

Complete open source IAM solution



Radovan Semančík
LDAPcon, November 2015

Radovan Semančík



Current:

Software Architect at **Evolveum**

Architect of Evolveum **midPoint**

Contributor to **ConnId** and **Apache Directory API**

Past:

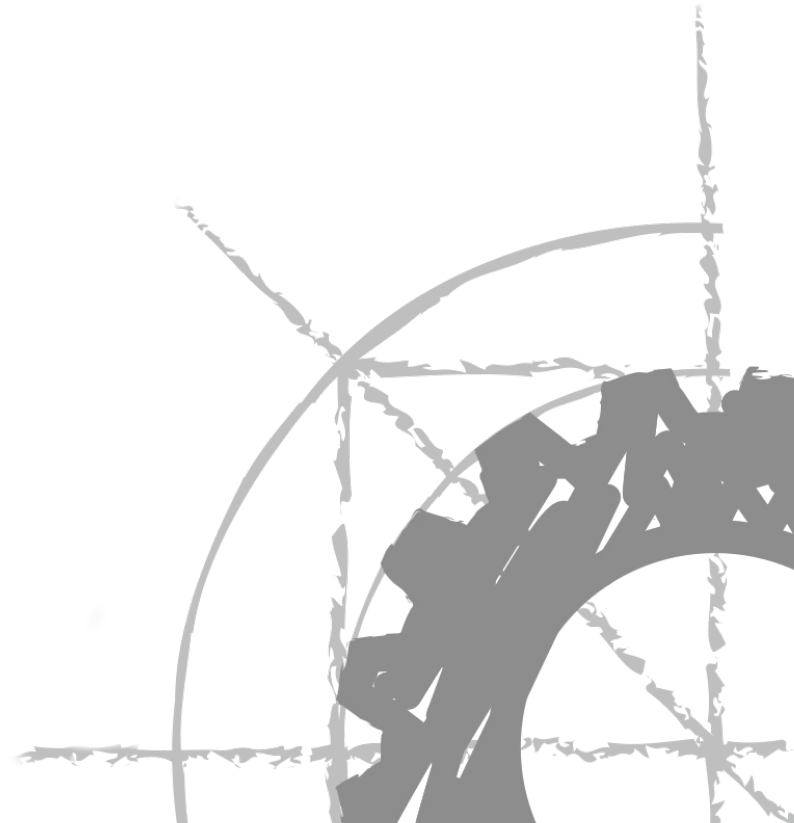
Sun LDAP and IDM deployments (early 2000s)

OpenIDM v1, OpenICF

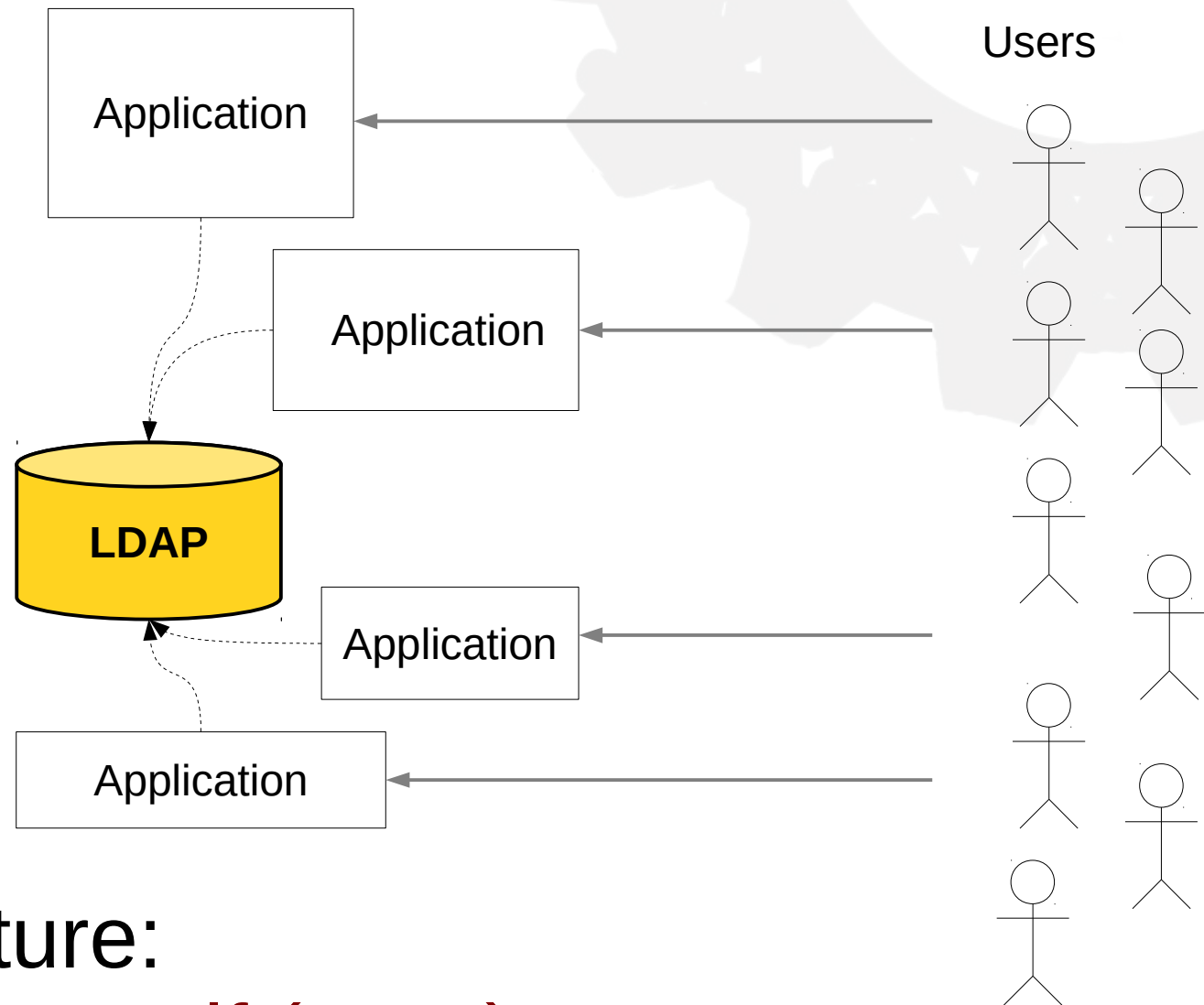
Many software architecture and security projects

