



**How To Maintain Million Lines
Of Open Source Code
And Remain Sane***
or
The Story of MidPoint

Radovan Semančík
Rubyslava, October 2018

* More or less. More less than more.

Radovan Semančík

Software Architect at Evolveum

Architect of midPoint

Apache Foundation Committer

Contributor to several open source projects

Still (more or less) sane





WARNING

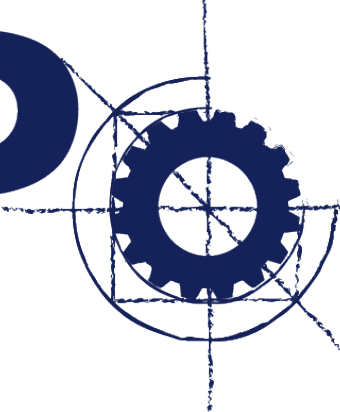
HIC SUNT LEONES

Controversial statements ahead!
Political correctness (very) limited.
Mental health hazards.
Dogmatic buzzword followers should leave now.

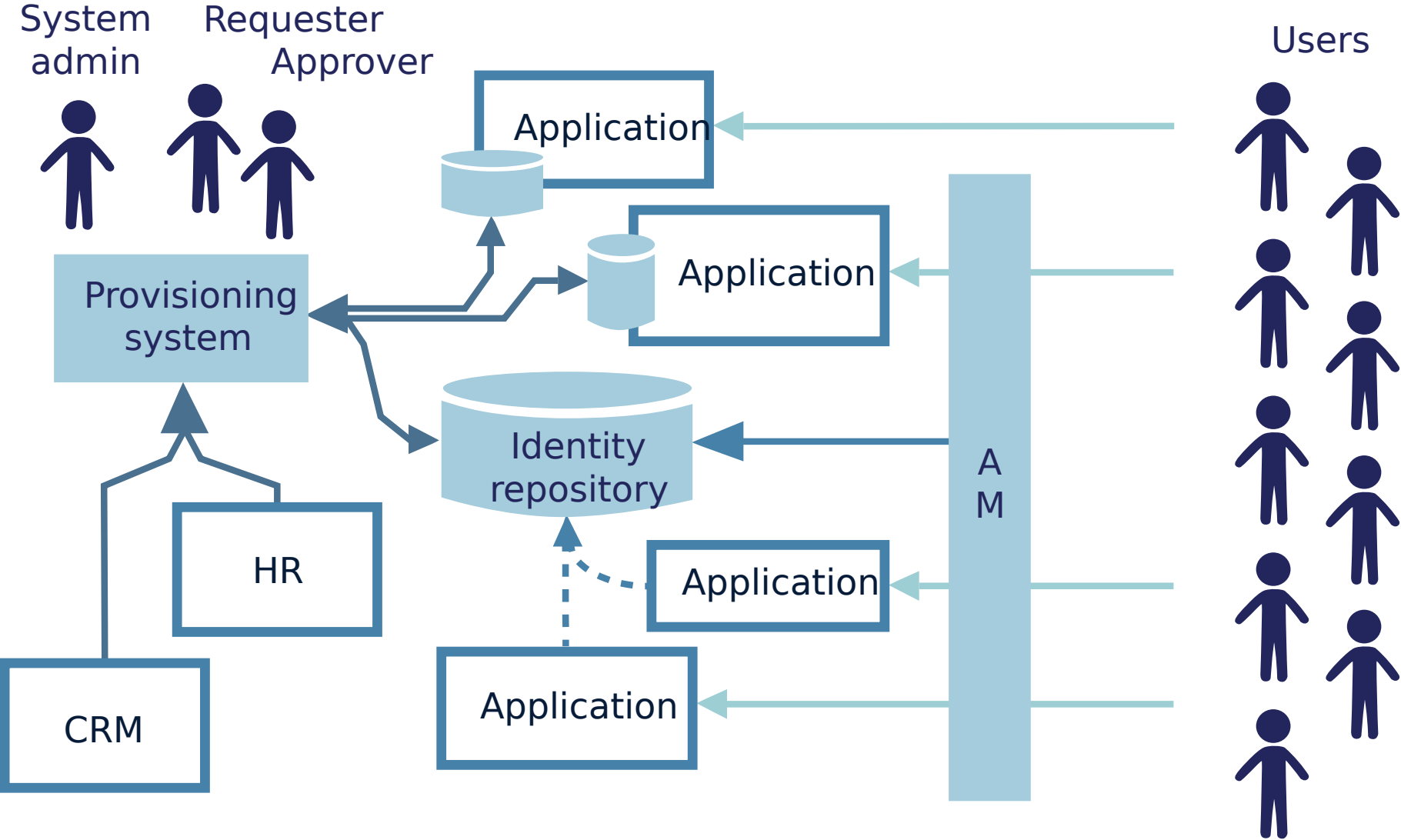
Project midPoint

- Identity management and governance
- Open source (Apache License)
- Started in 2011 by Evolveum (self-funded)
- Approx. 1 million lines of code
- Mostly written in Java

midPoint

The logo for 'midPoint' features the word 'midPoint' in a bold, dark blue, sans-serif font. The letter 'o' in 'Point' is replaced by a stylized gear or compass rose symbol, which consists of a circle with a gear-like outer edge and a crosshair pattern inside.

What is Identity Management?



... and Identity Governance?

- Beyond Role-Based Access Control (RBAC)
- Organizational structure
- Delegation, Audit, etc.
- Role assignment and re-certification
- Policies (e.g. SoD)
- Maintenance of role model (role lifecycle)
- Risk assessment
- Compliance

SELF SERVICE

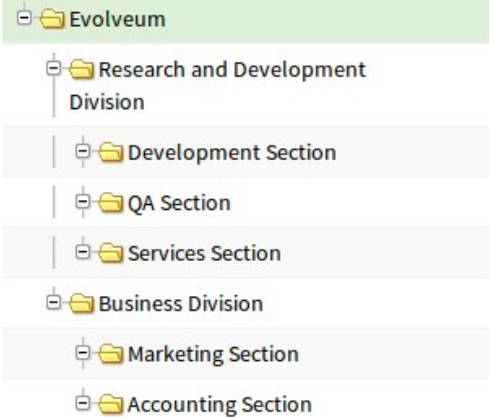
- Home
- Profile
- Credentials
- Request a role

ADMINISTRATION

- Dashboard
- Users
- Org. structure
- Organization tree
- New organization
- Roles
- Services
- Resources
- Work items
- Certification
- Server tasks
- Reports
- Configuration

- Evolveum
- Projects
- Role Catalog

Org. hierarchy



Managers



Mgr. Igor Farinič (ifarinic)
CEO

Enabled
Manager

Members

One level | Object

Name: All | More... | Advanced

Type	Name	Fullname/Display name	Identifier/Description	
	F1100	Research and Development Division	1100	
	F1200	Business Division	1200	
	ifarinic	Mgr. Igor Farinič	igor.farinic@evolveum.com	

+ | ↺ | ↻ | 1 to 3 of 3 | << | < | 1 | > | >> | ⚙

SELF SERVICE

- Home
- Profile
- Credentials
- Request a role

ADMINISTRATION

- Dashboard
- Users ▼
 - List users
 - Edit user**
 - New user
- Org. structure <
- Roles <
- Services <
- Resources <
- Work items 1
- Certification <
- Server tasks <
- Reports <
- Configuration <



**Ing. Katarína Valalíková** (katkav)

Software Developer
Development Section

 Enabled End user Manager

- Basic
- Projections 1
- Assignments 5
- Tasks 2
- Request a role
- History
- Delegations 0
- Delegated to me 1

Properties

Name *	<input type="text" value="katkav"/>
Full name	<input type="text" value="Ing. Katarina Valalíková"/>
Given name	<input type="text" value="Katarina"/>
Family name	<input type="text" value="Valalíková"/>
Honorific Prefix	<input type="text" value="Ing."/>
Title	<input type="text" value="Software Developer"/>
Email Address	<input type="text" value="katarika.valalikova@evolveum.com"/>
Employee Number	<input type="text" value="003"/>
Locality	<input type="text" value="Bratislava"/>
Jpeg photo	<input type="button" value="Browse..."/> No file selected.  

Activation

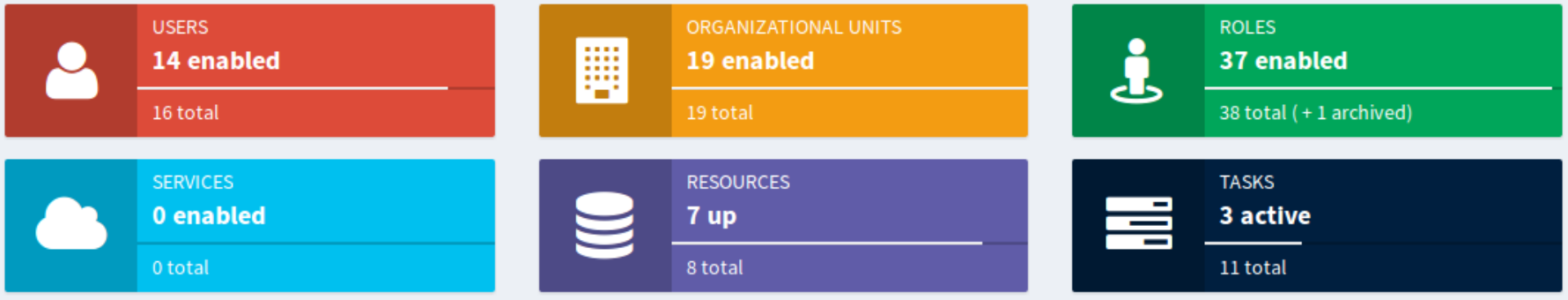
Lock-out Status: Normal

Password

Password: password is set

Metadata

Create timestamp: May 9, 2016 10:16:07 AM



midPoint Resource details Resources > Resources List > Resource details administrator

OpenLDAP UP

Details | Defined Tasks | Accounts | Entitlements | Generics | Uncategorized | Connector

