GID
- MTU: Library id
  - Generic: library/departmental ID
- MTU: Library Courtesy Card
  - Generic: library/departmental ID
- MTU: Subscriber ID
  - Generic: library/departmental ID
- MTU: Summer Youth id
  - Generic: library/departmental ID
- MTU: Summer Athletics id(s)
  - Generic: library/departmental ID

eEurope, eBusiness, eLearning ...

- They all depend on You being able to show to the system that You are You
- You show this by presenting your electronic identity to the system
- There are electronic identities of different strengths – levels of security
  - Registration of Username/Password by e-mail or on a web page
  - Registration of Username/Password after strict identification
  - Smart Cards with Public Key Cryptography
- Systems should require a higher strength to authorise us to do more sensitive operations

Electronic Identity

- You normally **choose** your electronic **identity** yourself
- One approach:
  - Establish the binding of the identity to You by securely presenting Your identity to a **trustworthy organisation** that issues a **certificate** stating that the electronic identity is Yours
- (Trust has to be earned)
- Every user of our systems has to have an Electronic Identity (at least username/password)
- The user community includes students and personnel.

Electronic Identities in a PKI

- In a PKI (Public Key Infrastructure), the **Identity** is a pair of **crypto keys** for Public Key Cryptography.
- A pair?
  - What is encrypted with one key can be decrypted with the other and vice versa.
  - The private key is kept secret by the owner and the public shall be made as widely known as possible.
- Identity?
  - I prove my identity be encrypting a message, which then only can be decrypted with my public key.
- In a PKI, certificates of public keys are stored and distributed

How Can an Electronic Identity Be Used? – For Authentication

- It can be used to **authenticate**
  - you (to systems)
  - documents or messages you have digitally signed (PKI based eID)
  - systems you are responsible for, to users (PKI based eID)
- **Authenticate** – establish the originality of
- **Non-repudiation** – A process or method that ensures that once you have signed a document or identified yourself to a system you can't deny that (PKI based eID)
- **Authentication** – can be realised as a middleware service
  - Requires a PKI and/or a Username/Password database
  - Implemented as a server or plug in

Authorisation

- **Authentication** – establishes identity to a certain strength
- **Authorisation** – controls what you may do
  - Policy Control, Access Control
  - Once authenticated and depending on the strength of the authentication you will (not) be authorised to do ...
- Authorisation – can be realised as a middleware service
  - Requires a high quality Enterprise directory to be really valuable
  - Can be implemented as a Server or plug in
- **Note!** - What from a simple application is considered authentication, is from an enterprise perspective an authorisation to use that application!

Enterprise Directory - Again

- More than a telephone book or an e-mail directory!
- It contains every person affiliated with the organisation -
  - Definition: The chairman must pe prepared to present the list to the dean and say:
    - This is my personnel
    - These are our students!
- Attributes of relevance for authorisation shall be registered
  - The maintenance shall reflect the delegation of responsibility
  - If for ex authority follows with being a chairman, the assignment of that attribute shall be done by those who appointed her

But…

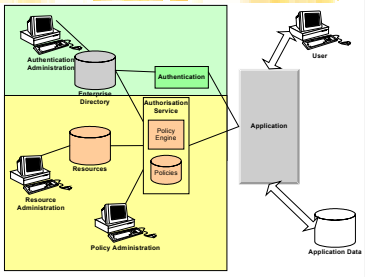
- … the users of an Enterprise Directory are users with different needs and authority
- … all information in the directory must not be available through an anonymous LDAP-request
- !We need an authorisation system to the Enterprise Directory
  - What attributes shall on what grounds be made available to what application (privacy issue, and organisational security issues)
  - The authority to maintain the Enterprise Directory shall reflect the delegation of responsibility
  - If for ex authority follows with being a chairman, the assignment of that attribute shall be done by those who appointed her

Vision for Authn/Authz Services

- We are all members of several virtual communities and will increasingly expect to be able to use the same authentication credentials in all of them.
- AA model:
  - Commonly accepted, federated authentication services
    - Federated - the authentication is done where the user belongs and the result is communicated in a uniform format
  - Authorisation attribute servers – enterprise directories
    - Web Services will require "standardised" external attributes – eduPerson as a base
    - I believe Authorisation will use attribute servers
  - Policy based, trusted authorisation service
    - controlled/accepted by the individual web service and vice versa
    - using attribute servers

How Do We Realise Authn/Authz? As Network Based Infraservices!

- By that we mean →
- The green part is the Authentication Service
- The yellow part is the Authorisation Service
- They shall be separate services
- The grey is the Enterprise Directory
- The application may or may not be a Web Service
- These are Infraservices (Middleware Services)
- We need to cooperate to realise such a middleware structure!



Directory/Authn/Authz Services at UmU Today

- We have implemented
  - an isolated LDAP directory with a limited schema structure
  - an ID server with name, id#, email, account name and course registration information – synchronises with Ladok
  - an Active Directory of students
  - a TACACS authentication system (not encrypted, username/password)
- We have
  - experience with the Authorisation Service SPOCP
  - done some preparations for the definition of a richer LDAP schema
  - started to look at available Authentication Service software, Metadirectory tools (Metamerge Integrator)
  - started to discuss Authorisation policies for the Enterprise Directory

Directory/Authn/Authz Services at UmU Today, cont

- We have not yet
  - chosen directory technology
  - initiated a directory project (schema, ownership of data etc)
  - initiated an authentication project
- A long way to go and we are expected to be up and running in the autumn!

Authentication Service

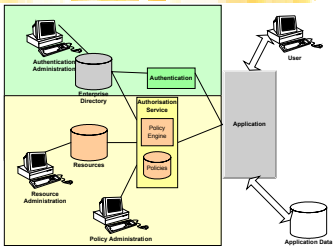
- We have a PKI (certificate infrastructure) in place
  - for those who want to use Public Key Cryptography
  - Policy CA – Umeå Universitet
  - CA Operation Service – Uppsala Universitet, Stockholms Universitet, Umeå Universitet ..?
- We need Software for an Authentication Service supporting at least use the authentication mechanisms:
  - Username/Password
  - PKI supported Public Key Cryptography
- There are a couple of alternative choices:
  - CAS – Yale University
  - Feide AT – Uninett as a partner in SPOCP

SPOCP – Authorisation Server Software for Higher Education

- (SPOCP – Simple Policy Control Project)
- A project to develop/provide a network based infrastructural service for authentication and policy based authorisation. 02-06-01 -- 03-05-31
- Partners
  - KI, LU, SU, Umu, Uninett, UU
- Financing - national funds and member fees
  - Sunet and NyA- funds
- Licensing
  - Open source
  - Commercial and non profit licences

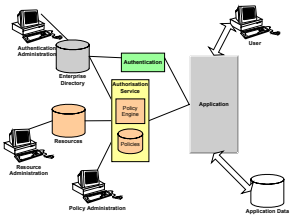
(Web) Service Model

- The web server authenticates the user to a certain strength
- When the user wants to do something in the application (have access to a resource), the application server asks the authorisation server for advice
- The authorisation server receives the request and, using the policies for the application and information about the resource and the user, does (not) recommend access
- The application does (not) grant access to the resource



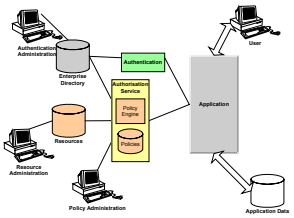
SPOCP - Deployment Models

- serve **one** particular application at **one** particular university
- **bundled** with **one** application that is deployed independently at several universities
- use a general role data base for a university and serve **several** smaller web applications at **one** university
- serve **one** application that has users in **different** universities



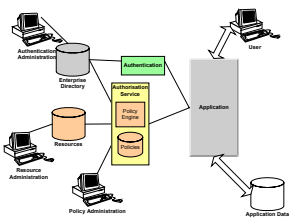
SPOCP - The First Model

- Serve **one** particular application at **one** particular university
  - An instance of the Authorisation Service is integrated with an application
  - The communication with the application may be on a secured local network or integrated with the serviced system
  - For instance as the authorisation system of NyA – the new admission system.



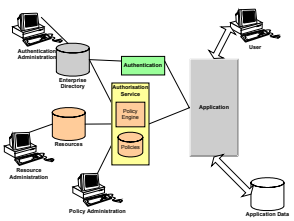
SPOCP - The Second Model

- **Bundled** with one application that is deployed independently at several universities
  - For instance serving "Ladok på web"



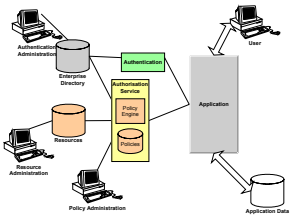
SPOCP - The Third Model

- Use a general role database for a university and serve **several** smaller web applications at **one** university
  - Personal portals
  - The smaller applications are calendar, file access, mail, bookmark stores, course material stores etc.



SPOCP - The Fourth Model

- Provide authorisation advice to ONE application that has users in different universities
  - One coordinated network logon for students, irrespectable of from which university they come
  - Library or other content access
  - Network University



Prerequisites for SPOCP

- You need a carefully designed and maintained enterprise directory
  - Rules for who belongs to different (internal) communities
  - Authority for assigning and rules for having roles
  - Rules for assignment of identifiers/account names
    - Who may assign an identifier
    - What identifiers shall we have
    - Is it allowed to reuse an identifier
- Policies for authorisation to use systems. Policies?
  - Based on roles and identity
  - Definition of authorities
  - Rules for delegation of authority
  - Conditions on authority relative the environment, for ex rules about office hours or cryptographic protection of the communication.

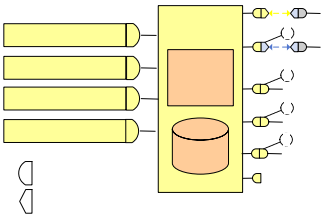
Authentication and SPOCP - The Approach

- We assume that the Authentication Service developed in Feide can be used for those who wish
  - Username/Password
  - X.509 Certificates
  - Federated Approach
  - We can use Swupki certificates –
    - UU tests a model to automatically issue certificates to persons registered in their AD
- Any other Authentication Service can be used!
  - it is a completely separate but equally necessary service

SPOCP – The Approach

- The project focuses on the Authorisation Server
- Policy based authorisation
- Policies are rights, expressed as restricted S-expressions
  - the Policy Engine does not know anything about the application
  - the application must be able to articulate authorisarion queries
- Authorisation query format convention:
  - Given these Conditions- May this Actor do this Operation on that Target
  - but the Policy Engine works equally well for other conventions
- Appointment and delegation of authority shall also be policy controlled

SPOCP – Implementation Status



SPOCP – Applications

- NyA
- Ladok på Web
- Authorisation to access info in directories in the Feide Authentication Software – discussed
- Mail "terminals" to control mail relaying
- uPortal

How Can You Become a SPOCP Site?

- We will not add organisations during the project phase
  - I want to discuss future maintenance cooperation
- We have beta released SPOCP as Open Source
  - with dont yet have support for policy administration and for delegation and appointment

And – You Need a Good Enterprise Directory

- If SPOCP sound interesting, start preparing!
- We need a new approach to directories
  - A policy based authorisation system (bootstrap!!)
    - Policy for updates of attributes
    - Policy for attribute release
    - eduPerson attributes as edge attributes
  - Based on a meta directory
    - Metamerge for ex – system for converting and synchronising between data sources
- Already this is a mouthful!