**Complete solution? Why?
Is LDAP not enough?**

Evolveum

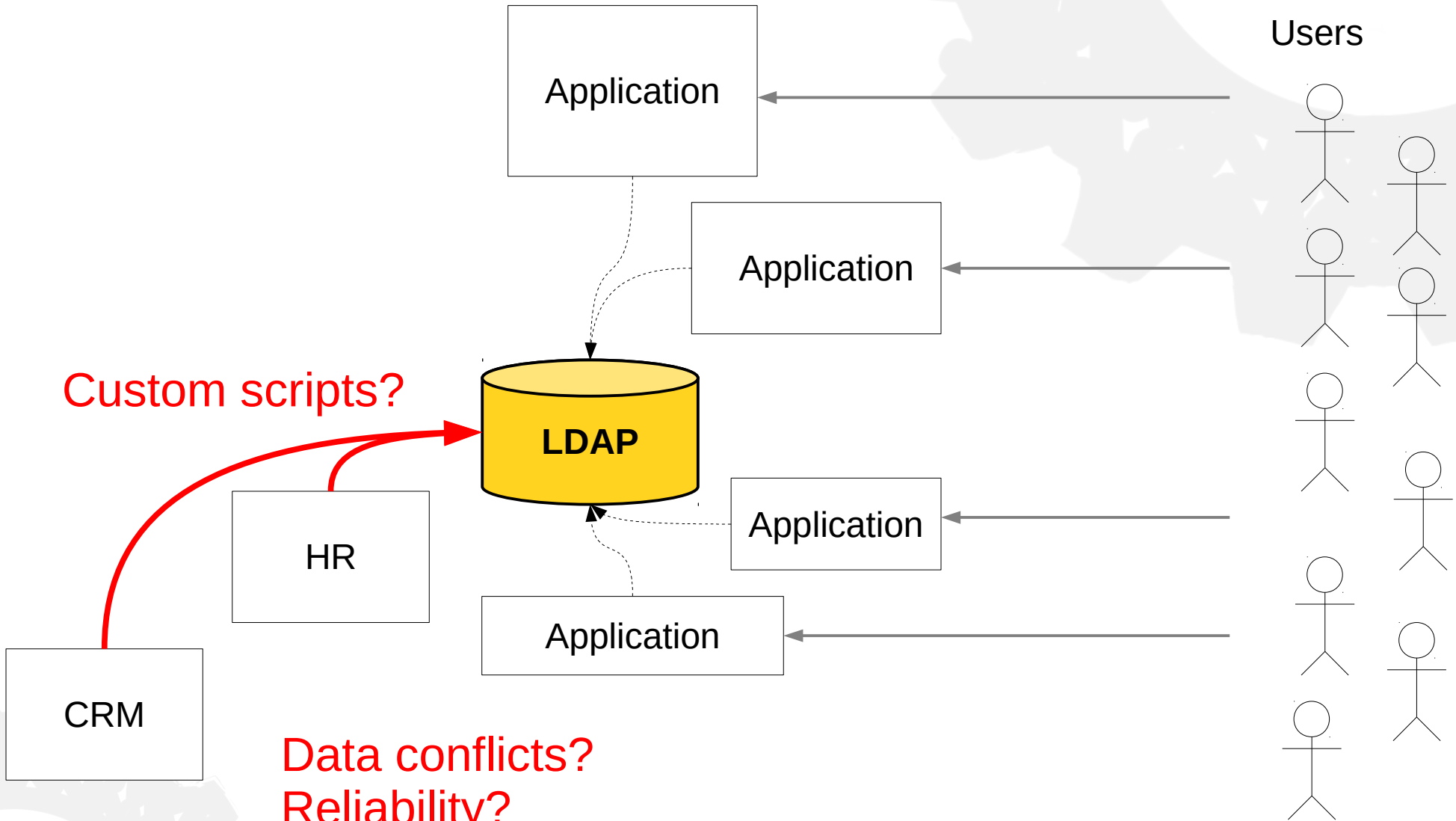
The Evolveum logo features the word "Evolveum" in a bold, sans-serif font. The letter "o" is replaced by a stylized gear icon. The gear has a central hub and several teeth, and it is overlaid on a circular grid pattern that resembles a technical drawing or a coordinate system.

Yes, theoretically ...



Good architecture:
Don't repeat yourself (DRY)

Practice: Data Sources



Custom scripts?