RESOURCE IS UP **LdapConnector** 1.4.3

MAPPINGS **Source and Target** Synchronization defined

SCHEMA **3 object types** 79 schema definitions

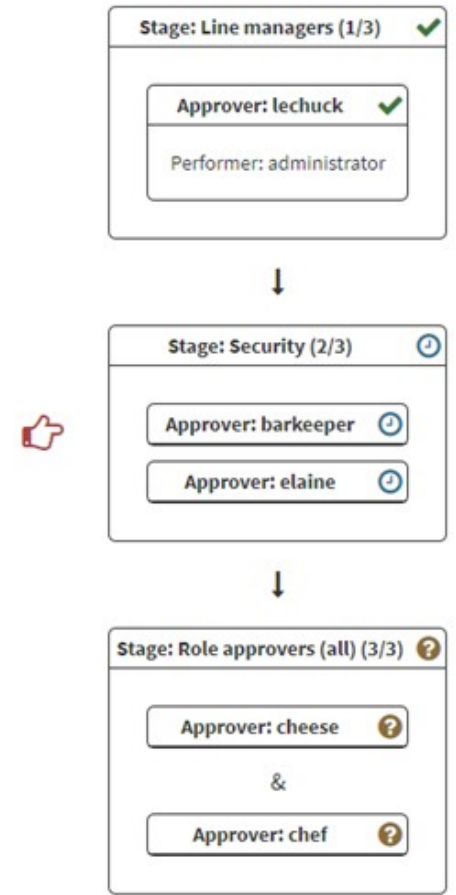
Capabilities

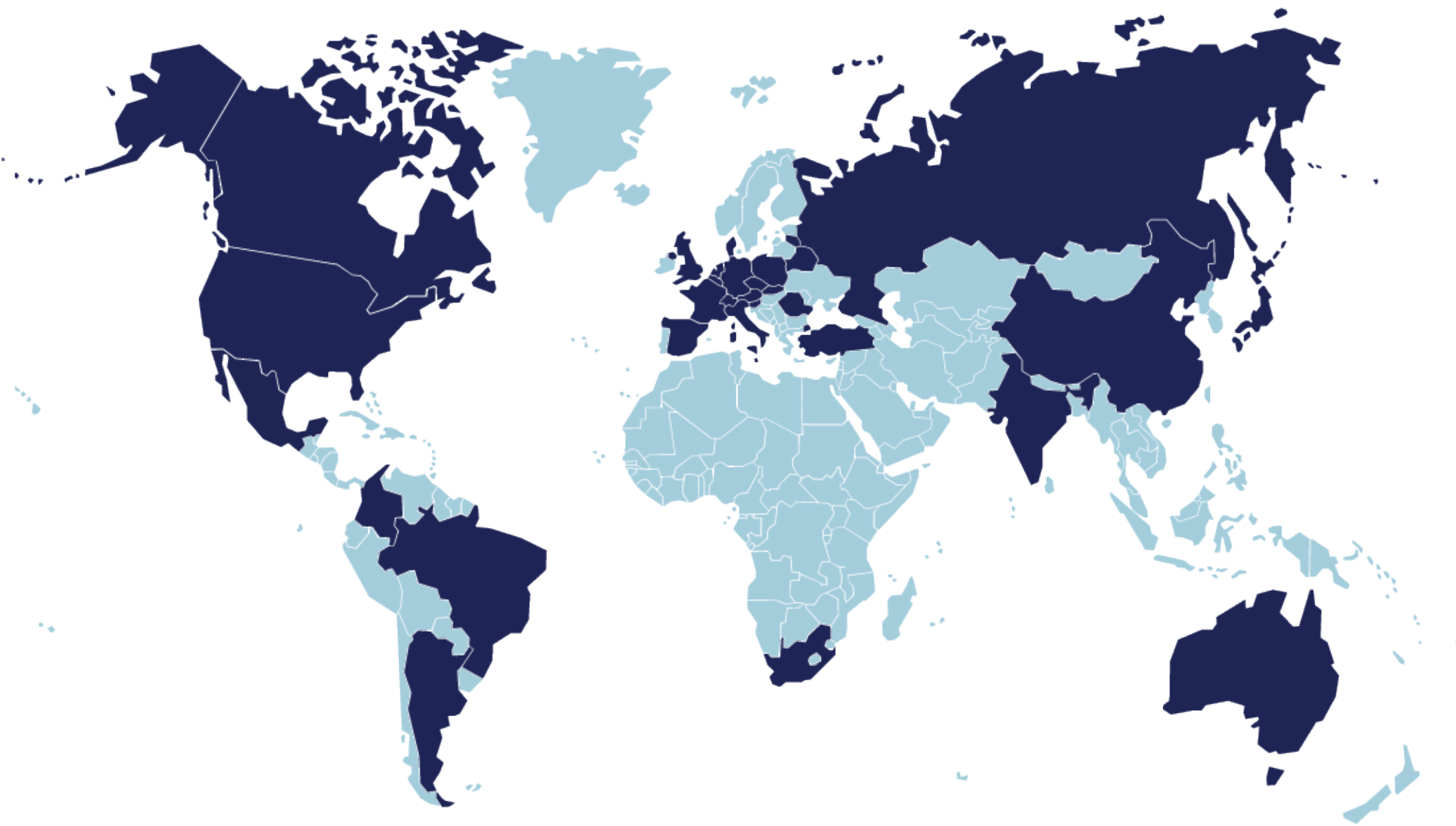
- Activation, Activation Lockout, Activation Status, Activation Validity
- Credentials, Password, Live sync, Test Connection, Script
- Auxiliary Object Classes, Create, Update, Add/Remove Values, Delete, Read, Count Objects, Paged Search

Kind	Object Class	Intent	Synchronization	Tasks
ACCOUNT	inetOrgPerson		true	
ENTITLEMENT	groupOfNames	ldapGroup	true	
ENTITLEMENT	posixGroup	posixGroup	true	

1 to 3 of 3 << < 1 > >> ⚙️

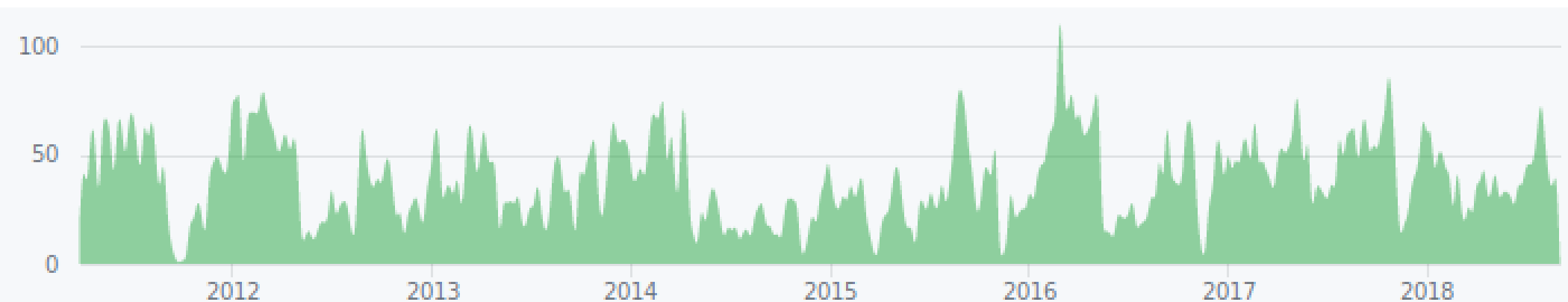
Back | Test connection | Refresh schema | Edit configuration | Show using wizard | Edit using wizard | Edit XML