Enterprise Directory at Umeå University

www.umu.se/it/umu_internt/entdir/EDC2003.pdf

Torbjörn Wiberg
CIO, UmU

LUKAS

Katalogtjänst@LiU

Mattias Carlsson

Systemadministratör

mattias@unit.liu.se

013-28 1753

LUKAS
Historik

- LDAP sedan sommaren 2000
- LUKAS-projektet
  - Linköpings Universitets Katalog för Anställda och Studenter
  - Två delar, teoretisk och praktisk
  - Ny LDAP igång 2002-12-20
  - Bara anställda i första vändan

LUKAS
Mål med ”omstarten”

- Ge alla anställda ett gemensamt användar-ID
- Centralt grepp på katalogtjänster
  - Centraliserad katalog för rättigheter
  - White pages
- Strukturera upp objektklasser, attribut och ACIer

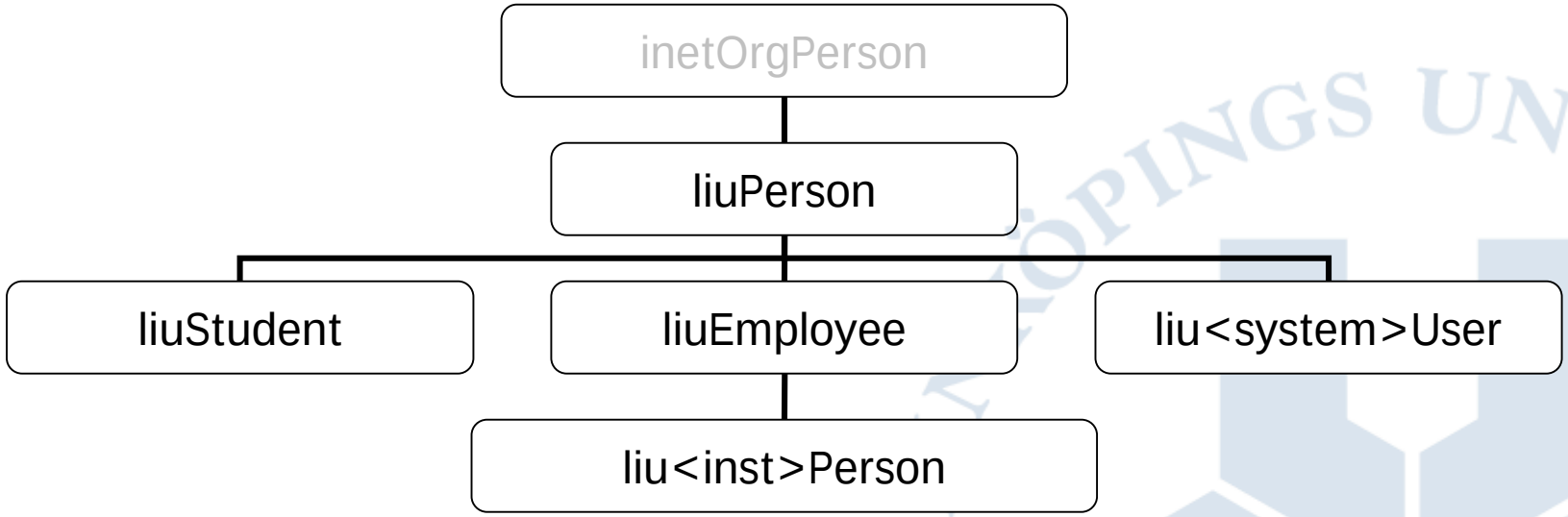
LUKAS
LDAP-struktur

- Övergått till DC-träd
  - dc=<inst>, dc=liu, dc=se
- Personal som finns i PA läggs in via ”primärbefattning” direkt (via Perl)
- Annan personal inlagd via BESLUT
- Användare tillhör institution även helt utan extra-attribut
  - Enkelt att nollställa konto vid t ex avslut

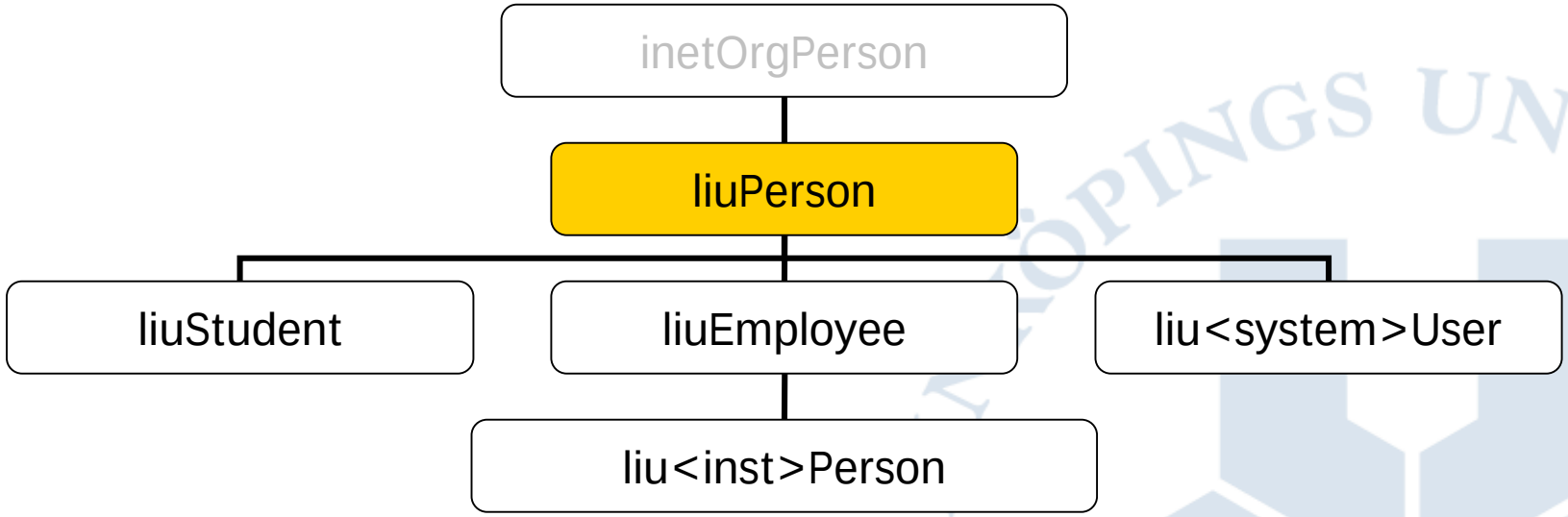
LUKAS
Databas

- En databas per institution
- Många replikeringsavtal blir det...
- Replikering av enskild institution möjlig
  - Replikering åt ”andra hållet” teoretiskt möjlig

LUKAS
Objektklasser



LUKAS
Objektklasser

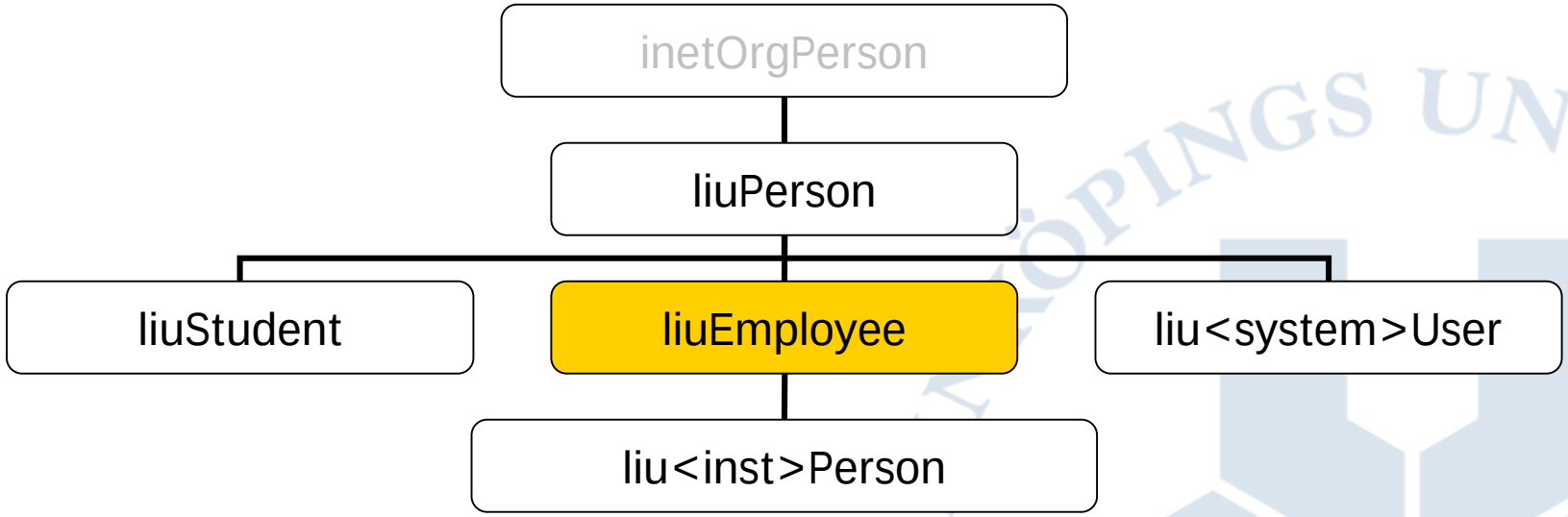


LUKAS

Objektklasser – liuPerson

- Ärver från inetOrgPerson
- Innehåller gemensamma attribut för både anställda och studenter
  - Personnummer
  - liuBiblCardNumber

LUKAS
Objektklasser

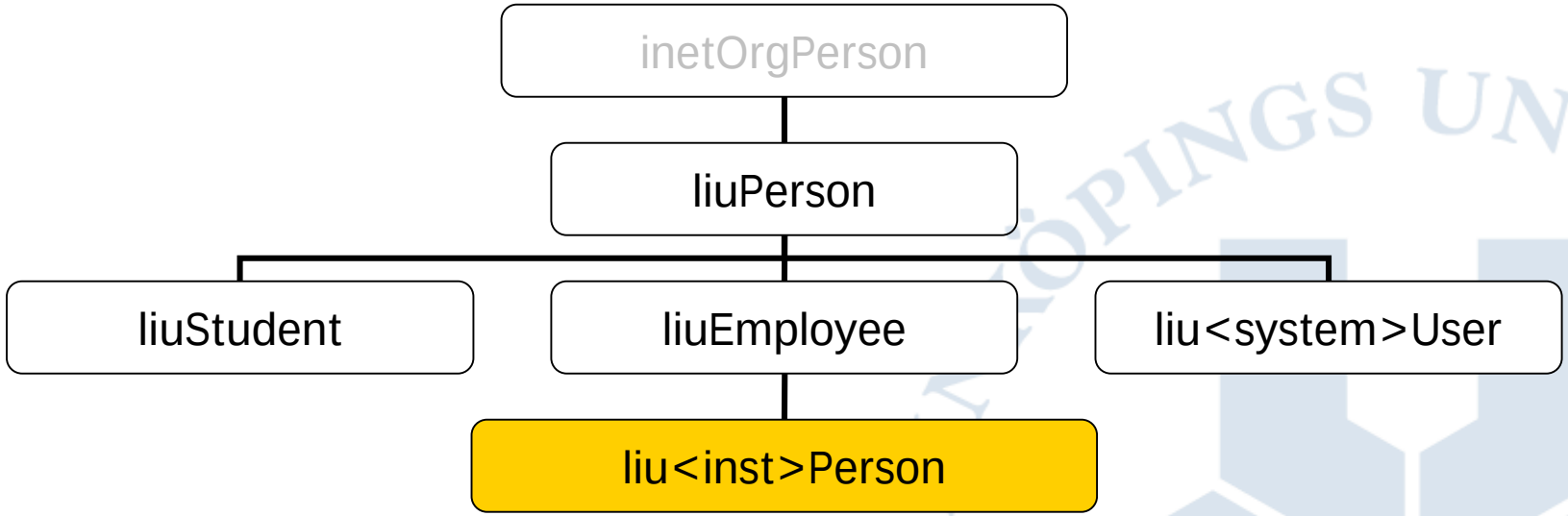


LUKAS

Objektklasser – liuEmployee

- Ärver från liuPerson
- Innehåller gemensamma attribut för anställda på alla institutioner/enheter