HR

LDAP

Application

Application

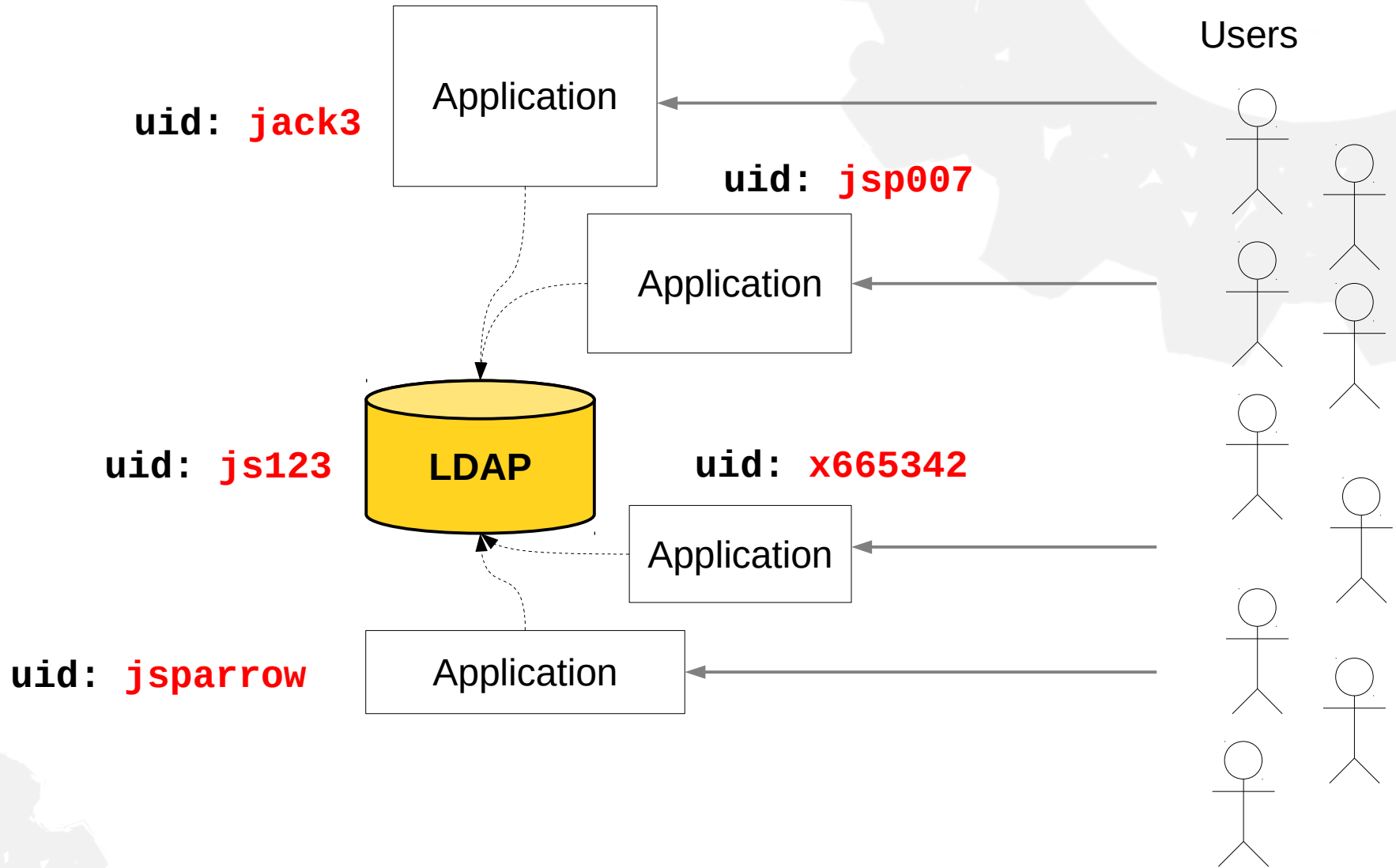
Application

Users

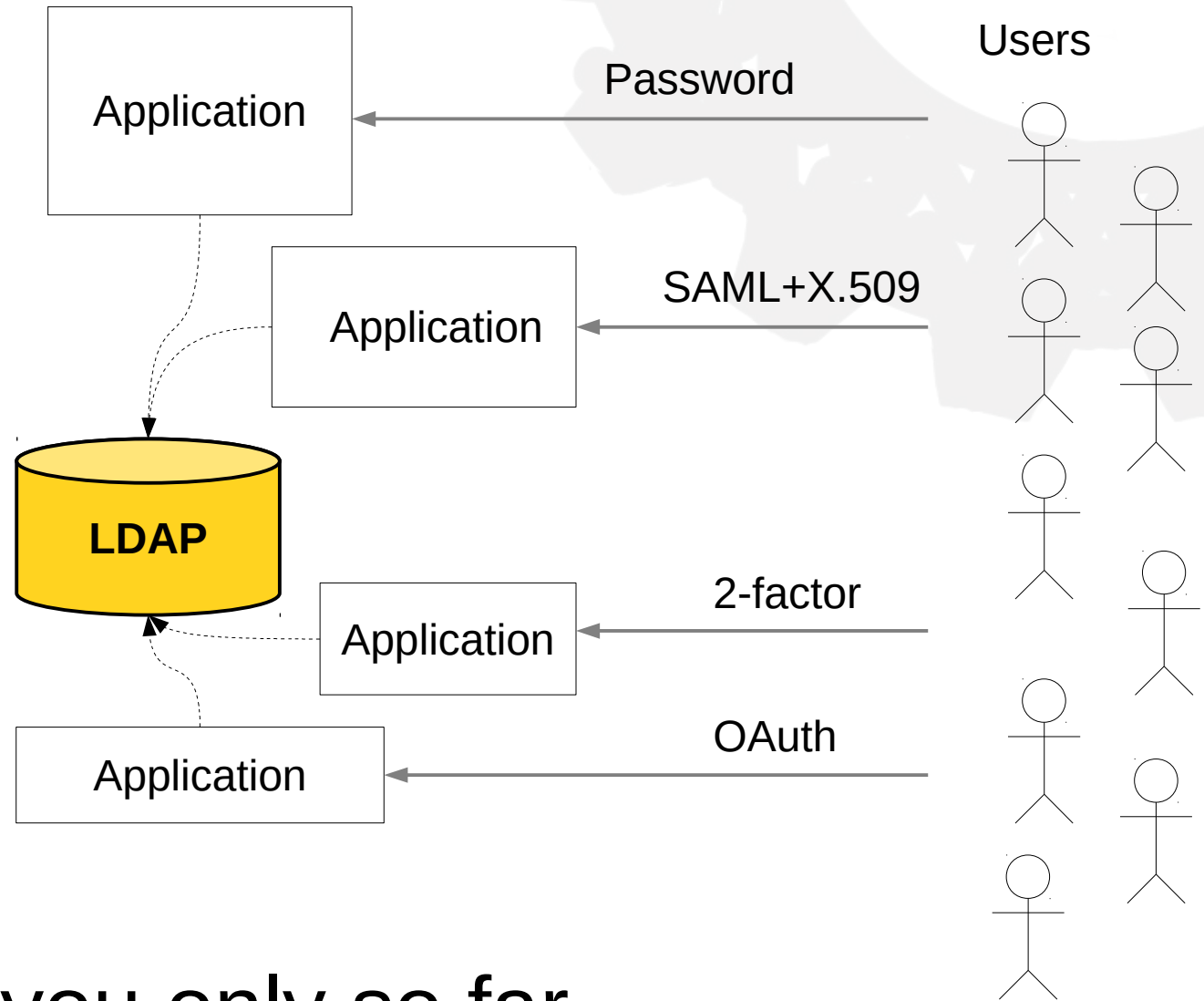
CRM

Data conflicts?
Reliability?
Maintenance?