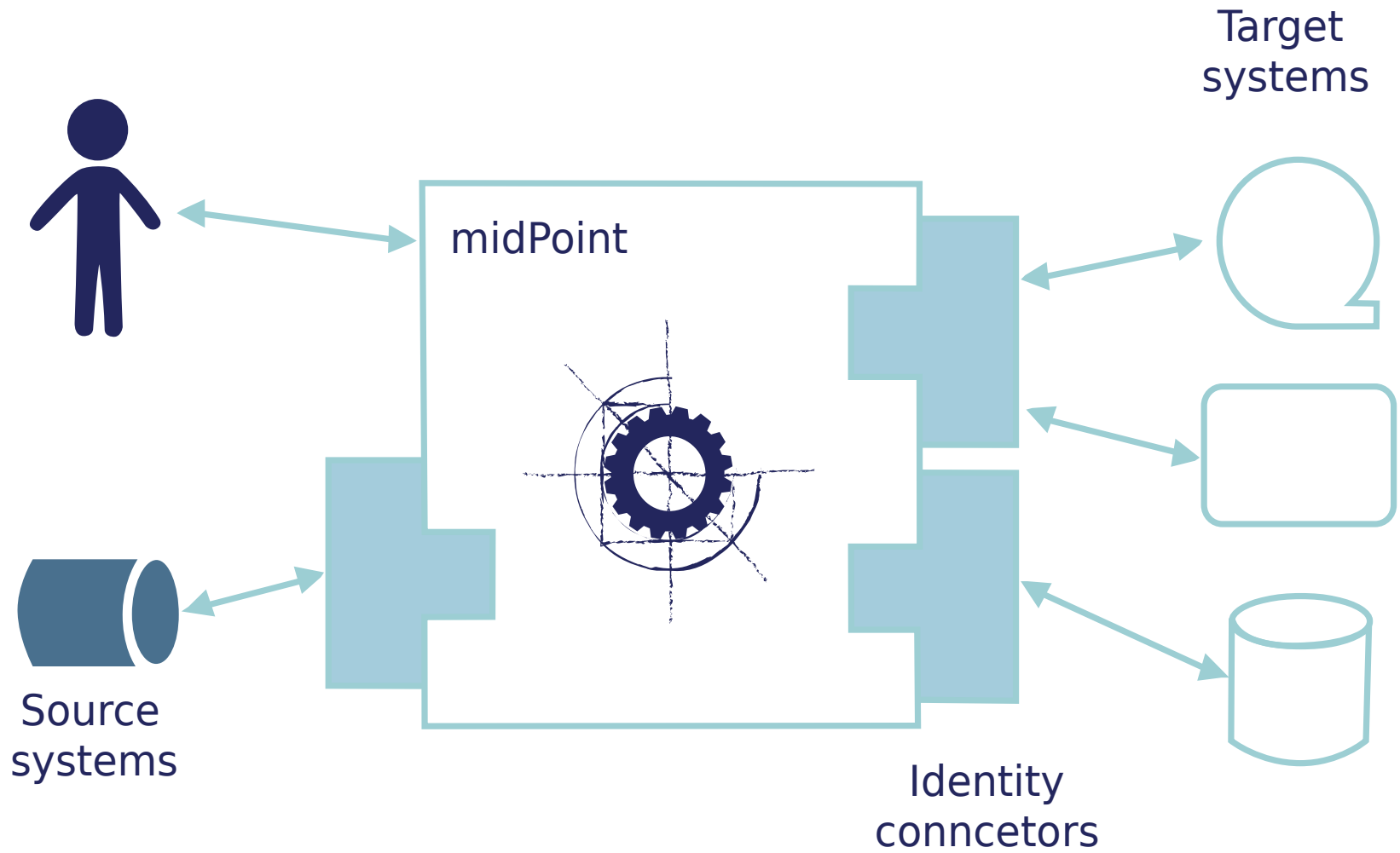


MidPoint Development

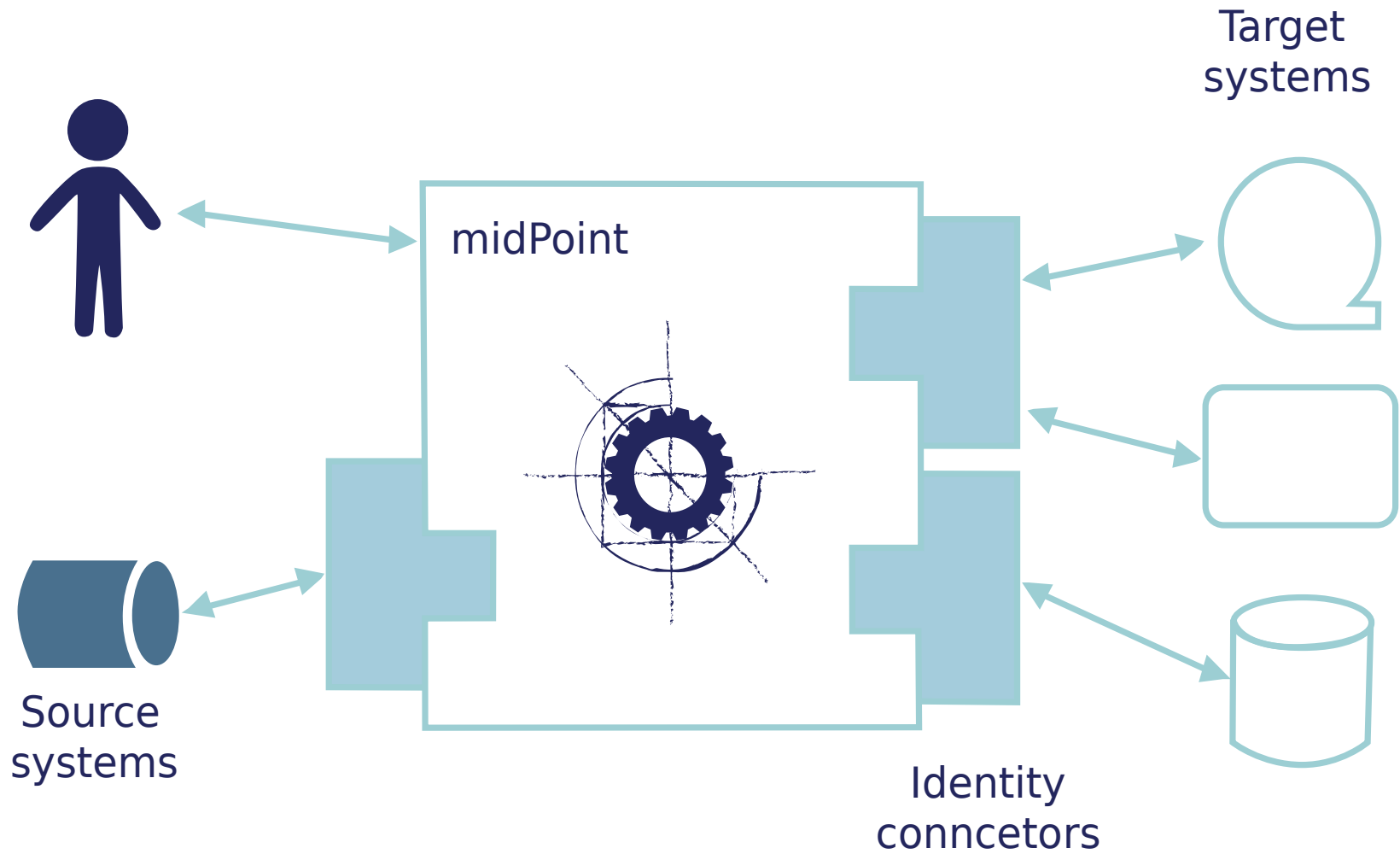
- Everything is open source (see github)
- Evolutionary approach (iterative+incremental)
- At least 2 releases per year (26 releases)
- Team of 5 full-time developers (+contributors)
- High development activity (100-200 commits/month)



MidPoint Big Picture

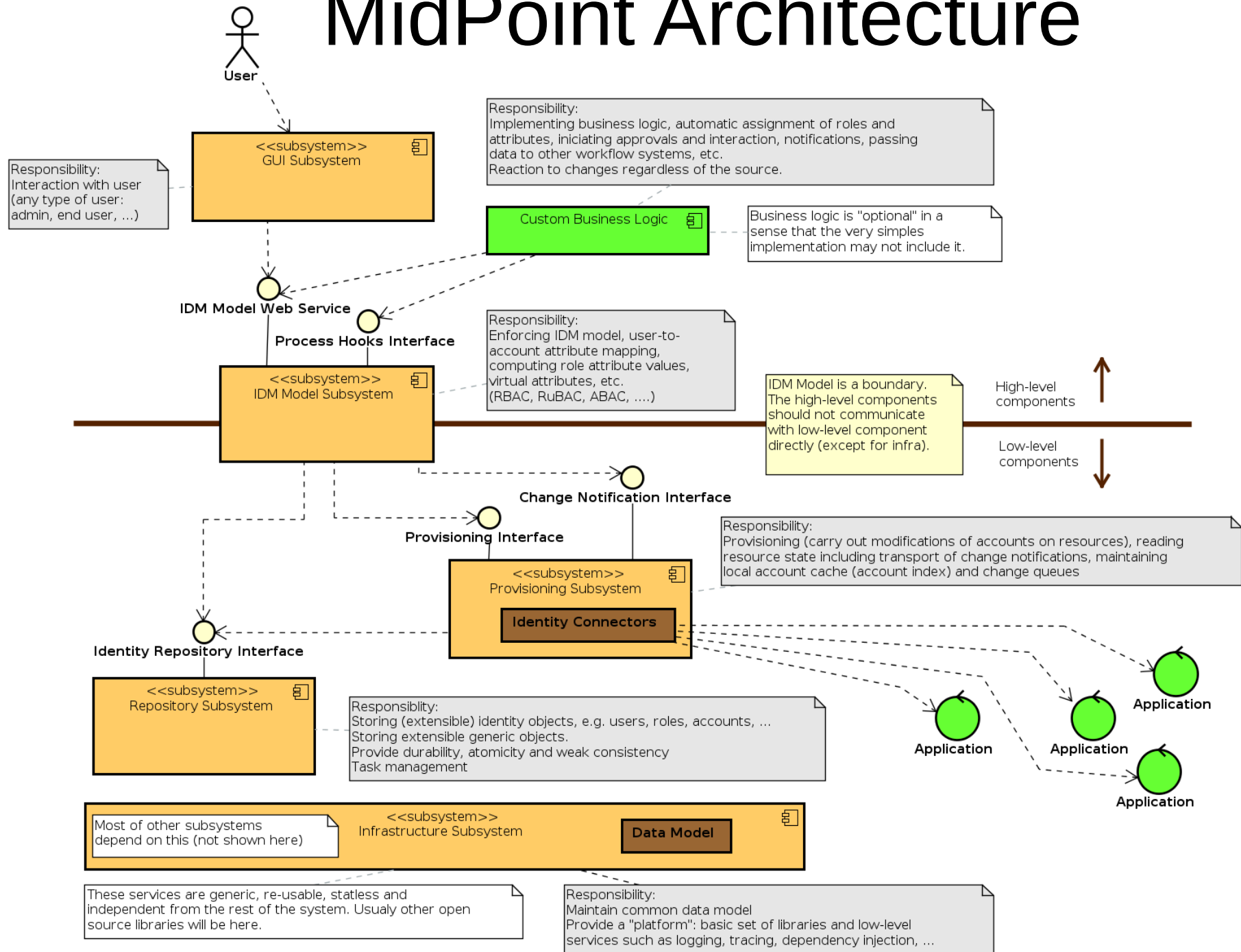


MidPoint Big Picture



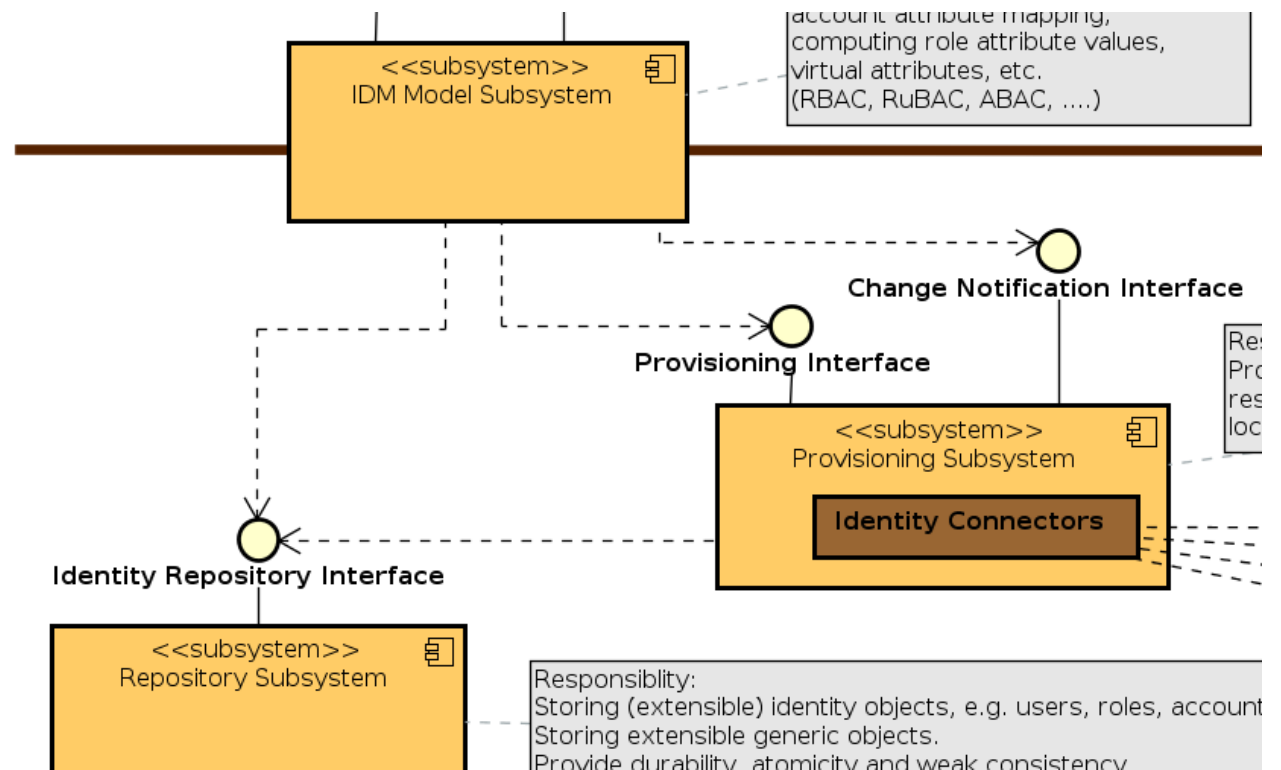
Monolith? Not really!

MidPoint Architecture



Components, Source Code Structure

- ▼ infra
 - ▶ common
 - ▶ maven
 - ▶ prism
 - ▶ prism-maven-plugin
 - ▶ schema
 - ▶ schema-pure-jaxb
 - ▶ target
 - ▶ test-util
 - ▶ util
 - ▶ ws-util
- ▶ maven
- ▼ model
 - ▶ certification-api
 - ▶ certification-impl
 - ▶ maven
 - ▶ model-api
 - ▶ model-client
 - ▶ model-common
 - ▶ model-impl
 - ▶ model-intest
 - ▶ model-test
 - ▶ notifications
 - ▶ notifications-api
 - ▶ notifications-impl
 - ▶ report-api
 - ▶ report-impl
 - ▶ target
 - ▶ workflow-api
 - ▶ workflow-impl
- ▶ provisioning



Dependencies (2010-2012)

- Spring
- Java Server Faces
- XML (DOM)
- JAX-B
- JAX-WS
- ESB (BPEL)
- Activiti BPM (BPMN.2)
- Jasper Reports
- Hibernate

Dependencies (2018)

- Spring + Spring Boot
- ~~Java Server Faces~~ Apache Wicket
- XML (DOM) + JSON + YAML
- ~~JAX-B~~ : (almost) replaced
- ~~JAX-WS~~ : not used much any more
- ~~ESB (BPEL)~~ : replaced before midPoint started
- ~~Activiti BPM (BPMN.2)~~ : going to be replaced
- Jasper Reports : not very useful, will it survive?
- Hibernate : may be replaced later on

Dependencies : Lessons Learned

- Faster start of the project
- Do not reinvent the wheel
 - ... unless the wheel is in fact a square
- Do not depend on dependencies too much
- Understand how they work – and why they fail
- Have a “Plan B” to replace them later on

Architecture?