LUKAS
Objektklasser

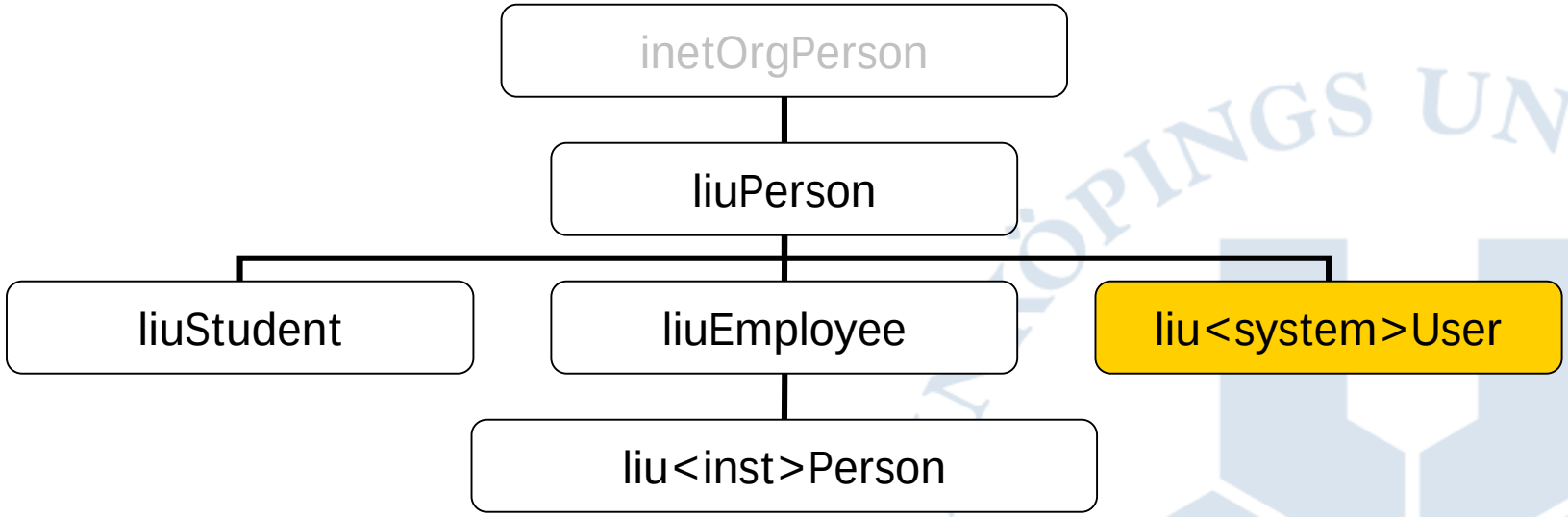


LUKAS

Objektklasser – liu<inst>Person

- Ärver från liuEmployee
- Innehåller gemensamma attribut för anställda på en viss institution
  - Interna arbetsgrupper
  - Andra institutionsspecifika data

LUKAS
Objektklasser

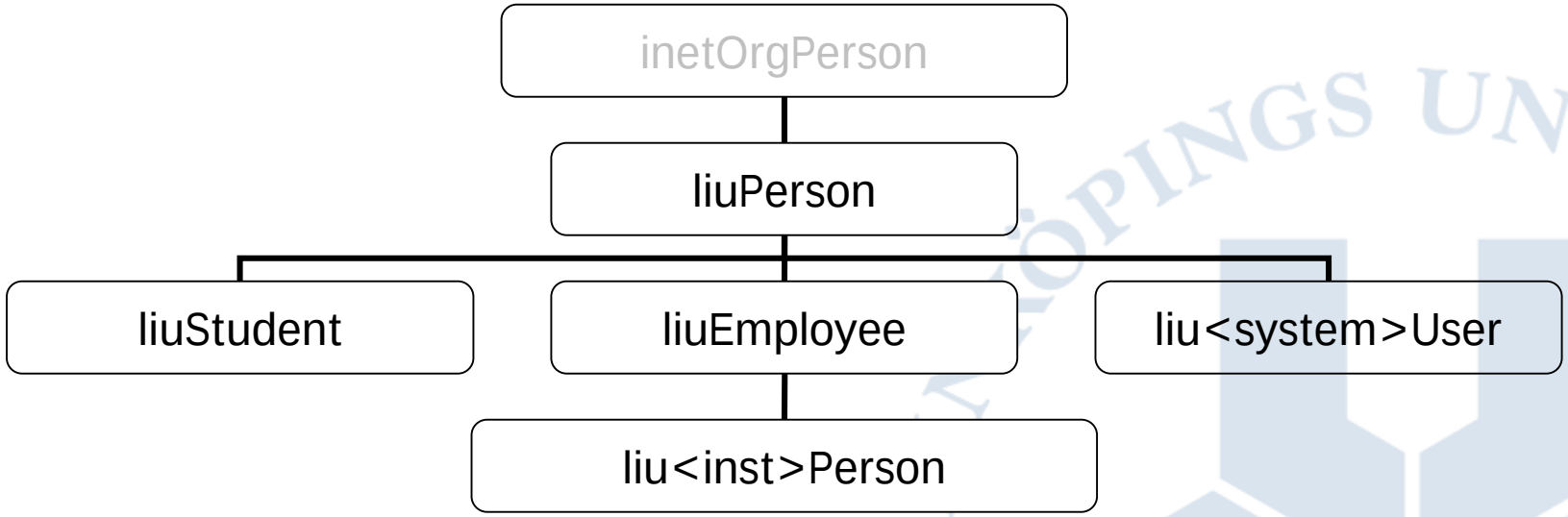


LUKAS

Objektklasser – liu<system>User

- Ärver från liuPerson
- Innehåller gemensamma LiU-specifika attribut för anställda eller studenter som använder ett visst system
  - liuMailOldAddress
- Används bara för komplexa system som kräver flera olika attribut

LUKAS
Objektklasser



LUKAS
Objektklasser – diskussion

- Nackdelar
  - Många objektklasser
  - Strukturen ändras i takt med organisationen
  - Mer jobb att hålla strukturen uppdaterad

LUKAS
Objektklasser – diskussion

- Fördelar
  - Varje enhet/system har ett eget namnutrymme
  - Det är uppenbart till vilken institution/enhet/system ett visst attribut hör
  - Rättigheter kan sättas snävare och mer exakt

LUKAS
Rättigheter

- Roller på alla nivåer
  - liuRole
  - liuEmployeeRole
  - liuUnitRole
  - liuMailRole
- ”Master-switch”
  - Ett huvud-attribut har veto över rollerna

LUKAS
ACIer

- Anonymitet
- Valbar begränsning av läsbart data
  - Centrala ACIer, begränsning på individnivå
  - Underlättar undantag för systemkonton
  - Begränsning i olika nivåer

LUKAS
Säkerhet

- Enbart personliga konton för slutanvändare
  - Gemensamma uppgifter löses i applikationen
- Systemkonton för applikationer som behöver det
  - Så få rättigheter som möjligt
  - Ej möjligt att logga in med dessa i andra system

LUKAS
Framtid

- Sätt rättigheter via central tjänst
- OS-integration på vissa system
- Web-applikationer
  - LDAP-auth via Apache? Via LiU-pass?
- Replikering
  - Chaining
  - Replikor

LUKAS

Katalogtjänst@LiU

Mattias Carlsson

Systemadministratör

mattias@unit.liu.se

013-28 1753

LiU-pass

Anv.identifiering vid Linköpings Universitet

Jim Nordlander

Projektledare/Systemutvecklare

jim@unit.liu.se

013-28 1752

Översikt

Mål och syften

Vad är LiU-pass

Framtid

Frågor

Vad ska LiU-pass lösa

Användaren ska kunna:

- Använda ett användarid och lösenord
- Byta lösenord centralt

För administratören

- Central administration

  skapa/stänga av konto

  byta lösenord.

Vad ska LiU-pass lösa 2

För säkerhetsansvarig

- Ta bort lösenorden för applikationerna
- Inga Admin-lösenord behövs
- En svag punkt att bevaka

Vad är LiU-pass

En liten SOAP-tjänst för användaridentifiering

- Inloggning (anv.id + lösenord), ger biljett (ticket, pass, token etc)

Vad är LiU-pass

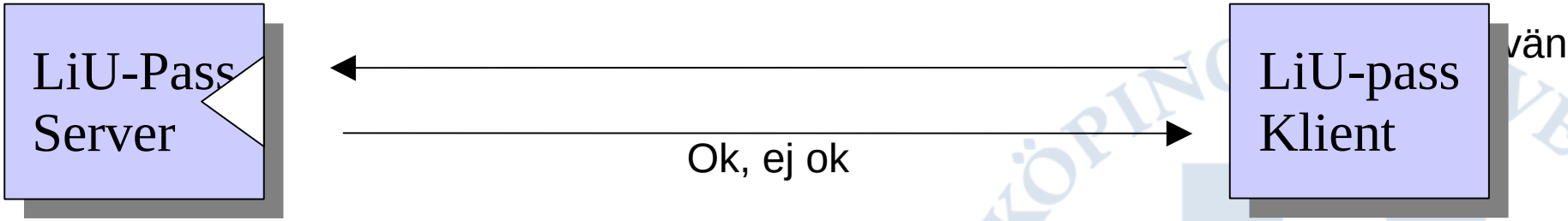
Inloggning

Vad är LiU-pass

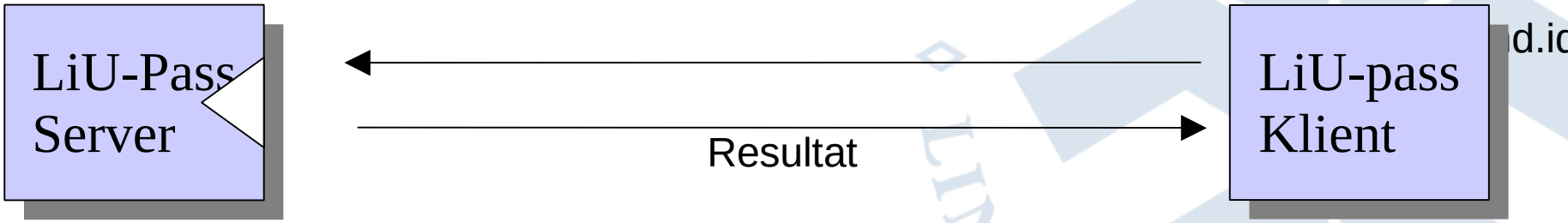
- Verifiering av biljett.
- Möjlighet att hämta attribut från LUKAS (LDAP)