Practice: Legacy



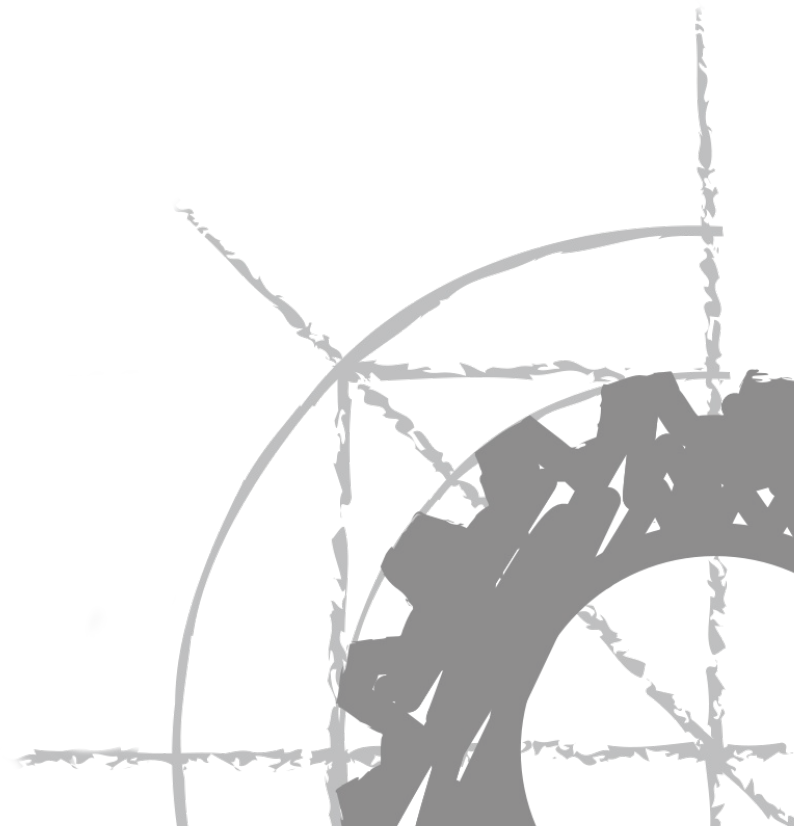
Practice: Authentication



SASL will get you only so far ...

**But ... these are
application problems!
Let's fix the applications and
standardize. We'll be fine.**

Evolveum



Standardization? Really?

```
dn: cn=foo,ou=groups,o=example
objectclass: groupOfNames
member: uid=bar1,ou=people,o=example
member: uid=bar2,ou=people,o=example
```

```
dn: cn=foo,ou=groups,o=example
objectclass: groupOfUniqueNames
uniqueMember: uid=bar1,ou=people,o=example
uniqueMember: uid=bar2,ou=people,o=example
```

RFC2256 (1997)

 **mandatory(!!!)**

Standardization? Really?

```
dn: cn=foo,ou=groups,o=example  
objectclass: groupOfNames  
member: uid=bar1,ou=people,o=example  
member: uid=bar2,ou=people,o=example
```

```
dn: cn=foo,ou=groups,o=example  
objectclass: groupOfUniqueNames  
uniqueMember: uid=bar1,ou=people,o=example  
uniqueMember: uid=bar2,ou=people,o=example
```

RFC2256 (1997)

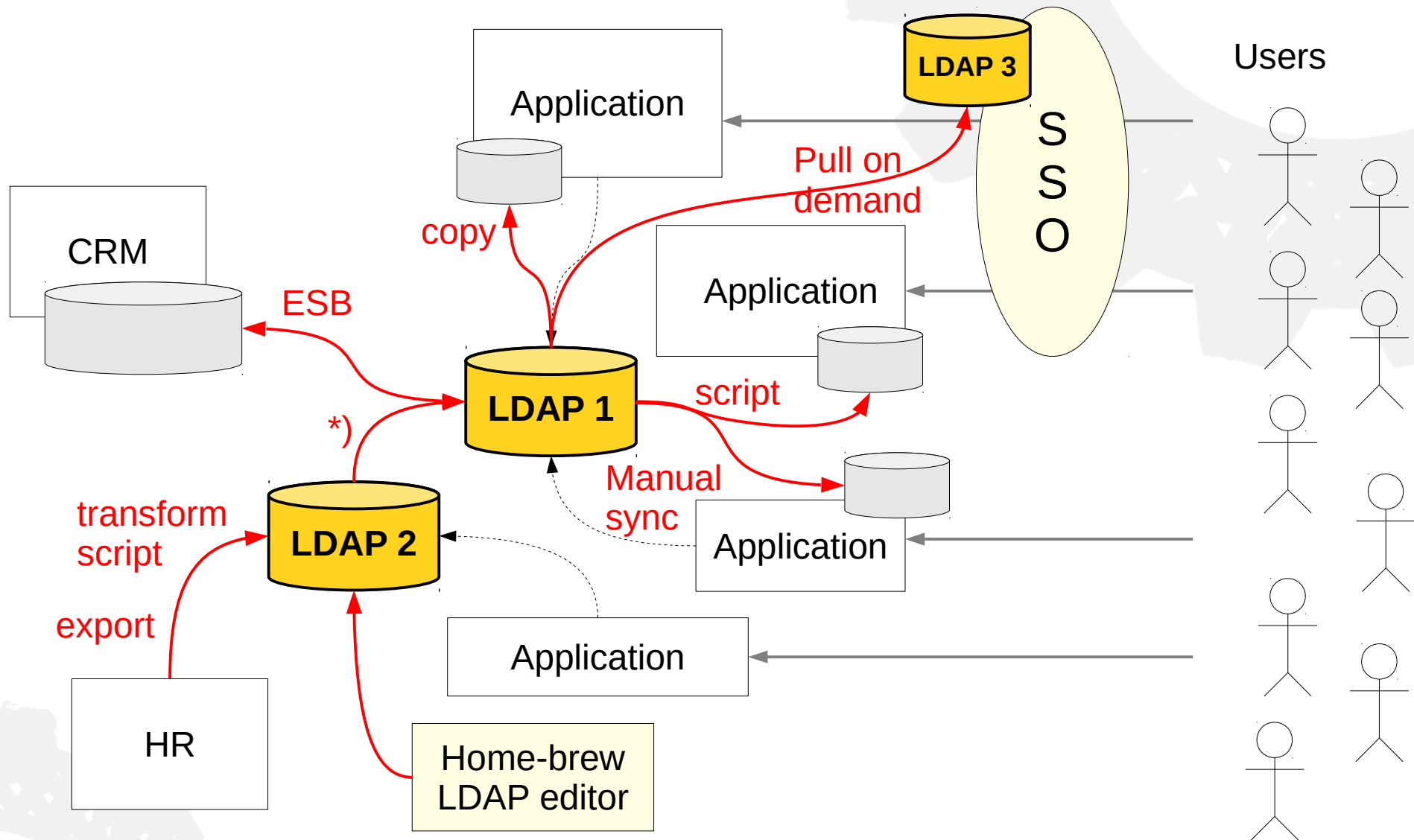
```
dn: cn=foo,ou=groups,o=example  
objectclass: posixGroup  
memberUid: bar1  
memberUid: bar2
```