“REST”, Microservices, Web frameworks, ...

That's not architecture!

Architecture!

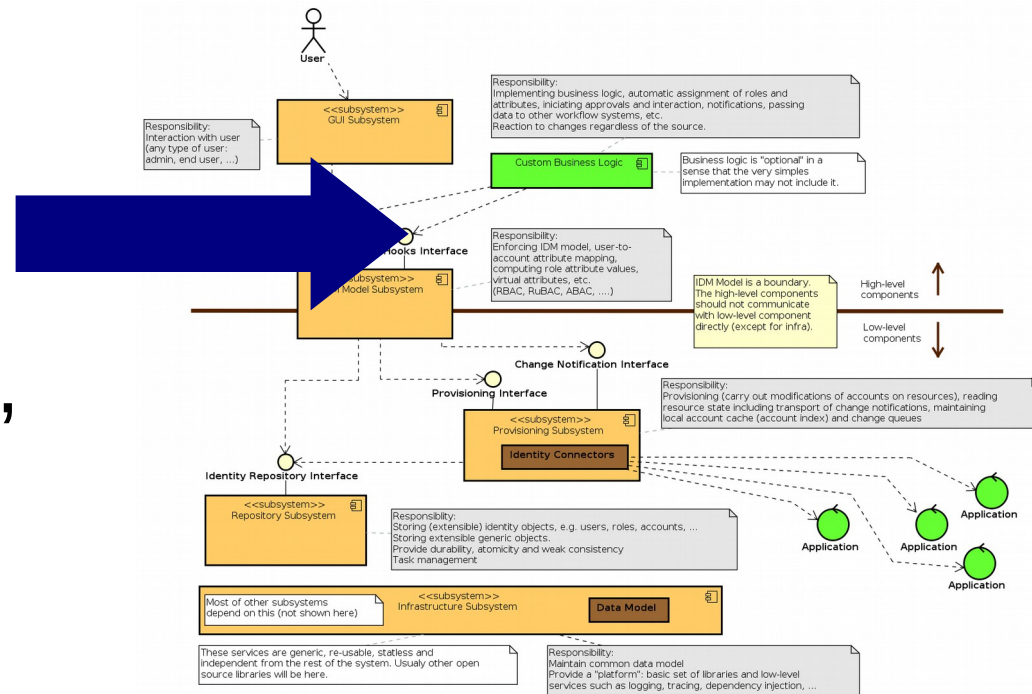
“REST”, Microservices, Web frameworks, ...

That's **not** architecture!

This is architecture

Components, subsystems,
interfaces, modules,
separation of concerns

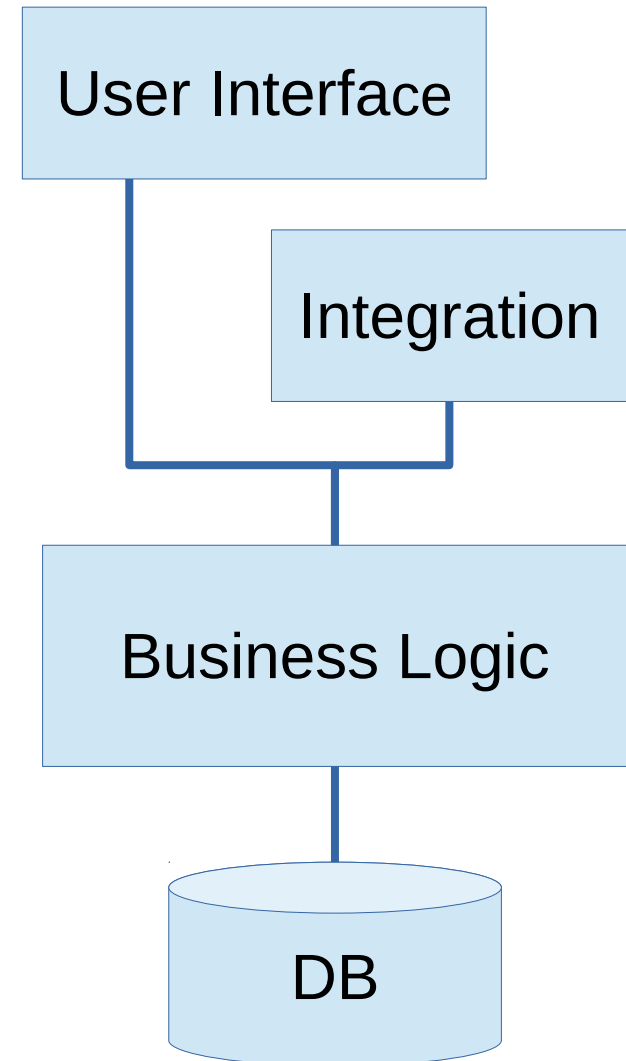
You really should pay attention in
software engineering classes.



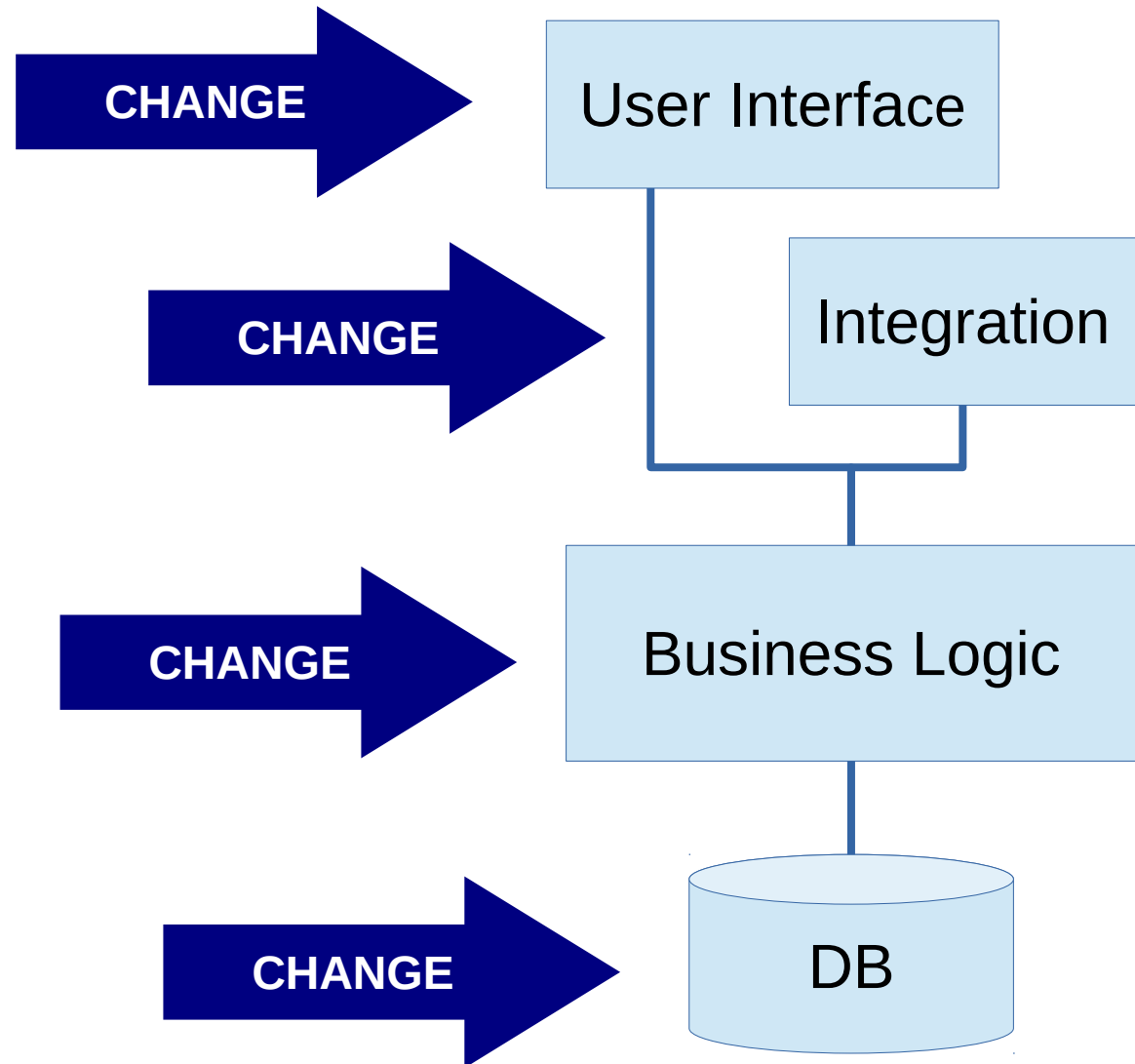
Data Model

- Extremely important
- As important as architecture
- Cross-cutting concern
- Performance, scalability, evolvability, ...
- Changes often – especially at the beginning
- Evolution - compatibility
- Experimental features