Vad är LiU-pass

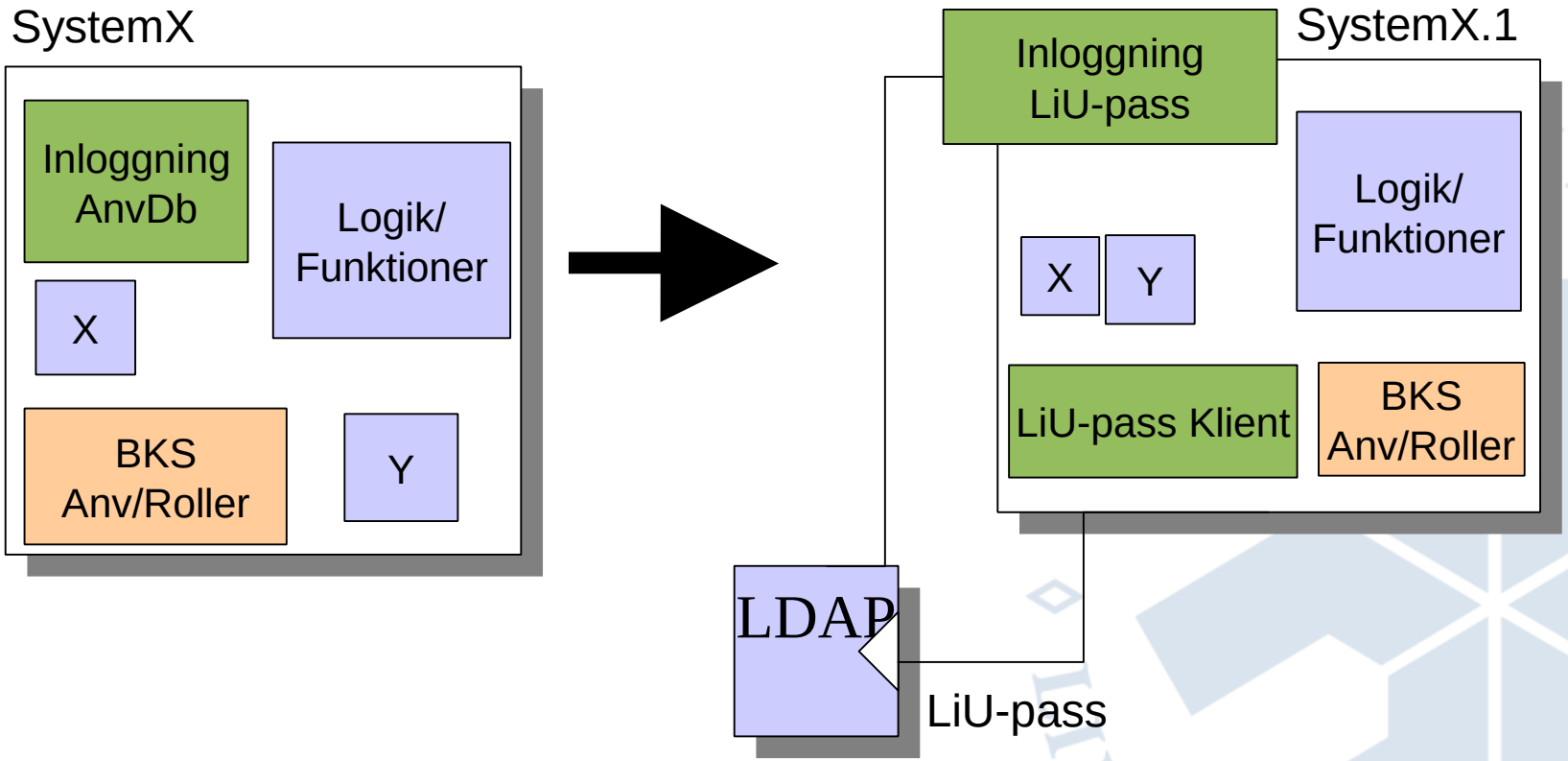
Verifiering av LiU-pass



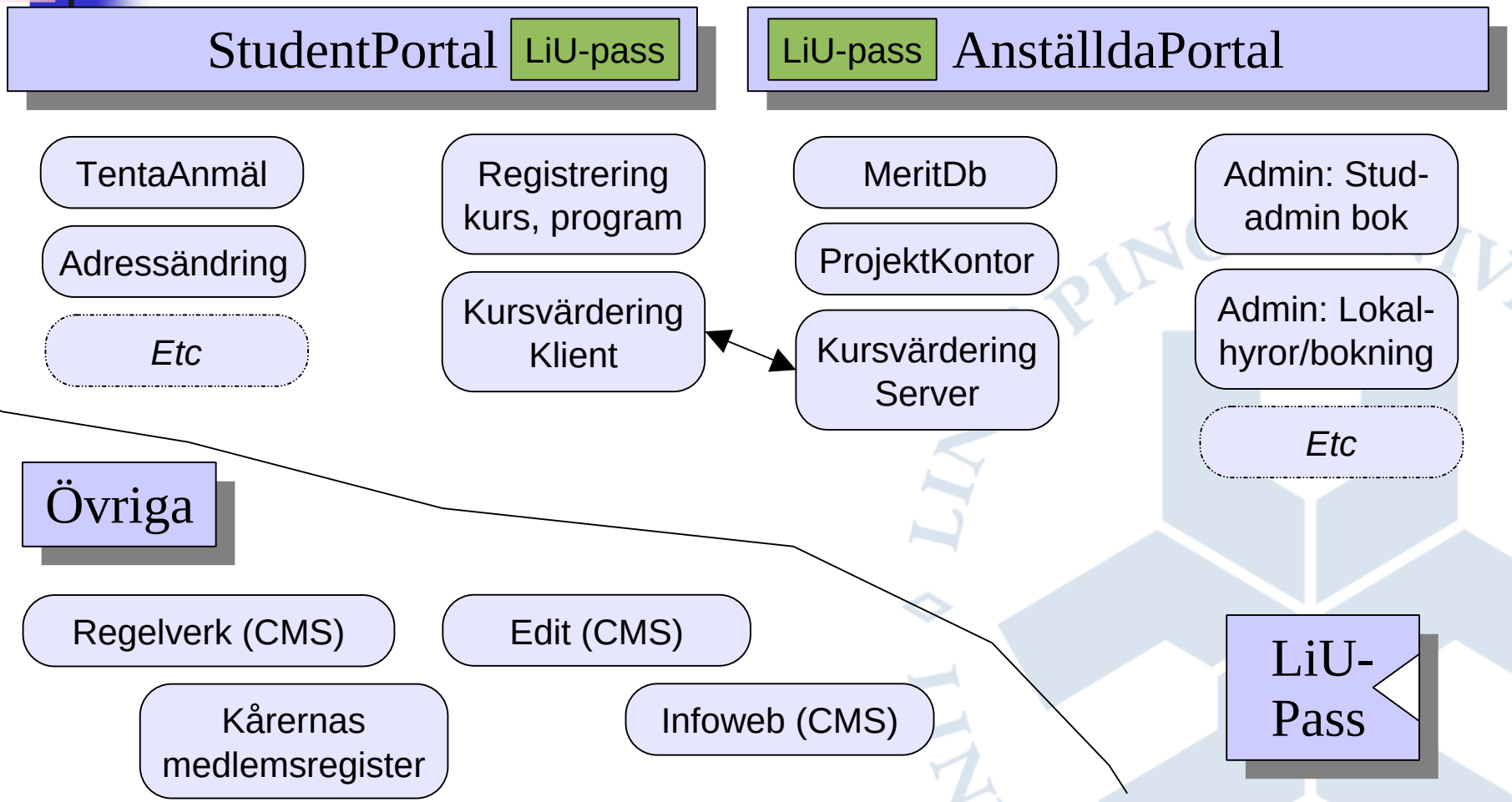
Attribut (t ex grupper)



Mål 1 - Använda LiU-pass



Portaler och tjänster

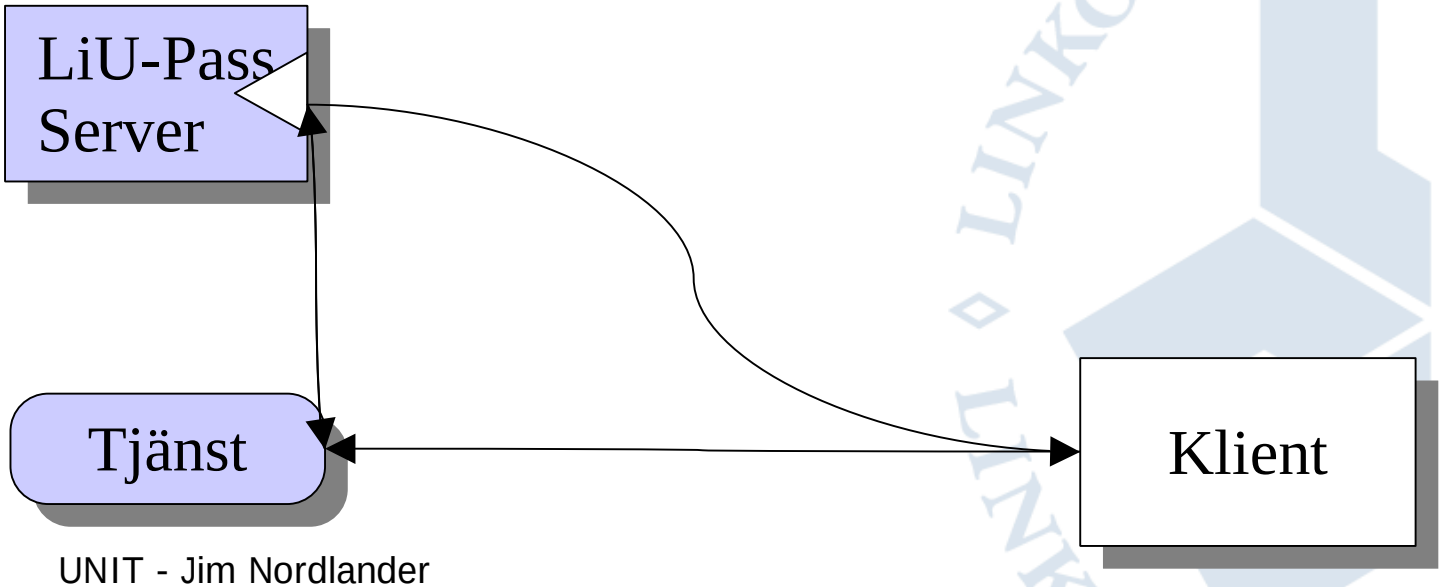


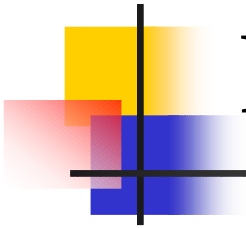
Framtid

Fortsätta konvertera gamla applikationer

Singel-sign-on (login.liu.se)

Central behörighetssystem (BKS)





Frågor

Säkerhet

Teknik

Architektur (helst ej LDAP :)