RFC2307 (1998)

Practice: more problems

- Password reset
- Adaptive authentication
- SSO
- Session management
- ACLs
- Account activation
(enabled/disabled status)
- “memberOf”
- Roles / RBAC
- Password policies
- Access policies (autz)
- Paging (SPR vs VLV)
- Audit
- Reporting
- Data consistency
- Management tools
- User experience
- Schema consistency issues
- Standard violations
- Common sense violations
- Too many data types
- ... most of them unsupported
- DN case sensitivity
- Synchronization

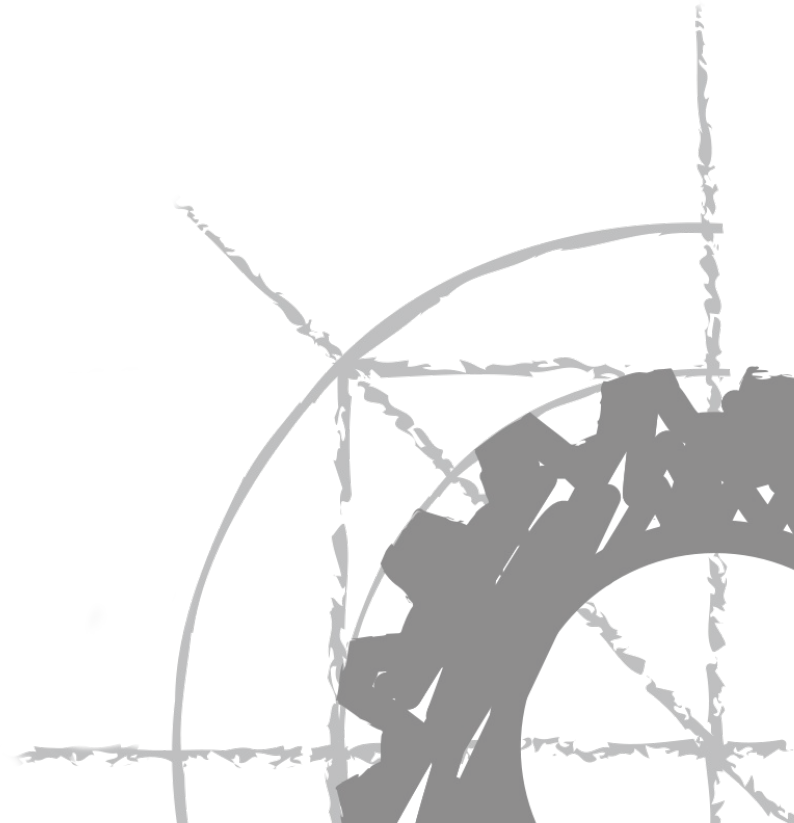
Practice: really messy



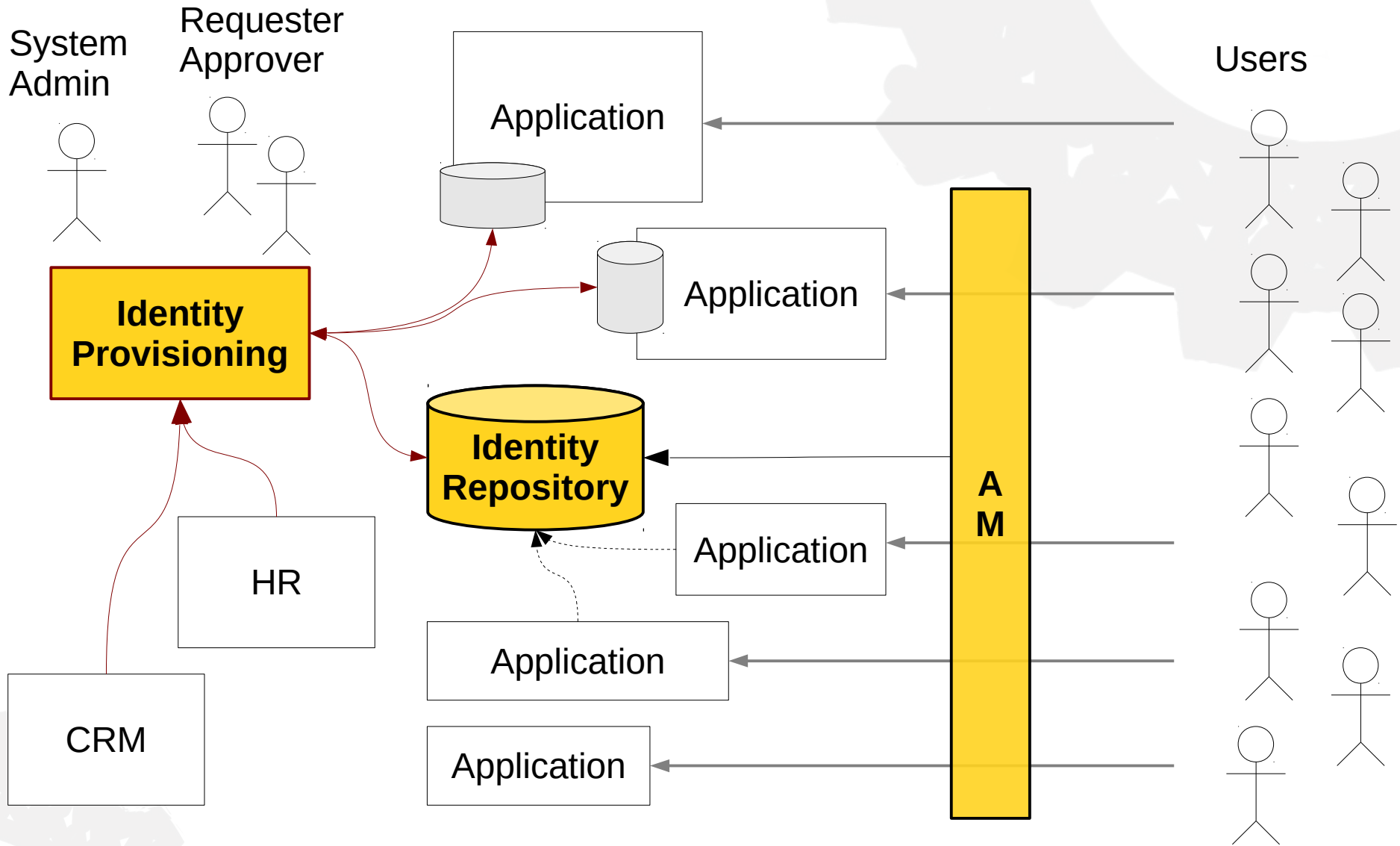
*) nobody really knows how this part works because the guy that did it left 3 years ago