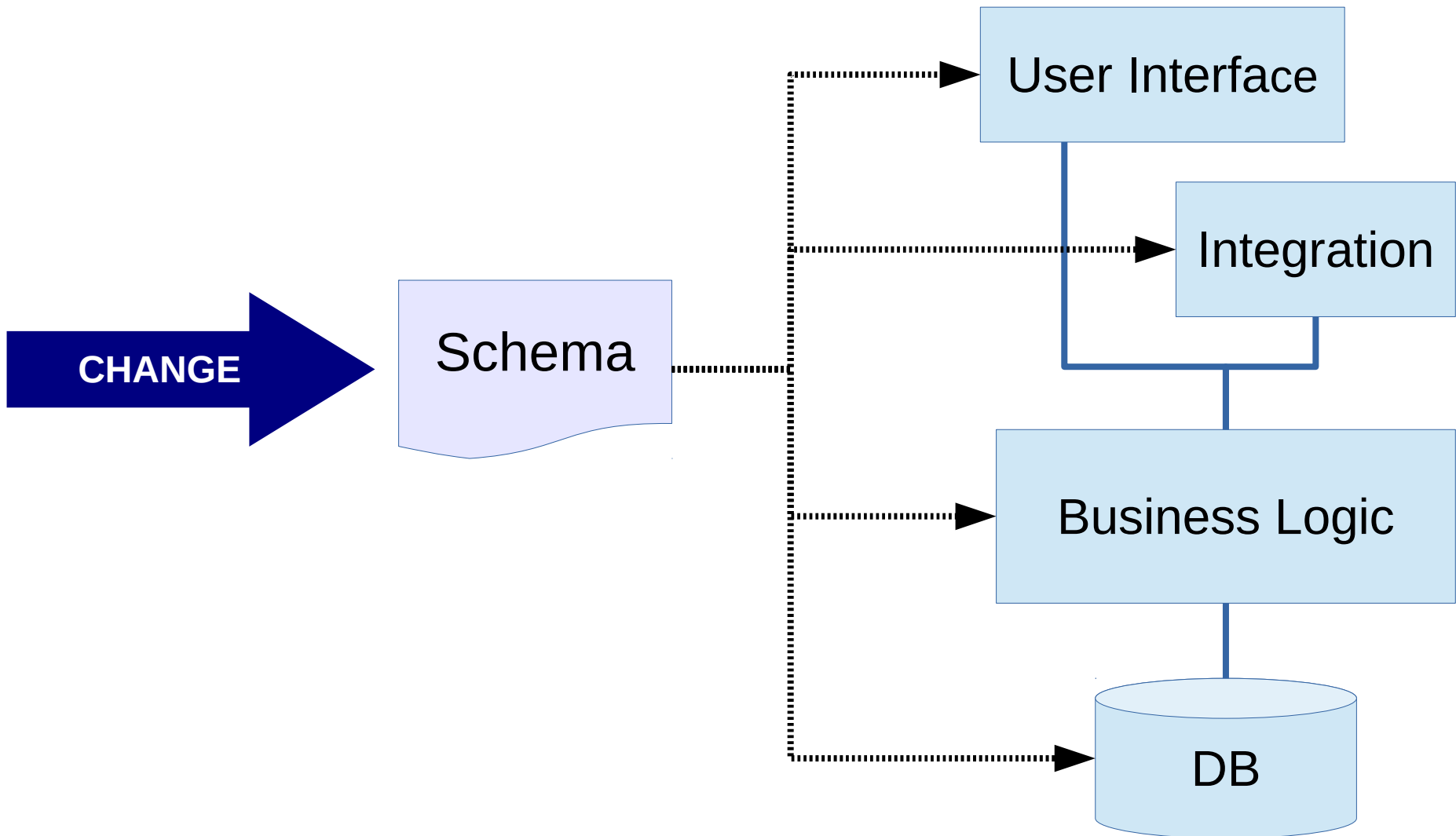
Data Model



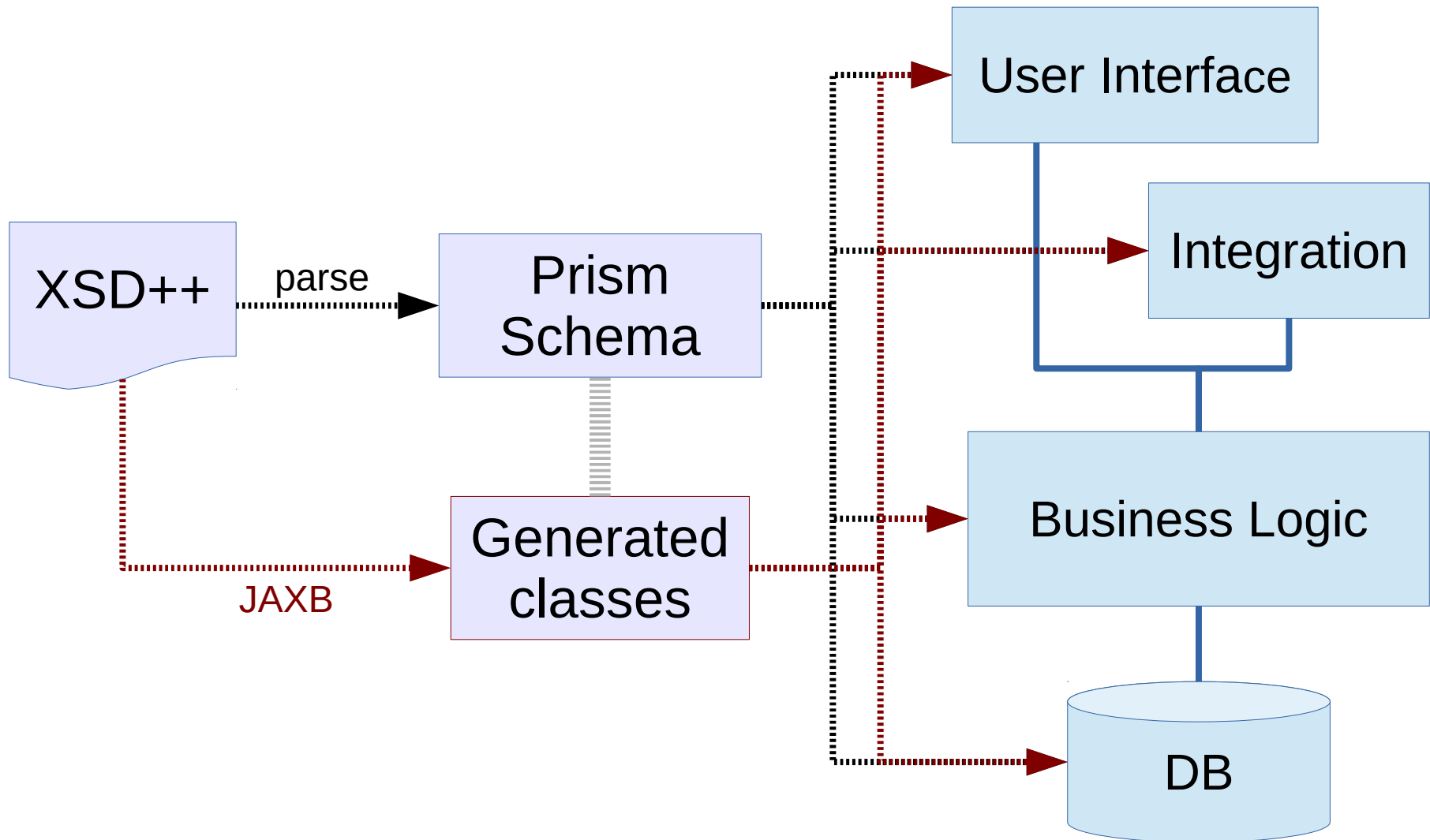
Data Model Change



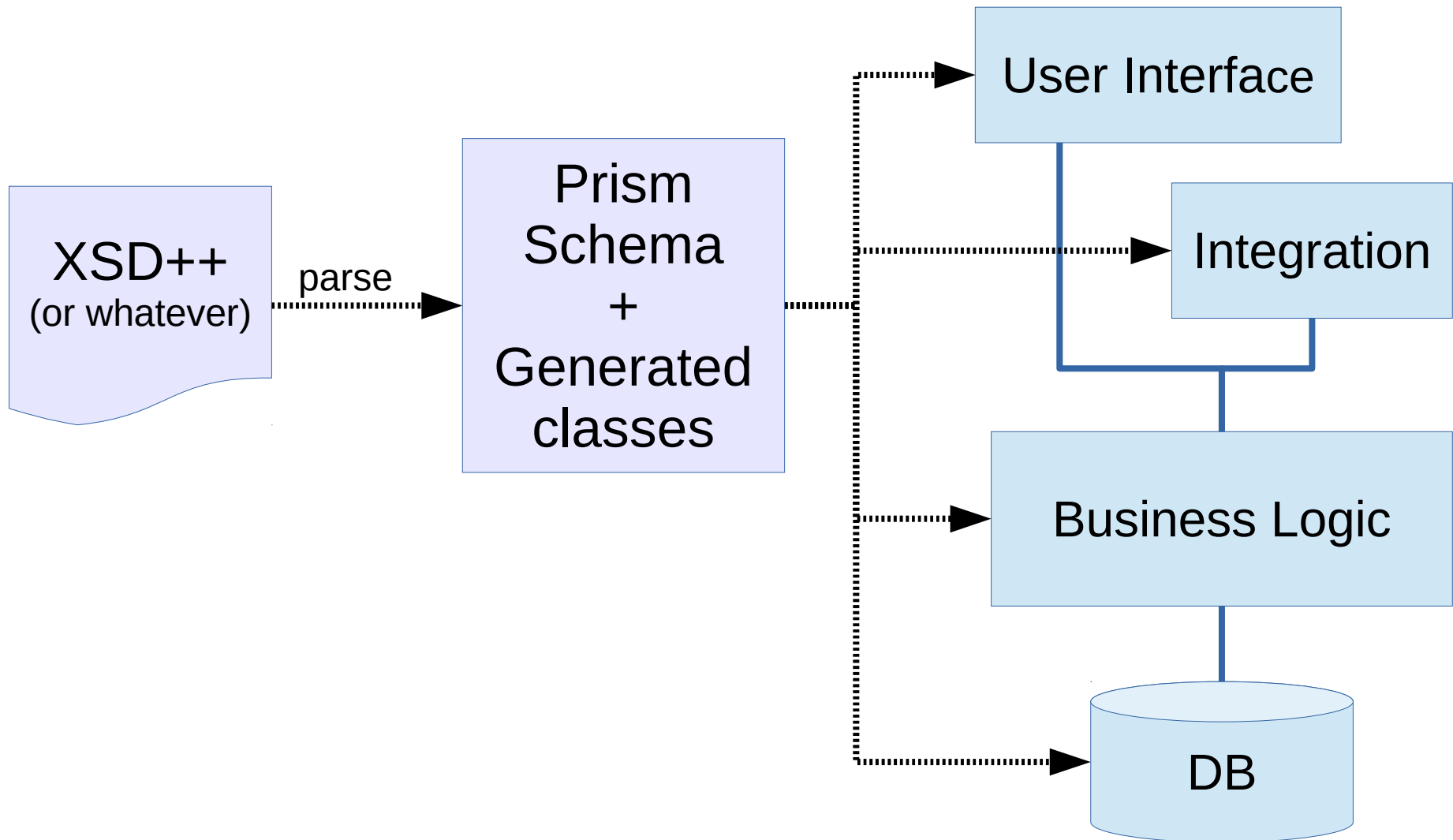
Data Model : Schema



MidPoint : Prism Schema (now)



MidPoint : Prism Schema (future)



MidPoint : Prism Schema in UI

 **Foo Bar**
(foo)

✓ Enabled
End user
✗ No organizations

Basic | Projections **1** | Assignments **3** | History | Tasks **0** | Personas | Delegations **0** | Delegated to me **0**