LiU-pass

Anv.identifiering vid Linköpings Universitet

Jim Nordlander

Projektledare/Systemutvecklare

jim@unit.liu.se

013-28 1752

Sun ONE Identity Management

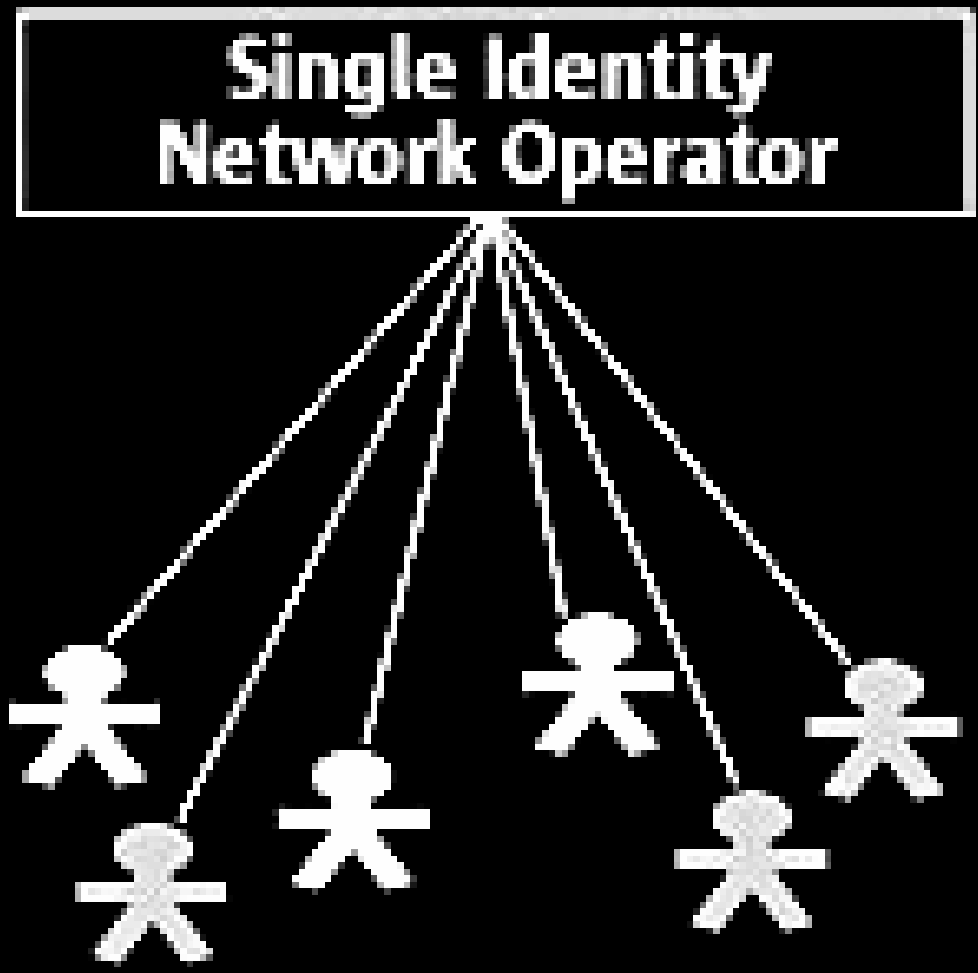
peter@inserve.se

“Identity is the most basic element in a high value relationship”

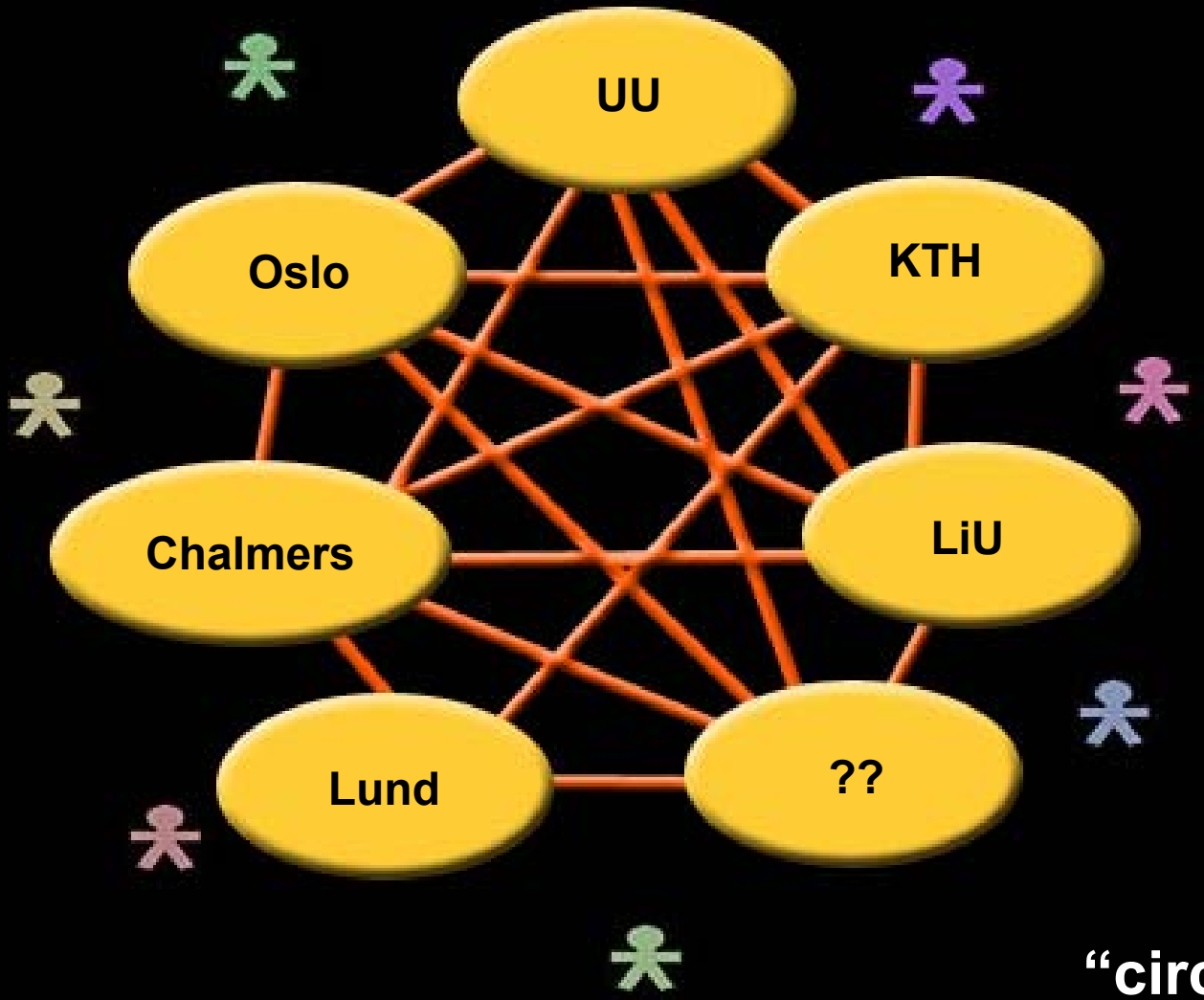
Intro

- Project Athena 1989
  - Hesiod & Kerberos
  - Teknik < allt annat
- 80/20
  - Det enkla eller svåra först?
  - Bygga själv eller köpa?
- Netscape - iPlanet - Sun ONE ☺
- Integrerbar vs Integrerad
- Arkitekturperspektiv

Centraliserad modell



Federerad modell



“circle of trust”

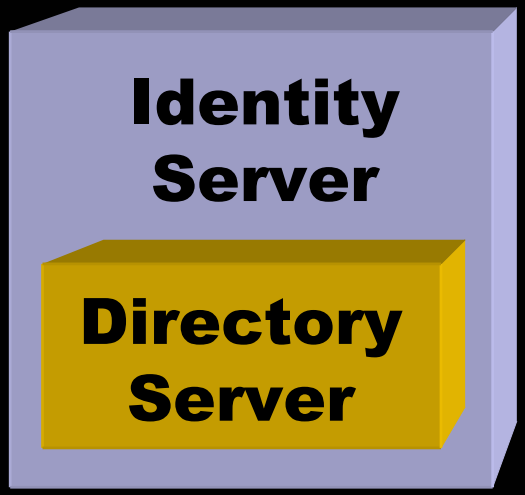
- En allians för att etablera en **öppen standard** för federerad nätverksidentitet.
- Skapa förutsättningar för ett stort antal **plattformsneutrala** identitets-baserade produkter och tjänster. Man levererar ett antal **specifikationer**.
- Göra det möjligt för företag och organisationer att realisera **nya tjänster**, **kostnadsbesparingar** och hantera sin data på sina egna villkor, inte någon annans.
- http://www.projectliberty.org

Directory Server

- LDAP Directory
- XML, DSMLv2, LDAPv3
- Massive scale (>50M) performance, replication
- Multi-platform support

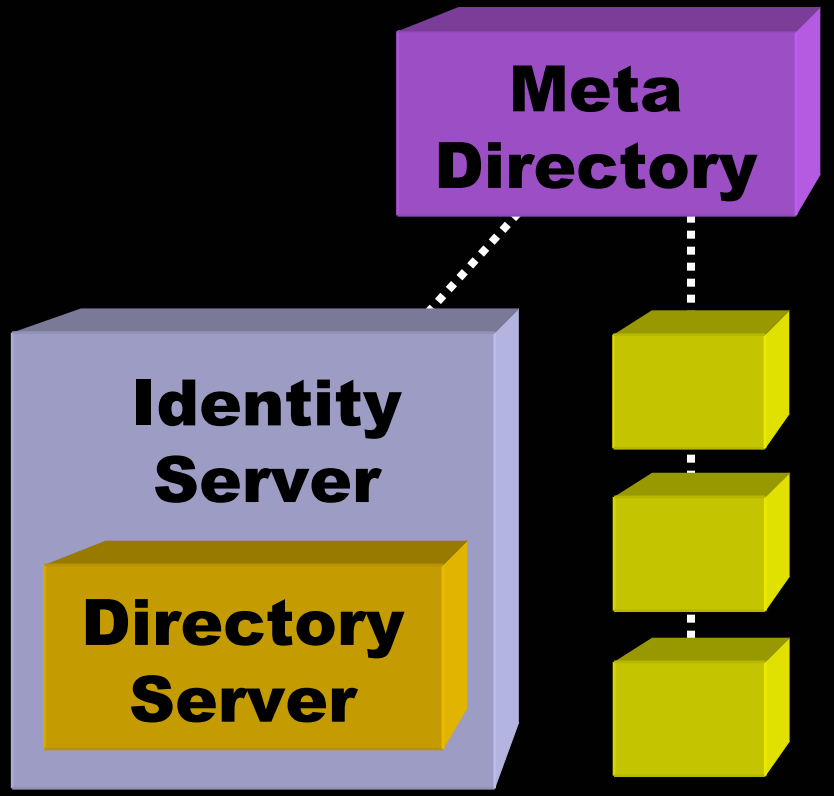
- LDAP SDKs in C and Java
- Tools

*) Security Assertion Markup Language (SAML)