**LDAP-only solutions work
only in simple cases.**

Evolveum

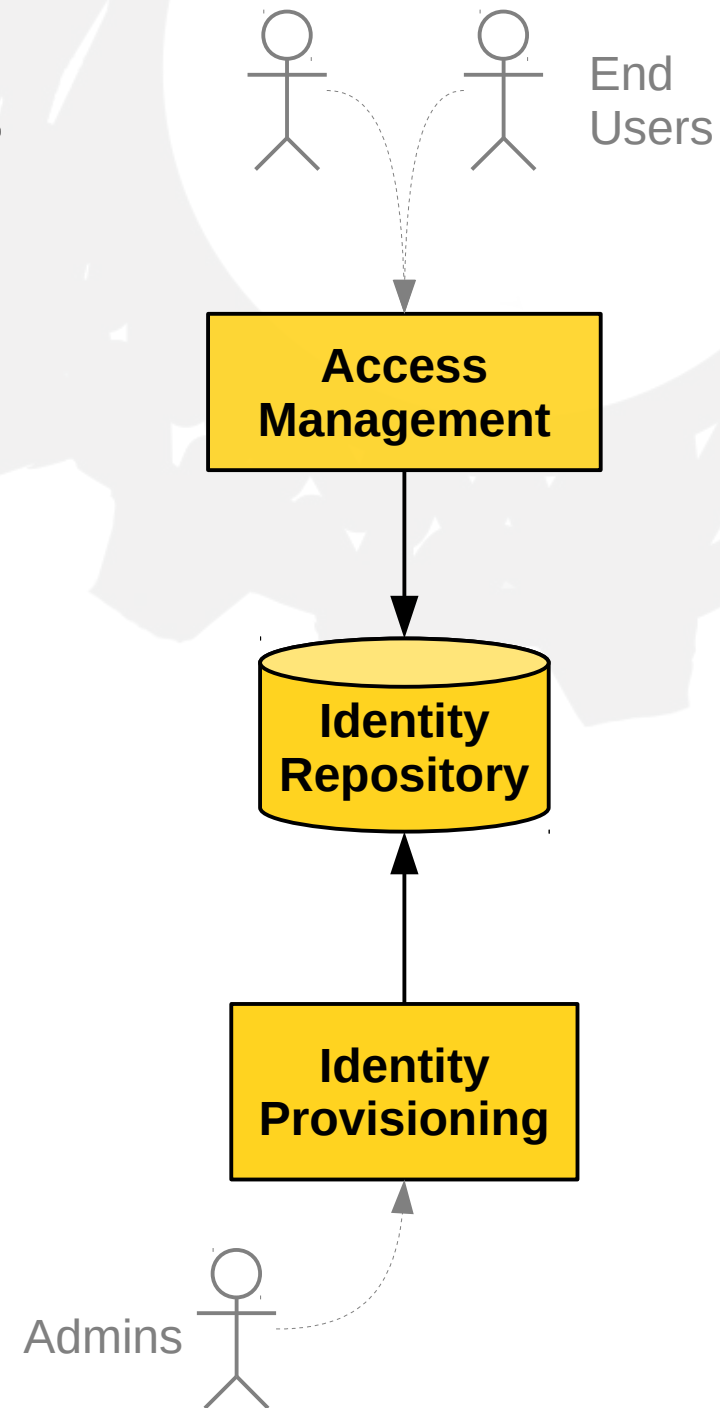


IAM needs more components



Basic IAM Components

- **Access Management**
 - Authentication, single sign-on
 - Basic authorization
- **Identity Repository**
 - Storage of identity data
- **Identity Provisioning**
 - Management (data, policies, workflows)
 - Synchronization



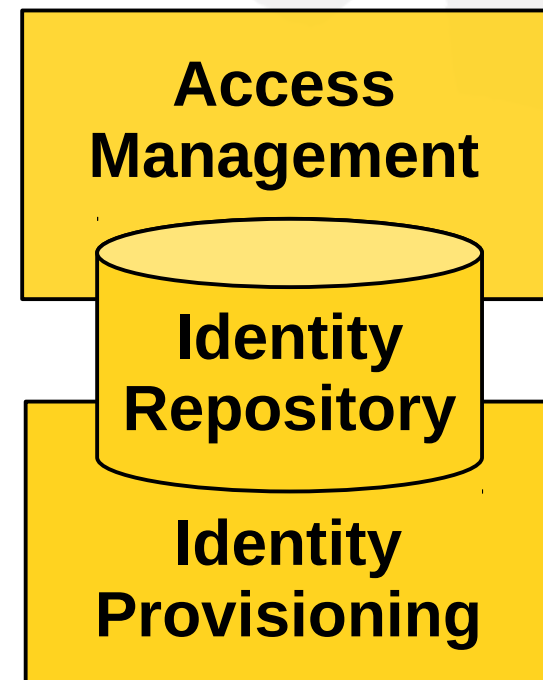
Interoperability

- The components should work together
as **one system**
- Easy product integration
- Smooth user experience
 - The user should not see component boundaries

Technology stacks

“Stack” is the obvious answer to interoperability problem.