▼ Properties * ↓

Name *	foo
Full name	Foo Bar
Given name	Foo
Family name	Bar
SSN	1234567890

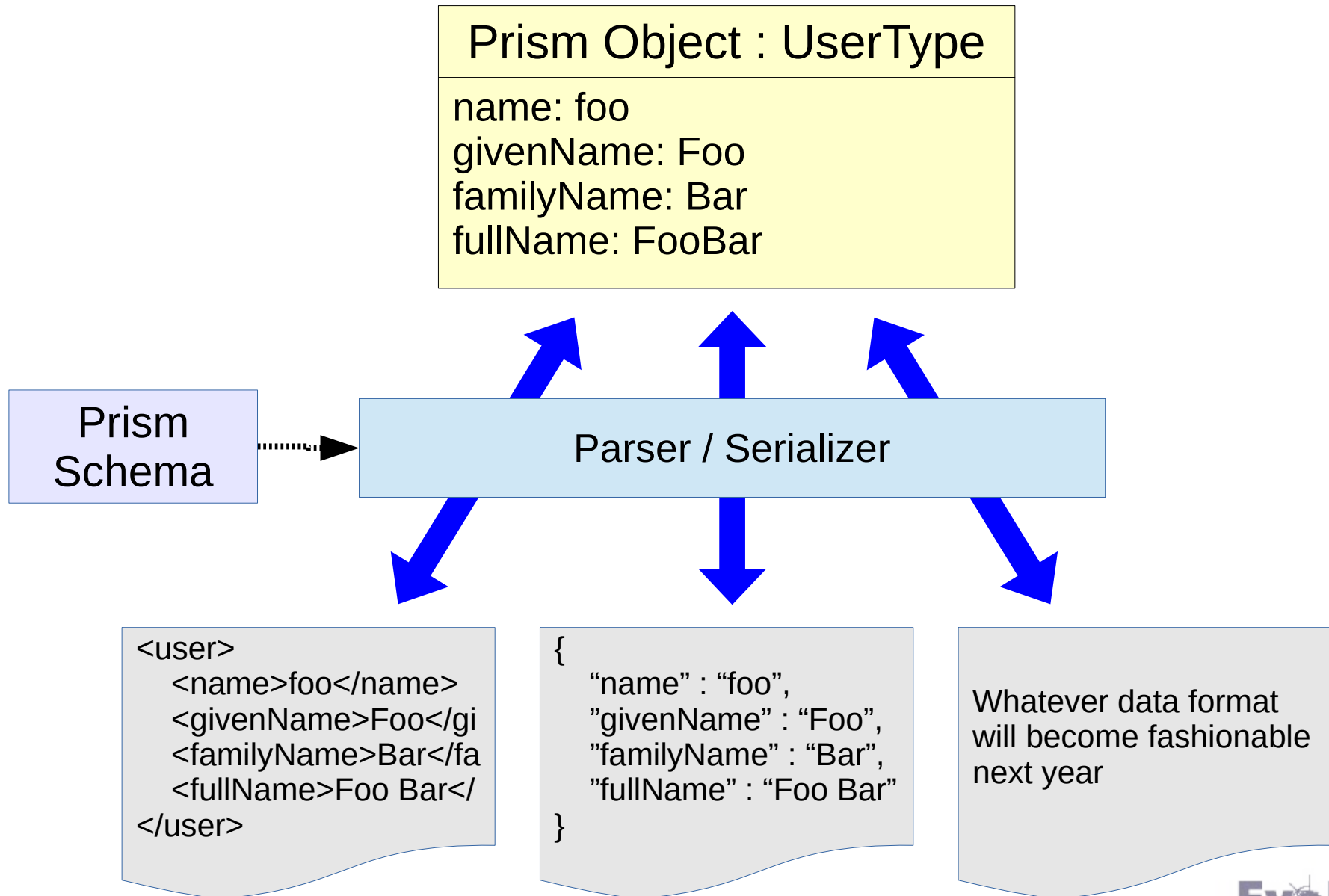
Show empty fields



▼ Activation ⓘ ↓

Lock-out Status ⓘ Normal [Set to "Normal"](#)

XML, JSON, YAML and Friends



XML, JSON, YAML and Friends

The diagram illustrates the transformation of data from XML to JSON and then to YAML. It consists of three screenshots of a code editor, each with a green arrow pointing to the left, indicating the flow of data from left to right.

Top Screenshot (XML): Shows an XML document with the following structure:

```
1 <user xmlns="http://midpoint.evolveum.com/xml/ns/public/common/common-3" xmlns:c="http://midpoint.evolveum.com/xml/ns/public/common/common-3" xmlns:icfs="http://midpoint.evolveum.com/xml/ns/public/common/extension-3">
2   <name>foo</name>
3   <extension xmlns:gen165="http://midpoint.evolveum.com/xml/ns/samples/extension-3">
4     <gen165:ssn>1234567890</gen165:ssn>
5   </extension>
6   <metadata>
7     <requestTimestamp>2017-12-15T09:40:35.277Z</requestTimestamp>
8     <requestorRef oid="00000000-0000-0000-0000-000000000002" relation="http://midpoint.evolveum.com/xml/ns/public/common/org-3#default" type="http://midpoint.evolveum.com/xml/ns/public/common/common-3#UserType"/>
9     <createTimestamp>2017-12-15T09:40:37.402Z</createTimestamp>
10    <creatorRef oid="00000000-0000-0000-0000-000000000002" relation="http://midpoint.evolveum.com/xml/ns/public/common/org-3#default" type="http://midpoint.evolveum.com/xml/ns/public/common/common-3#UserType"/>
11    <createChannel href="http://midpoint.evolveum.com/xml/ns/public/common/org-3#default" type="http://midpoint.evolveum.com/xml/ns/public/common/common-3#UserType"/>
12    <modifyTimestamp>2017-12-15T09:40:37.402Z</modifyTimestamp>
13    <modifierRef oid="00000000-0000-0000-0000-000000000002" relation="http://midpoint.evolveum.com/xml/ns/public/common/org-3#default" type="http://midpoint.evolveum.com/xml/ns/public/common/common-3#UserType"/>
14    <modifyChannel href="http://midpoint.evolveum.com/xml/ns/public/common/org-3#default" type="http://midpoint.evolveum.com/xml/ns/public/common/common-3#UserType"/>
15    <lastProvisioningTime>2017-12-15T09:40:35.277Z</lastProvisioningTime>
16  </metadata>
17 </operationExecution>
18 <timestamp>2018-01-01T00:00:00.000Z</timestamp>
19 <operation>
20   <objectDelta>
21     <t:change>
22       <t:object>
23         </objectDelta>
24   </operation>
25 </operationExecution>
26 </user>
```

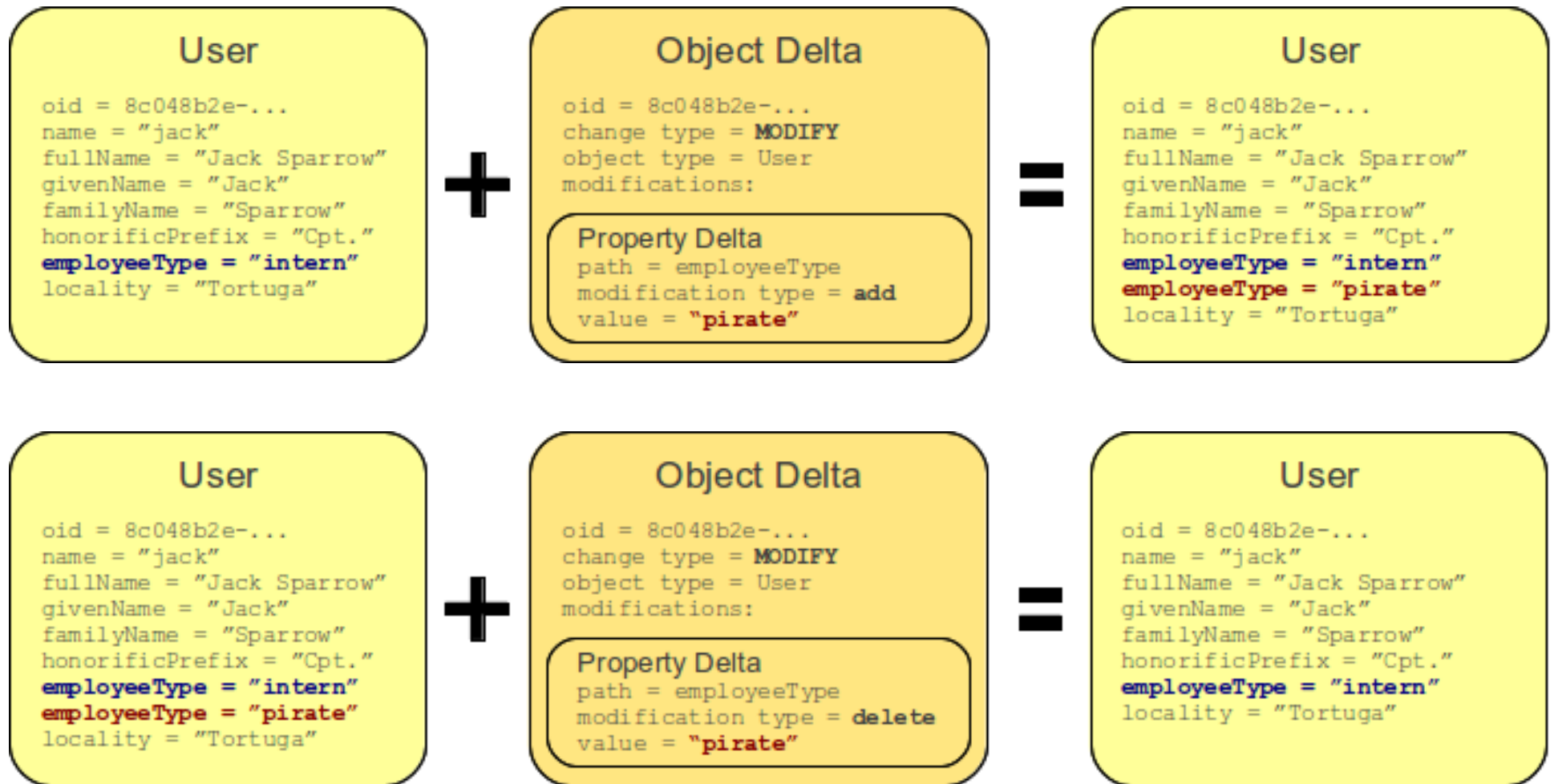
Middle Screenshot (JSON): Shows the XML converted to JSON:

```
1 {
2   "@ns": "http://midpoint.evolveum.com/xml/ns/public/common/common-3",
3   "user": {
4     "oid": "fc41bc2d-6f81-45f3-a103-6e05741fb882",
5     "version": "26",
6     "name": "foo",
7     "extension": {
8       "@ns": "http://midpoint.evolveum.com/xml/ns/samples/extension-3",
9       "ssn": "1234567890"
10    },
11    "metadata": {
12      "requestTimestamp": "2017-12-15T09:40:35.277Z",
13      "requestorRef": {
14        "oid": "00000000-0000-0000-0000-000000000002",
15        "relation": "http://midpoint.evolveum.com/xml/ns/public/common/org-3#default",
16        "type": "http://midpoint.evolveum.com/xml/ns/public/common/common-3#UserType"
17      },
18      "createTimestamp": "2017-12-15T09:40:37.402Z",
19      "creatorRef": {
20        "oid": "00000000-0000-0000-0000-000000000002",
21        "relation": "http://midpoint.evolveum.com/xml/ns/public/common/org-3#default",
22        "type": "http://midpoint.evolveum.com/xml/ns/public/common/common-3#UserType"
23      },
24      "createChannel": "http://midpoint.evolveum.com/xml/ns/public/common/org-3#default",
25      "modifyTimestamp": "2017-12-15T09:40:37.402Z",
26      "modifierRef": {
27        "oid": "00000000-0000-0000-0000-000000000002",
28        "relation": "http://midpoint.evolveum.com/xml/ns/public/common/org-3#default",
29        "type": "http://midpoint.evolveum.com/xml/ns/public/common/common-3#UserType"
30      }
31    }
32  }
33 }
```