Identity Server

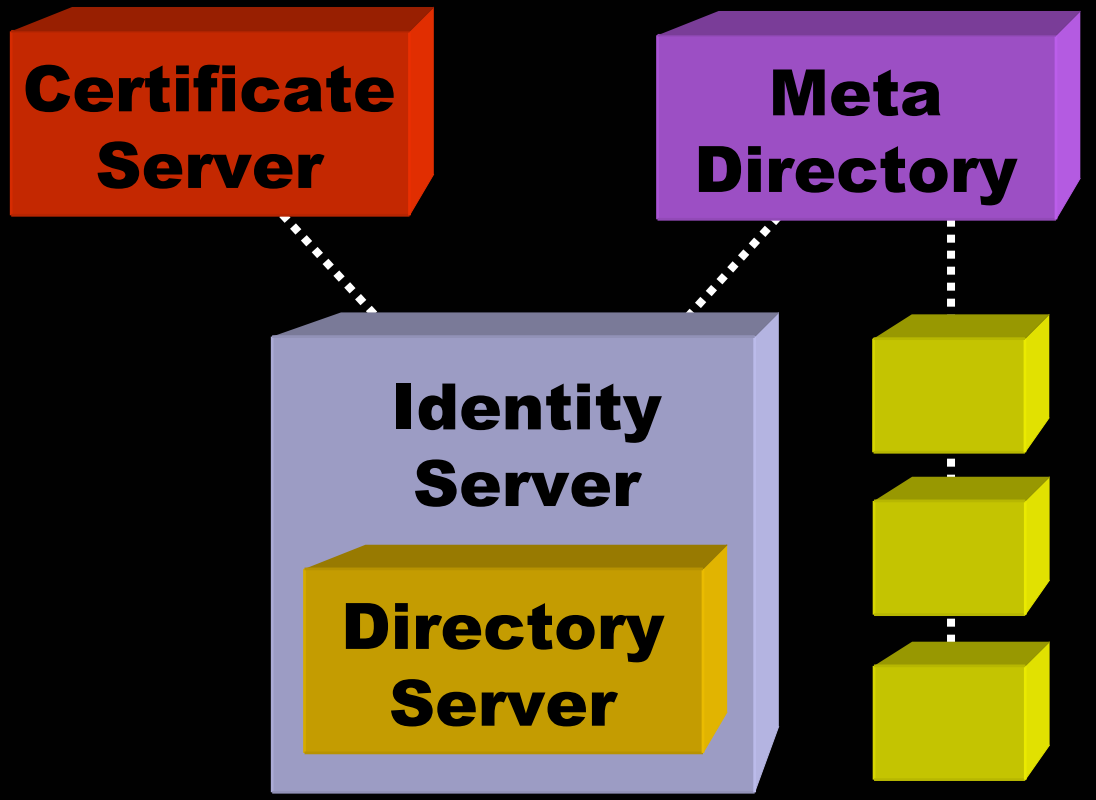
- Role Based Access
- Authentication, Authorization, Logging
- Federated Identity, SAML*
- JAAS, XML DSIG, SOAP
- Web SSO + API
- Policy-based management
- Delegated administration of identities, policies, groups
- Self-registration / self-management
- URL Agents



Meta Directory

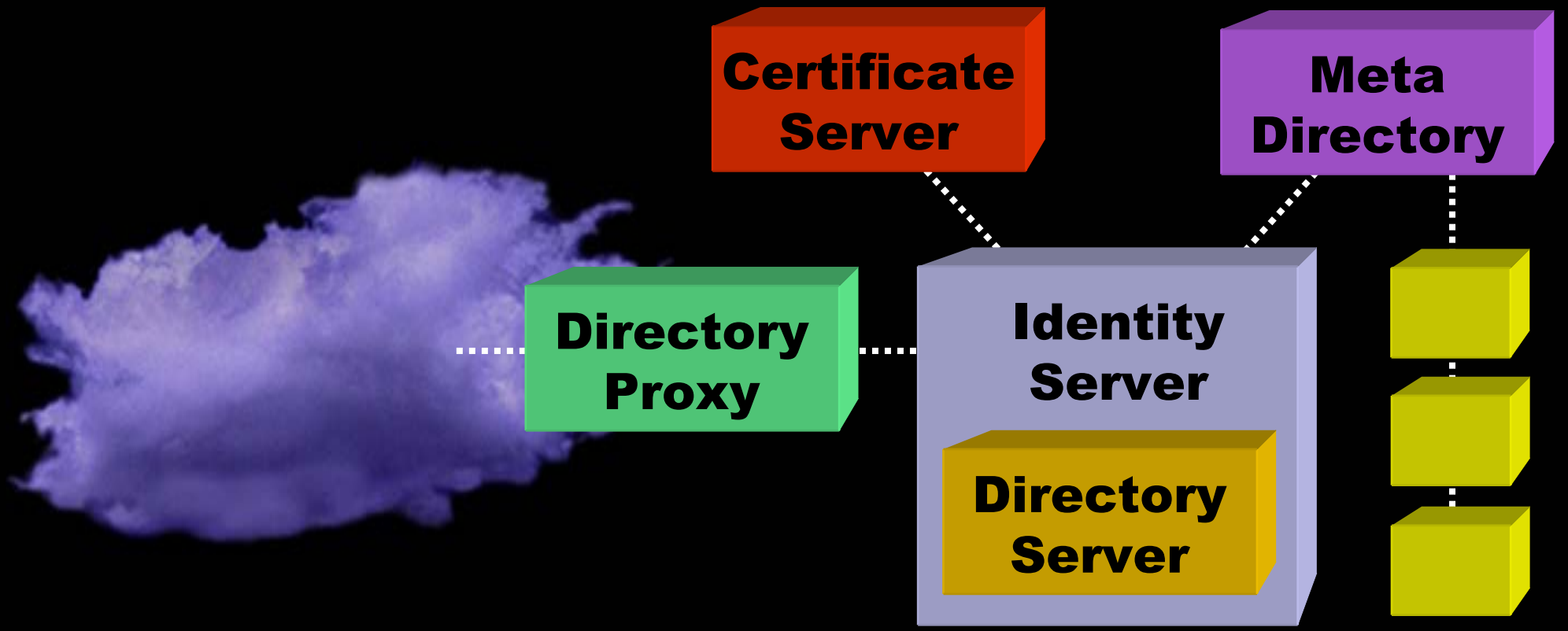
- Publish, Synchronize, Join across databases, apps and directories
- Provisioning + business rules
- Connectors for AD, Oracle, NT Domains

- Exchange, Notes, Novell,
- Java API (JCA)
- XML data bus



Certificate Server

- Create/Publish/Revoke X.509v3 Certificates
- PKCS standards compliance
- Registration/Certification Authority
- OCSP
- FIPS compliance



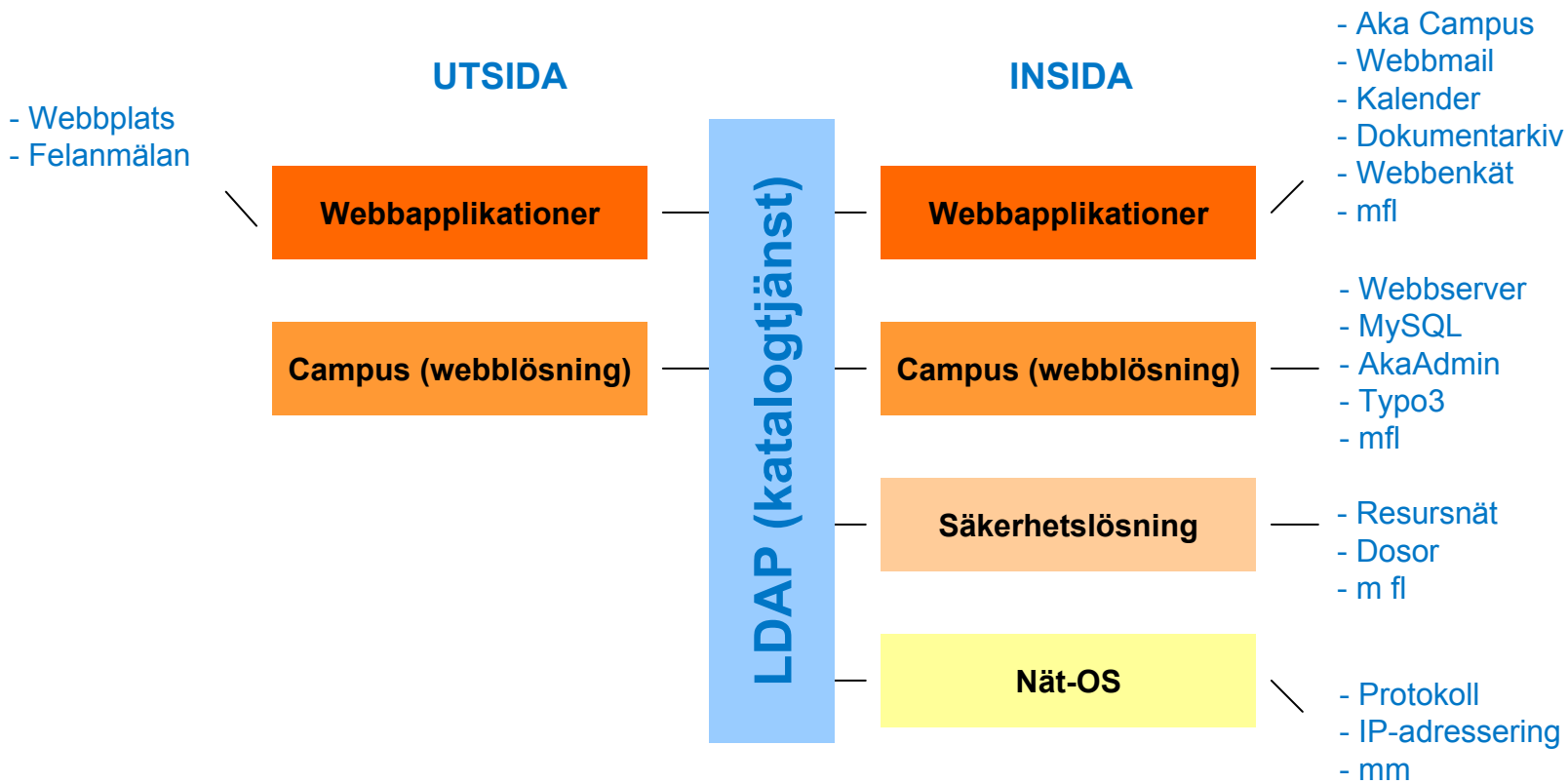
Directory Proxy

- Firewall-like security
- Prevents DOS attacks
- "always on" directory services
- fail-over, fail-back
- Client interoperability
- Query filtering and routing.

Om man började om...

1. **Assessment**
   Strategi; vad har vi, vart vill vi, hur viktigt är det, hur kan vi räkna hem det?
2. **Architect**
   Skilj på identitet, affärslogik och hur saker och ting kopplas samman. Standardkomponenter + integration.
3. **Proof-of-concept**
   Verifiera att det funkar. Ställ krav på leverantör.
4. **Implementera**
   80/20 regeln. Var hård internt/externt.

Arkitektur



AKADEMISKA HUS

www.sun.com/sunone

peter@inserve.se
inge.persson@sun.com

User administration inside Finnish Higher Education Institutes

results from the KATO project

Barbro Sjöblom

EDS 2003

Uppsala 30.1.2003

FEIDHE

- FEIDHE (Electronic Identification in Finnish Higher Education)
- Goal: Investigate possibilities for implementing a smart card based electronic identification system
- Started in June 2000 and completed in March 2002
- Cooperation between the Computer Centers at Finnish Higher Education Institutes (HEI), national student unions and Center for high-performance computing and networking (CSC)
- Documentation: https://hstya.funet.fi/
- New projects: HAKA and KATO

HAKA – Directories in User Administration

- ”Hakemistot käyttäjähallinnossa”
- 1.5.2002 – 1.5.2003
- Goal I: To give recommendations for a common interface for the User Administration that supports Electronic Identification and information exchange, both between HEI and between a HEI and a common service.