... Or ... **is it?**



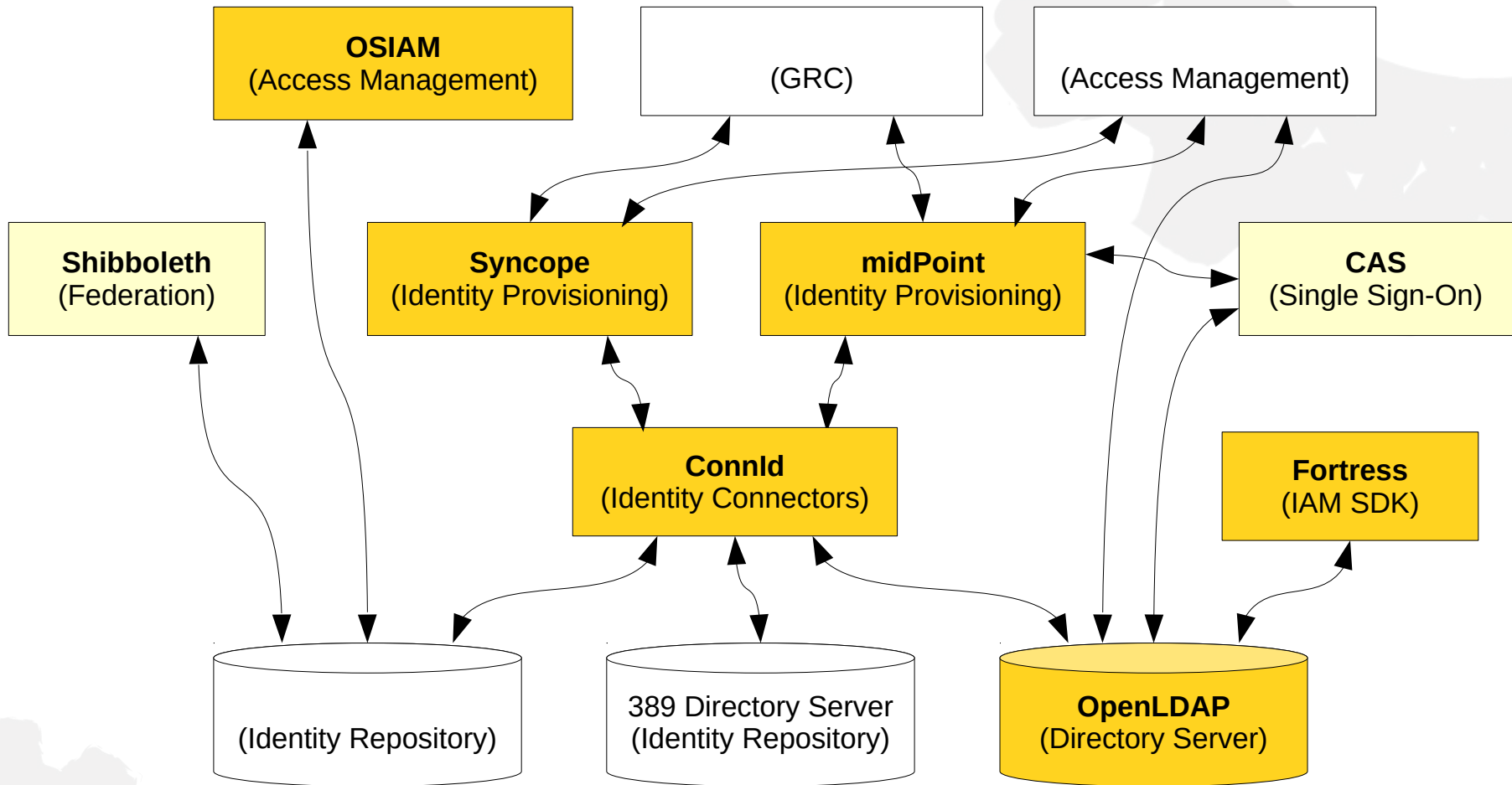
What's wrong with stacks?

- Usually **single-vendor** stacks
- Still quite **heterogeneous** due to acquisitions
- **Vendor lock-in**
 - You can check out any time you like, but you can never leave
- **Limited** integration options
 - Just one option for each component
 - Proprietary interfaces

Is there any better way?