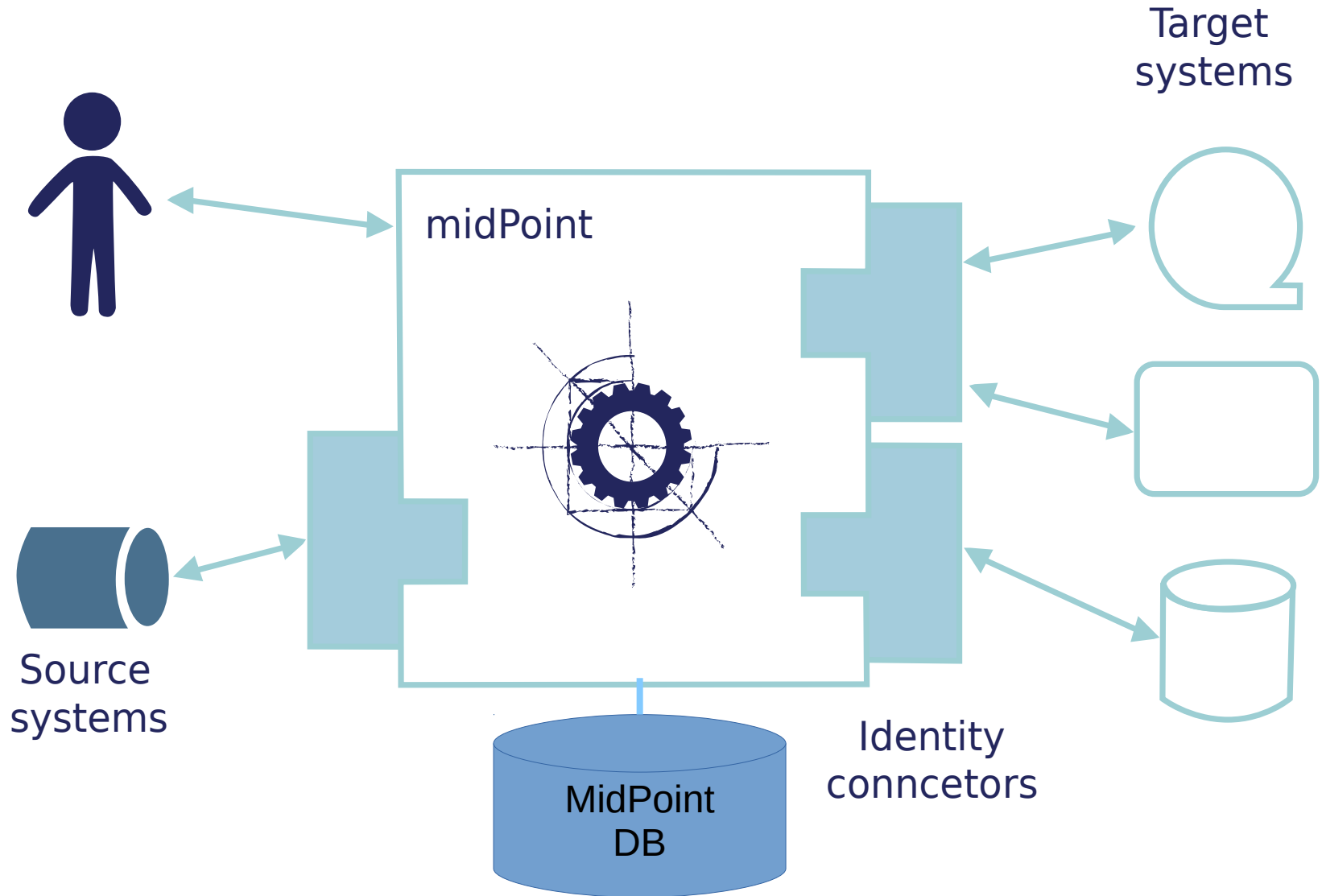
Bottom Screenshot (YAML): Shows the JSON converted to YAML:

```
1 ---
2 '@ns': "http://midpoint.evolveum.com/xml/ns/public/common/common-3"
3 user:
4   oid: "fc41bc2d-6f81-45f3-a103-6e05741fb882"
5   version: "26"
6   name: "foo"
7   extension:
8     '@ns': "http://midpoint.evolveum.com/xml/ns/samples/extension-3"
9     ssn: "1234567890"
10  metadata:
11    requestTimestamp: "2017-12-15T09:40:35.277Z"
12    requestorRef:
13      oid: "00000000-0000-0000-0000-000000000002"
14      relation: "http://midpoint.evolveum.com/xml/ns/public/common/org-3#default"
15      type: "http://midpoint.evolveum.com/xml/ns/public/common/common-3#UserType"
16    createTimestamp: "2017-12-15T09:40:37.402Z"
17    creatorRef:
18      oid: "00000000-0000-0000-0000-000000000002"
19      relation: "http://midpoint.evolveum.com/xml/ns/public/common/org-3#default"
20      type: "http://midpoint.evolveum.com/xml/ns/public/common/common-3#UserType"
21  
```