HAKA – Directories in User Administration

- Actions:
  - document the existing needs (HEI, service providers)
  - document existing schemas
  - define a common schema
  - security and data protection issues
  - testing different architectures (LDAP, Shibboleth, PAPI)
  - minimi requirements for the user administration at a individual HEI
  - provide information about middleware and user administration to the HEI

HAKA – Directories in User Administration

- Goal II: Support further development of the User Administration in the HEI
- Actions:
  - document best practices of user administration
  - encourage HEI to use strong authentication
  - will use the results from the KATO project

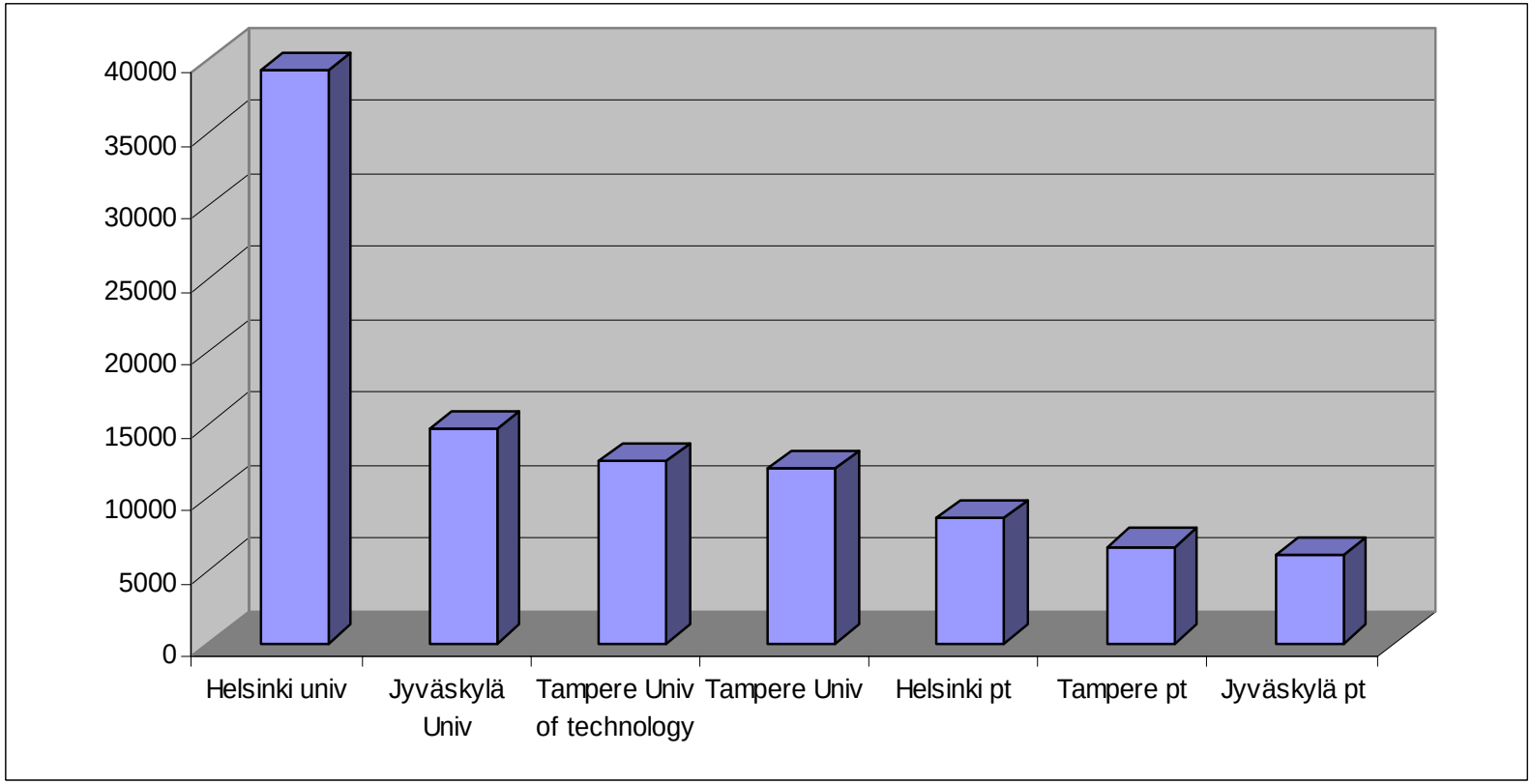
KATO project

- April 2002 – October 2002
- Goal: to document best practices of user administration in HEI
- Scope: inside the HEIs
  - not across HEI boundaries
- Computing center staff interviewed in 7 HEIs
- Document available since 10/2002
  - in Finnish (http://www.csc.fi/proj/kato/)
- Some outcomings described here

The operational environment: The users

- students
  - undergraduate, postgraduate, supplementary, visiting students...
- employees
  - universities: a lot of employees and new contracts
  - polytechnics: not so many employees
- some users both student and employee
  - very common in universities
- some legitimate users neither student nor employee
  - e.g. researchers of the Academy of Finland

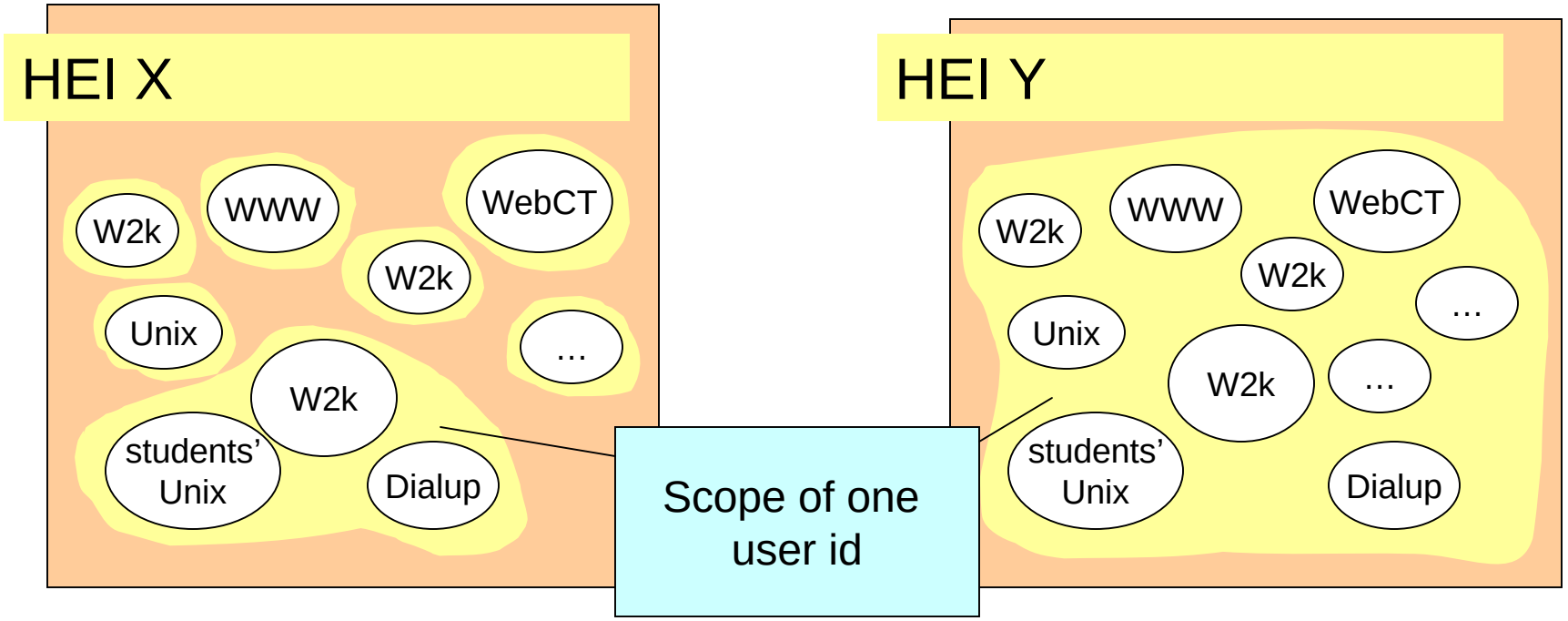
The operational environment: Number of users



The operational environment: How the IT systems are maintained

- centralized: all the workstations, servers and services maintained by the computer center
  - typical in polytechnics and smaller universities
- distributed: maintainance is done by the institutes
  - typical in some larger universities
  - workstations used by faculty: maintained by the institutes
  - workstations used by students at TUT and University of Tampere: 50% maintained centrally, 50% by institutes

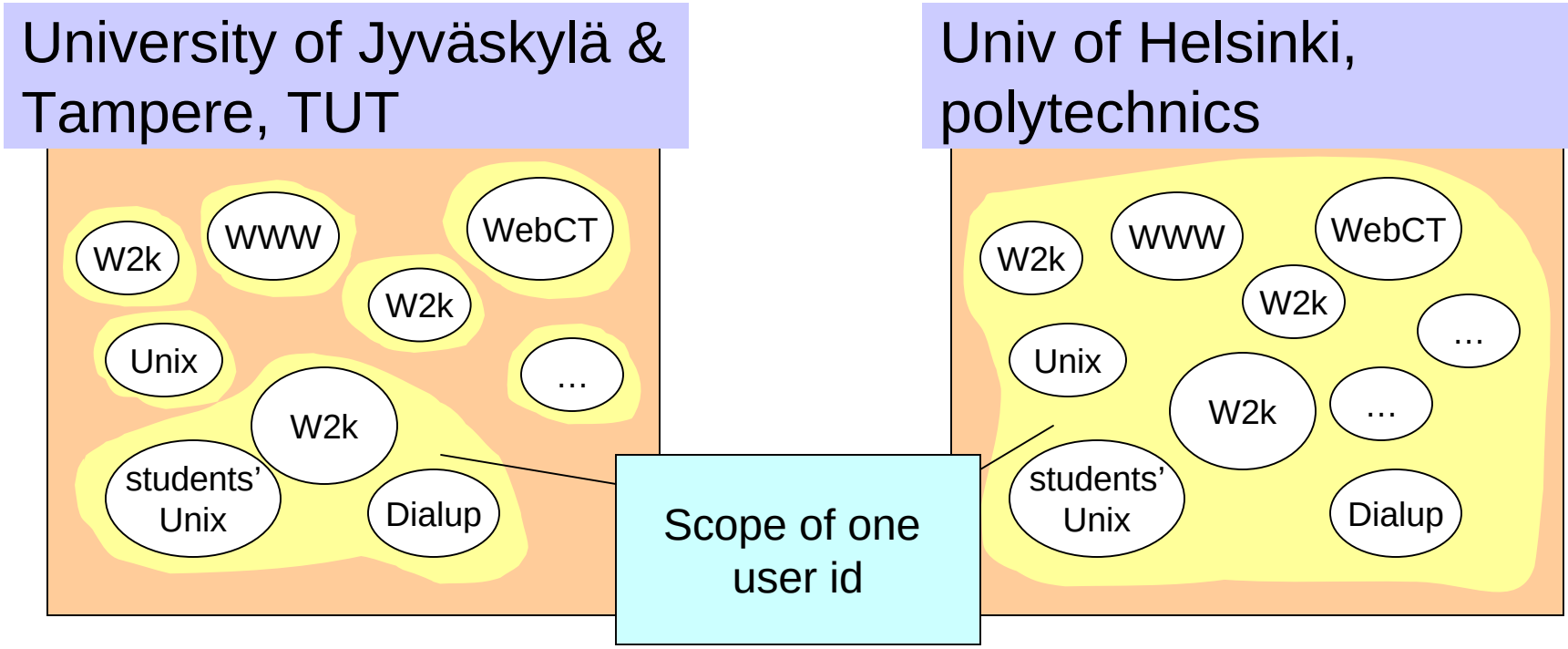
A fundamental issue: The scope of user identity