The Ecosystem

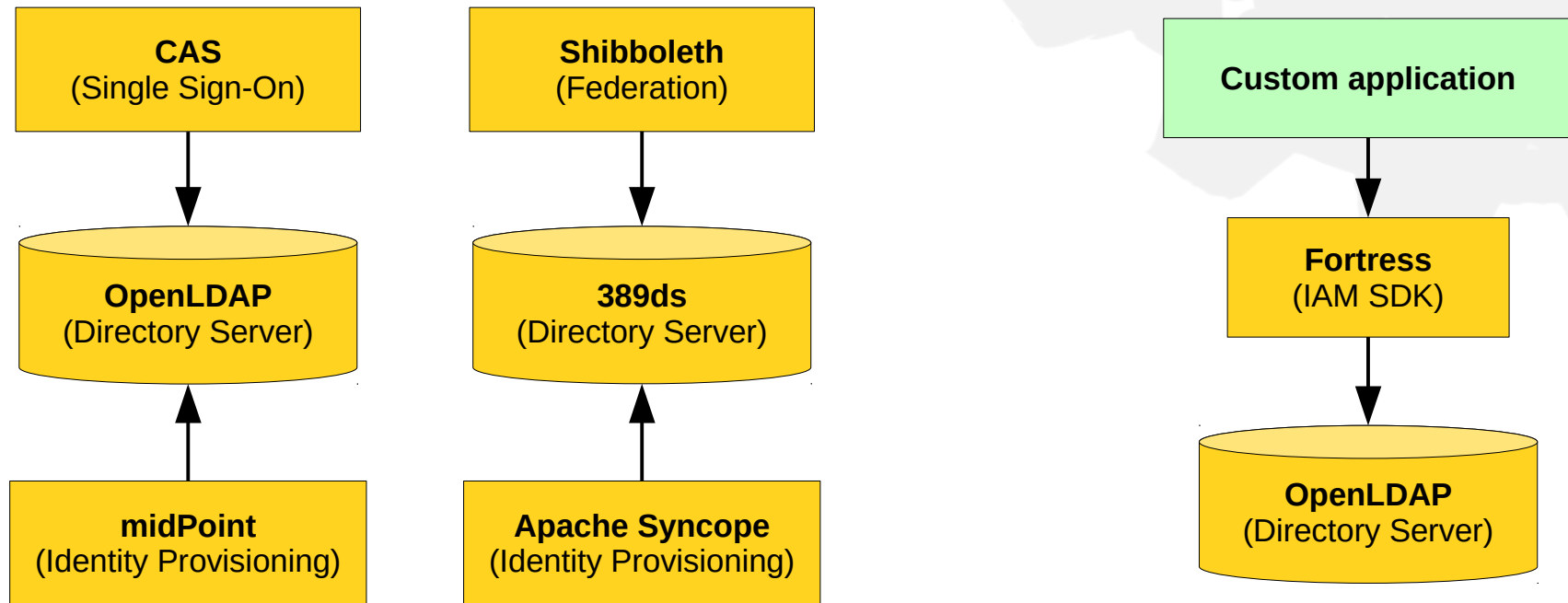
Open Source Identity Ecosystem



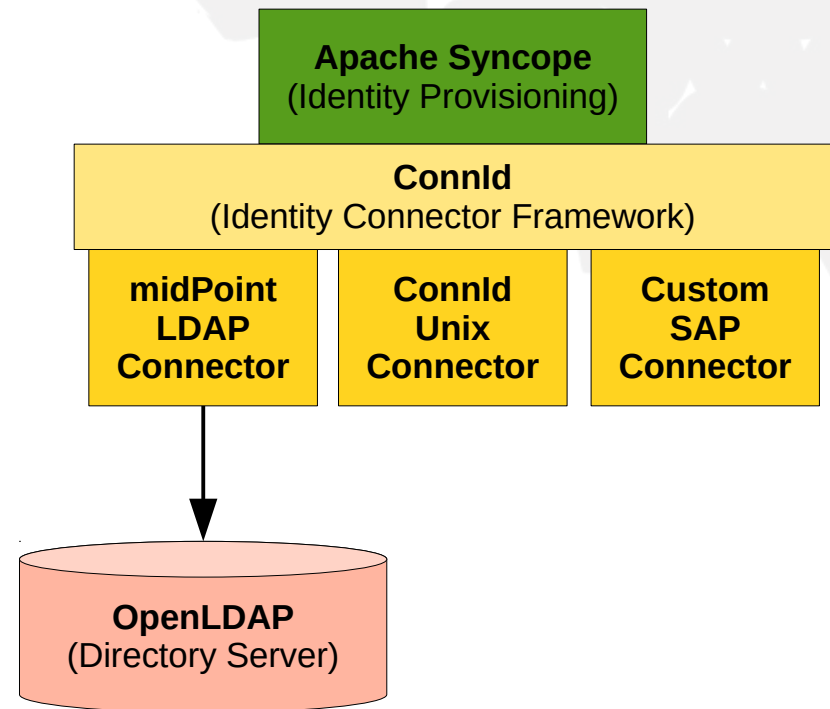
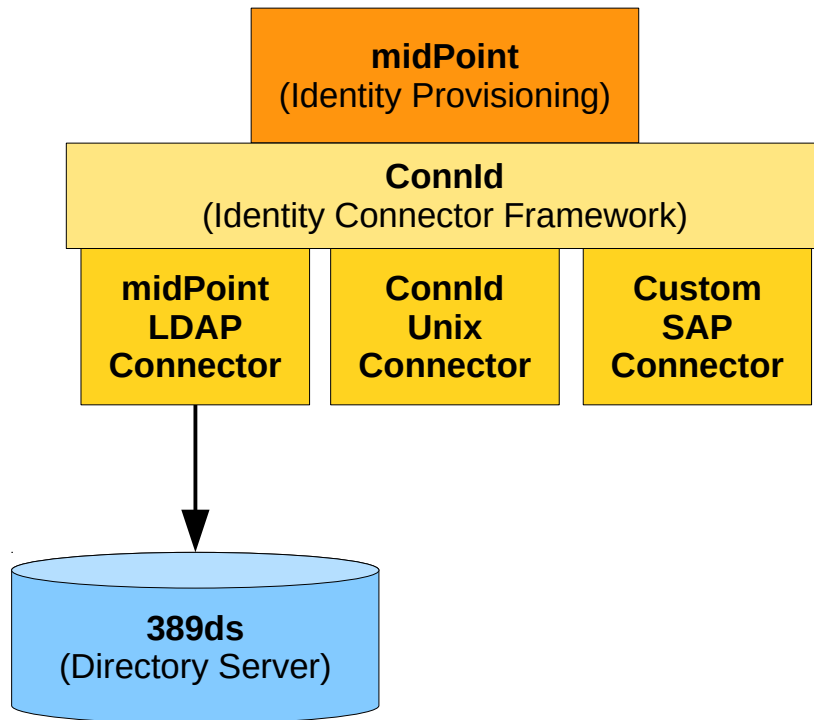
Open Source Identity Ecosystem

- Pure **open source** model
 - Any engineer can have complete understanding of the technology
 - Technological excellence and efficiency
- Standardized or open source **interfaces**
 - Unlimited integration options
 - Replaceable components → no vendor lock-in
- **Cooperation** instead of domination
 - Trade influence for control to get substantial benefits

Ecosystem Deployment Examples



Ecosystem Deployment Examples



We know that it works, because ...

- we have **tested** the technology
 - test suites, pilots, real projects
- we share the same **goal**
- there are **business** agreements in place

Join the **Ecosystem** now!

Evolveum



Questions and Answers



Thank You

Radovan Semančík

www.evolveum.com