System specific identity (mostly)

Organization wide identity

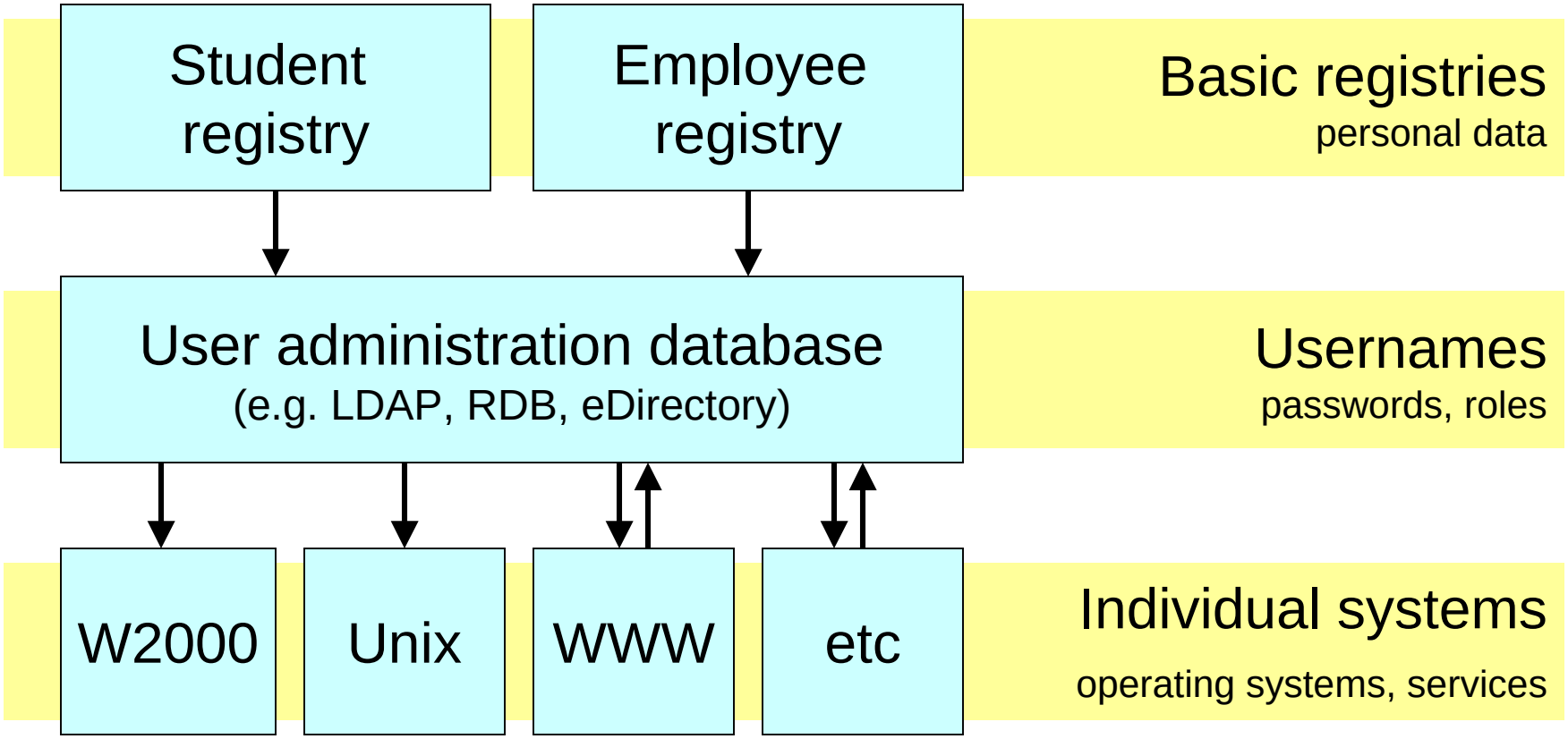
A fundamental issue: The scope of user identity



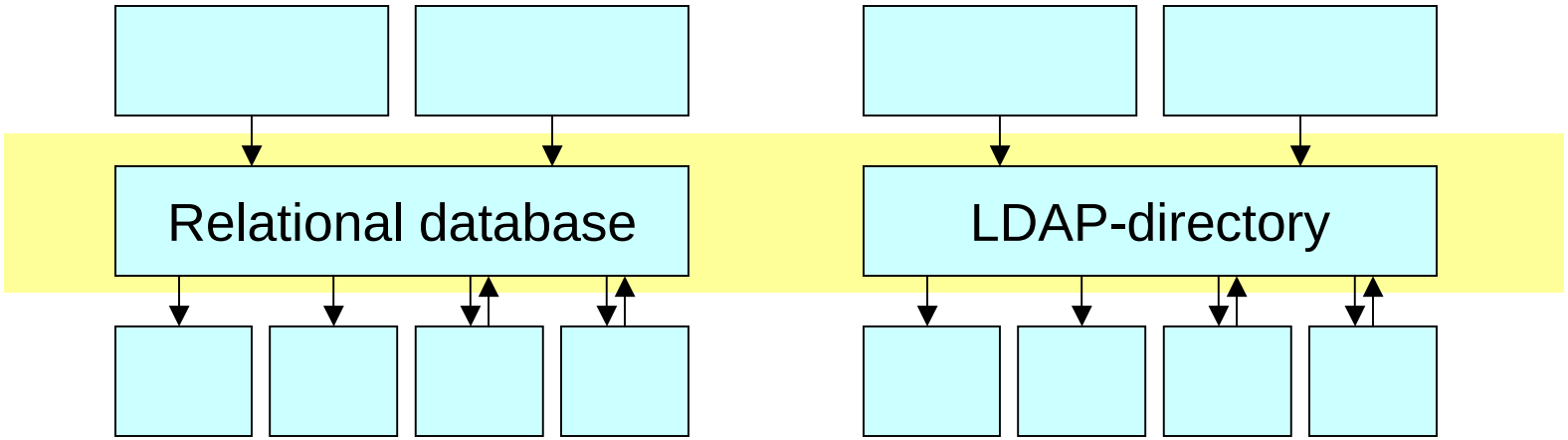
System specific identity (mostly)

Organization wide identity

The fundamental architecture



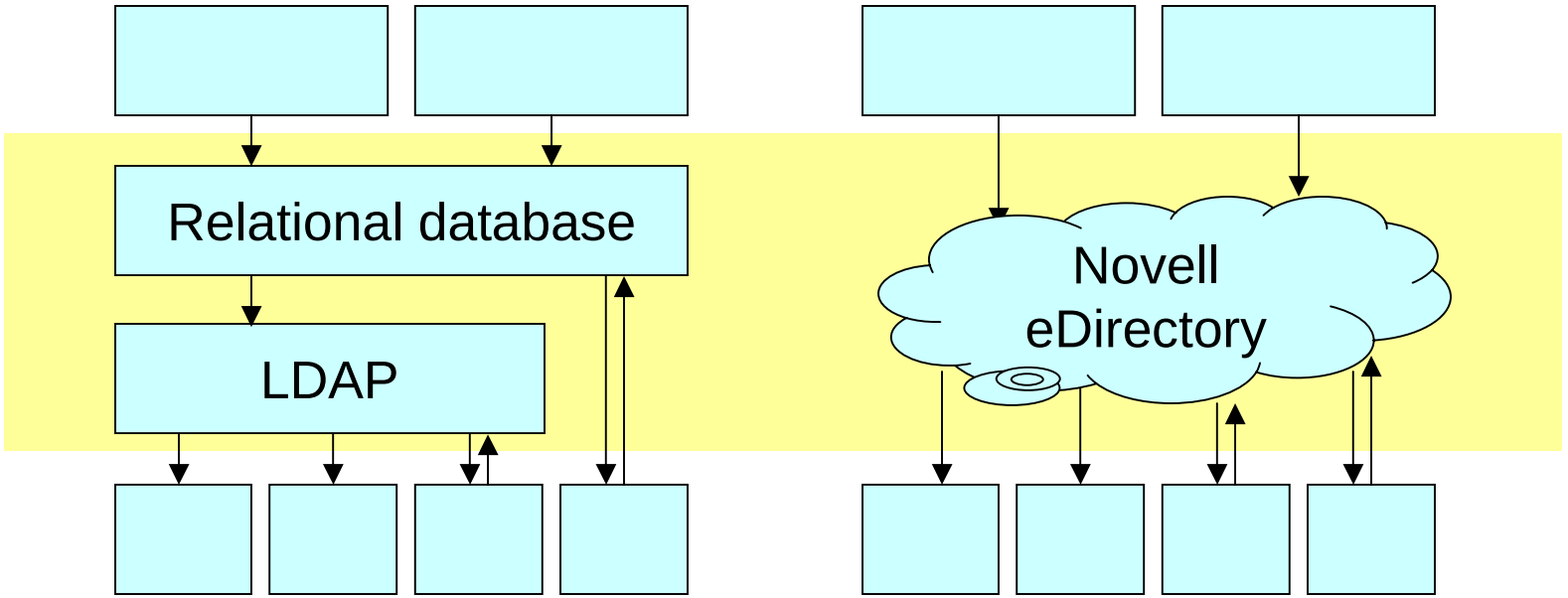
Different ways to implement (1/2)



University of Helsinki, Tampere

Tampere polytechnic, Åbo Akademi

Different ways to implement (2/2)



TUT, University of Jyväskylä

Polytechnic of Helsinki, Jyväskylä

A fundamental issue: roles

- If a user is both a student and an employee, does he have two usernames?
  - new services on the web with role-based access control
- If a user is neither a student nor an employee, how is his account administrated
  - there is no basic registry for them
  - in TUT they are called UFOs
    - nobody knows who they really are and where they are coming from...

A fundamental issue: unique identifiers

- identifiers represent the identity of a user
  - username, email-address
  - student/employee number
  - social security number (what about foreigners?)
  - other identifiers
- related questions
  - Can identifiers be revoked?
  - Can identifiers be reassigned?

A fundamental issue: authentication

- Passwords
- How may passwords per user?
  - One per user identity
  - Several per user identity (one low-security etc...)
- Single sign-on
  - On the web? Usability, trust relationships
  - On the workstation
- PKI and personal certificates
  - smart cards

Further information:

- The FEIDHE project: https://hstya.funet.fi/

- HAKA: http://www.csc.fi/proj/hakemistot/haka.phtml
- KATO: http://www.csc.fi/proj/kato

- Mikael Lindén (Mikael.Linden@csc.fi). Leader of the HAKA and KATO projects.
- Barbro Sjöblom (babo@abo.fi). User administration at Åbo Akademi.