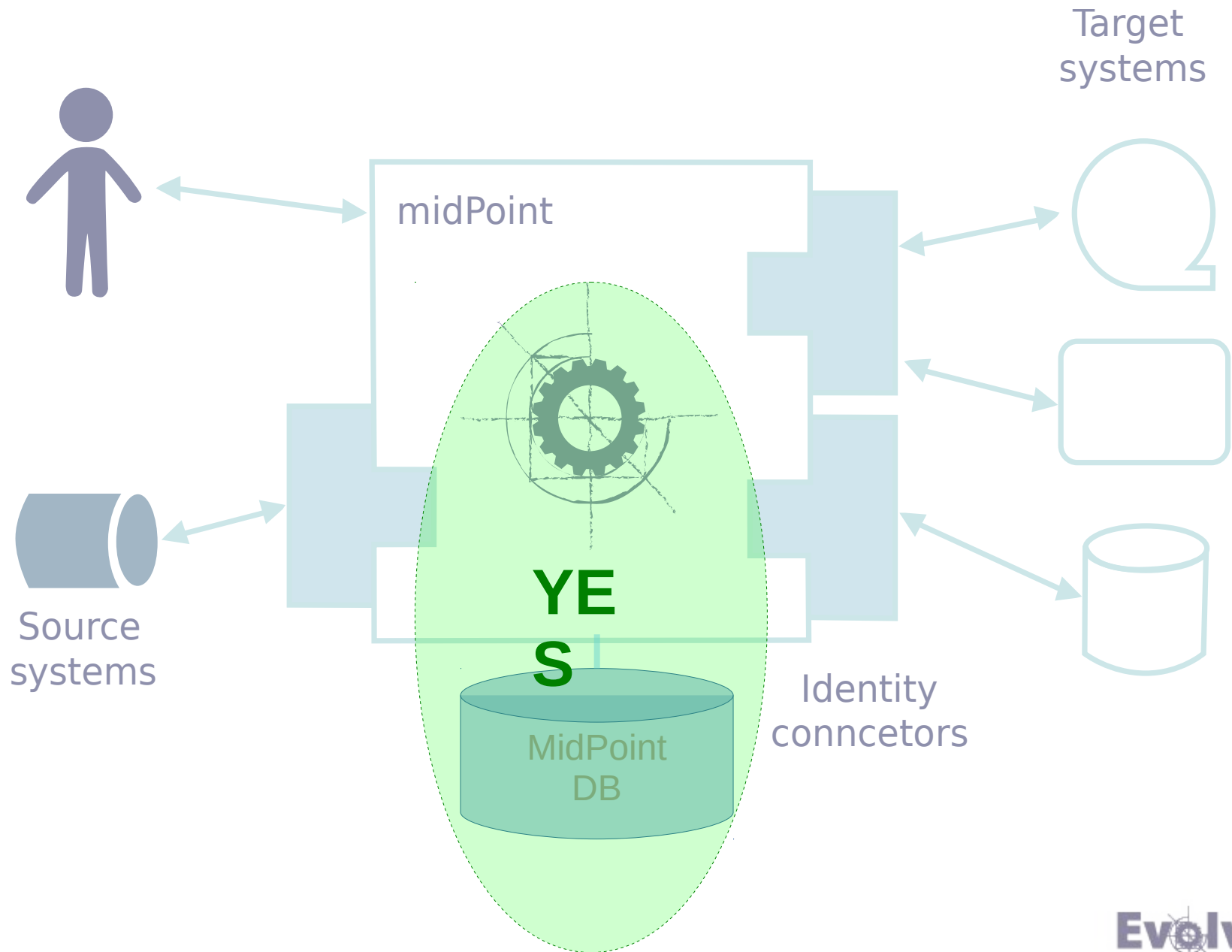
Prism Deltas



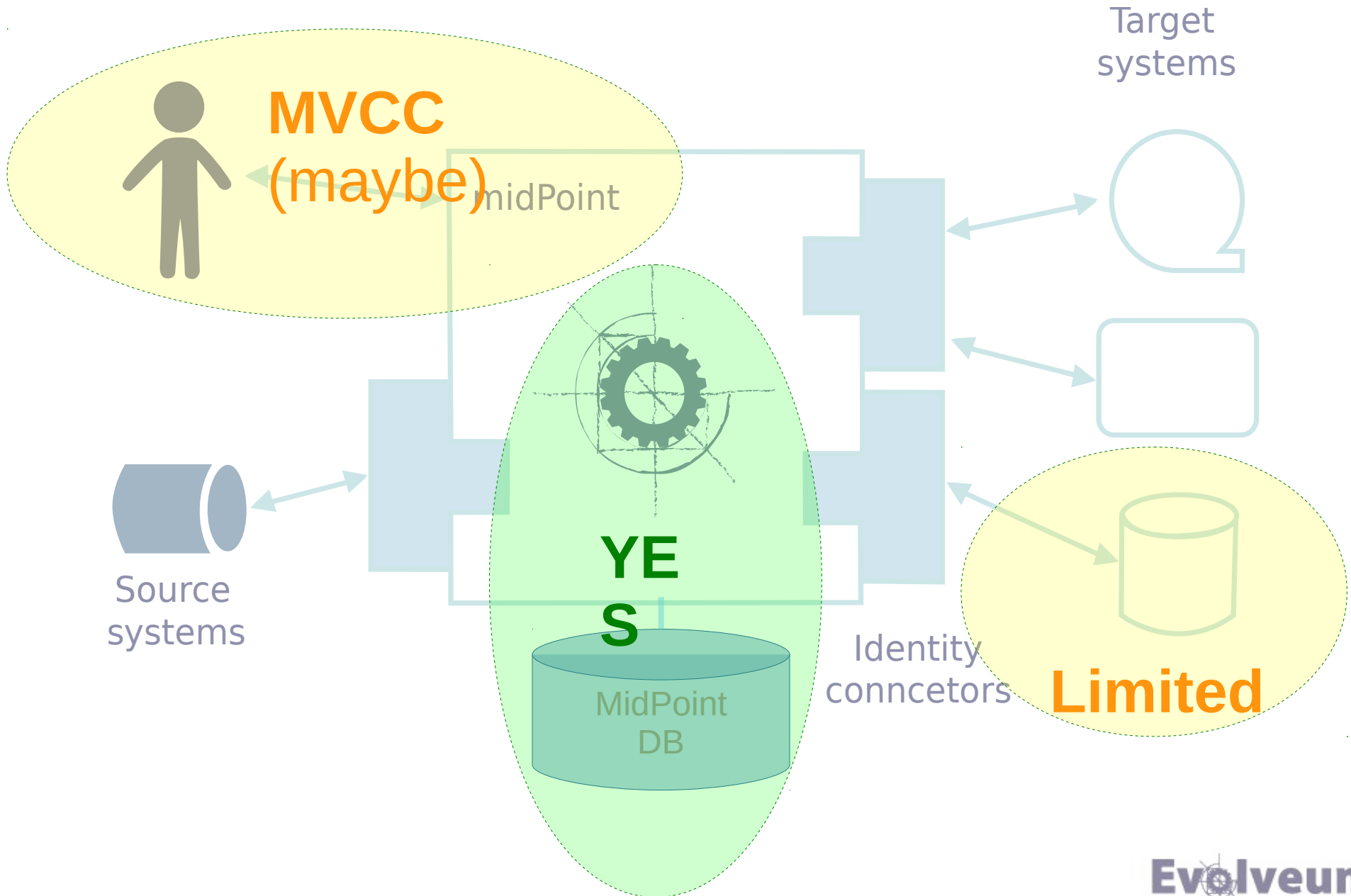
Big Problem of Consistency



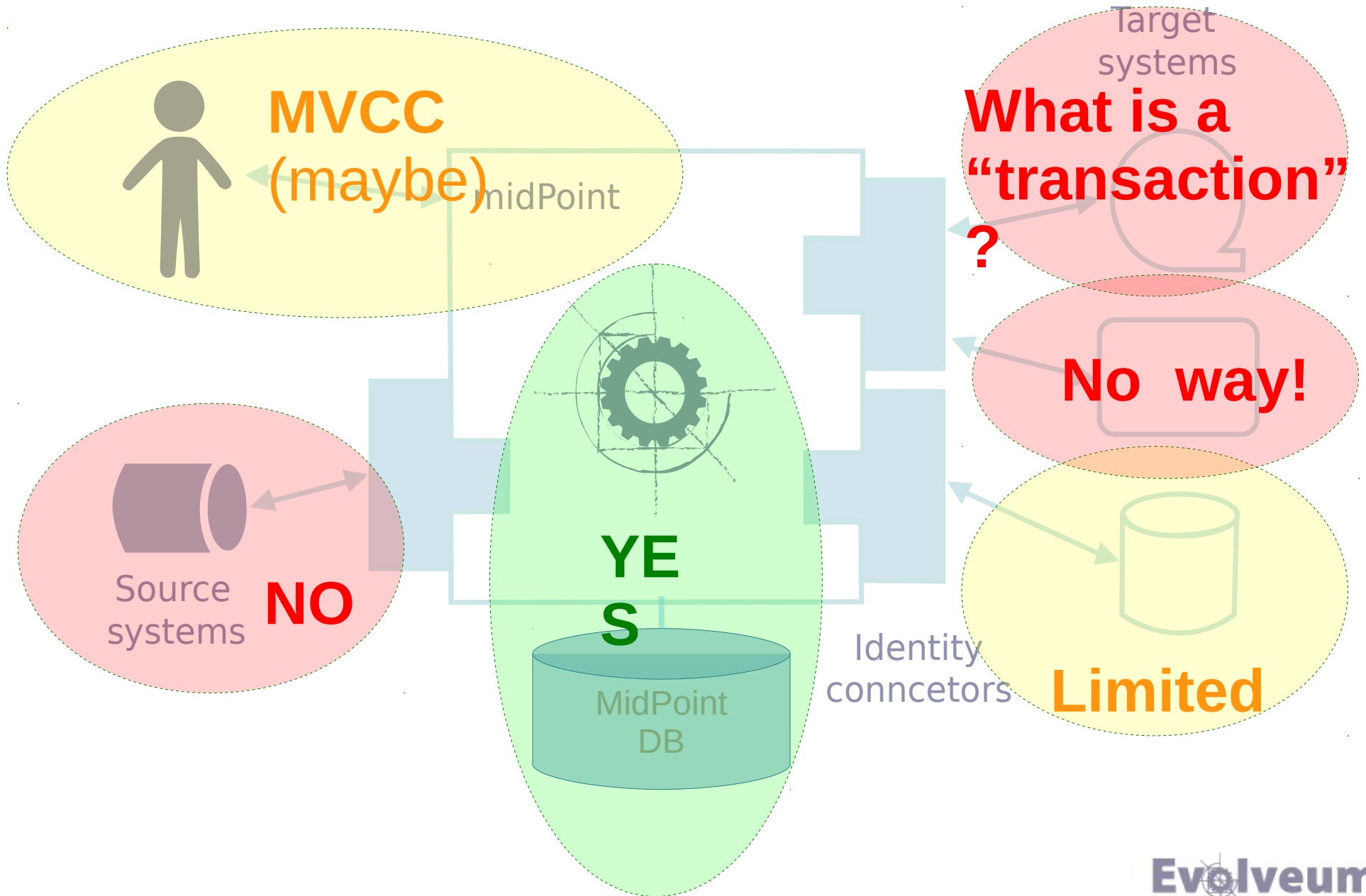
Transactions / MVCC



Transactions / MVCC



Transactions / MVCC



“Relativistic” Consistency

- Deltas are usually relative (*add, delete*)
- Apply delta in any order => equivalent value
- We need unordered multi-values for that
... but ordering is seldom needed
- There are still some weak spots (e.g. *replace*)
- But conflicts are quite unlikely
- “Reconciliation” as safety net

Heureka! It works!

Prism : Much More

- Static schema (compile-time)
- Dynamic schema (run-time)
- “Superdynamic” schema
- Raw data
 - We do not have complete schema at parse-time
- Deltas (schema-aware)
- Search filters (schema-aware, of course)
- Lifecycle (versioning, deprecated, experimental)

Questions You Surely Want To Ask

- Why XSD?
 - Because midPoint started in 2011
 - Because JSON Schema and others are equally bad
- Namespaces? QNames?
 - Yes, we use them (even in JSON and YAML)
 - No, we are not crazy (yet)
 - End user (usually) does not need to deal with them
 - QName == URI
 - Benefits: extensibility, versioning

RESTful API

RPC REST
(almost)

`http://.../rest/users`

`http://.../rest/users/02c15378-c48b-11e7-b010-1ff8606bae23`

`http://.../rest/tasks/c68d7770-...-9bec1fc3b57c/suspend`

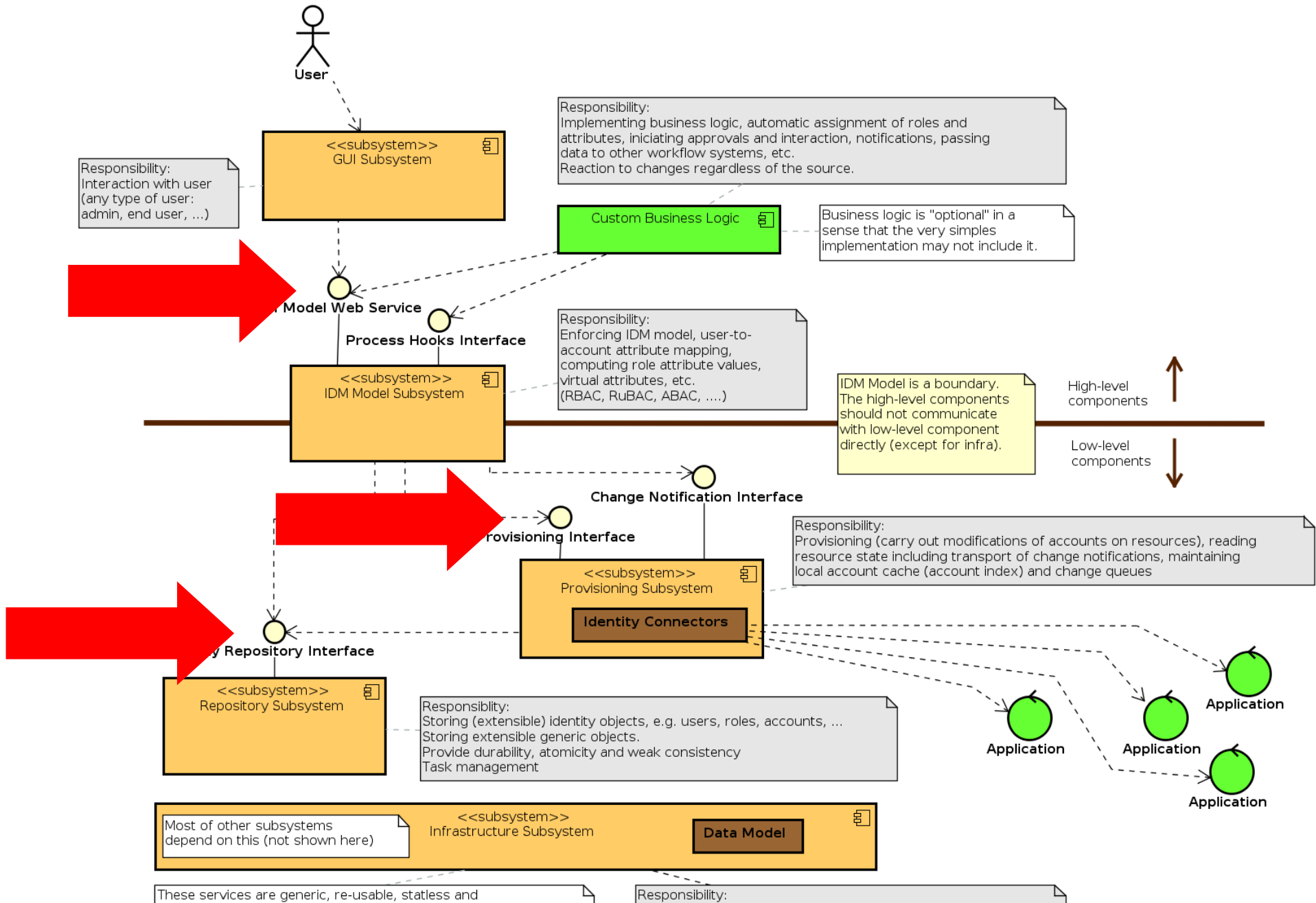
`http://.../rest/notifyChange`

- “REST” part and RPC part (and some overlap)
- Full schema support: XML, JSON, YAML
- Big problem of REST: modifications
... but we do not worry, we have deltas
- SOAP to REST in five easy steps

Testing

- Automated **integration testing**
 - Thousands of test cases
 - Still based on unit test framework (TestNG)
 - Embed what you can (DB, LDAP server, ...)
- Not that much **unit tests**
 - Are you crazy? Yes ... I mean: No!
 - Remember: code generated from schema + compiler
 - Unit test maintenance is very expensive
- End-to-end tests – in progress
- Test-Driven Bugfixing (TDB)

Designed For (Integration) Testability



Rolling-Wave Approach

2018	2019		2020		2021
v3.9 exact plan	v4.0 rough plan	v4.1 some plan	v4.2 maybe	v4.3 probably	??? v5.0 here or maybe not

2018	2019		2020		2021
v3.9 done	v4.0 exact plan	v4.1 rough plan	v4.2 some plan	v4.3 most likely	v5.0 here maybe

2018	2019		2020		2021
v3.9 done	v4.0 done	v4.1 exact plan	v4.2 rough plan	v4.3 some plan	v4.4 maybe v5.0 probably


Rolling-Wave Approach

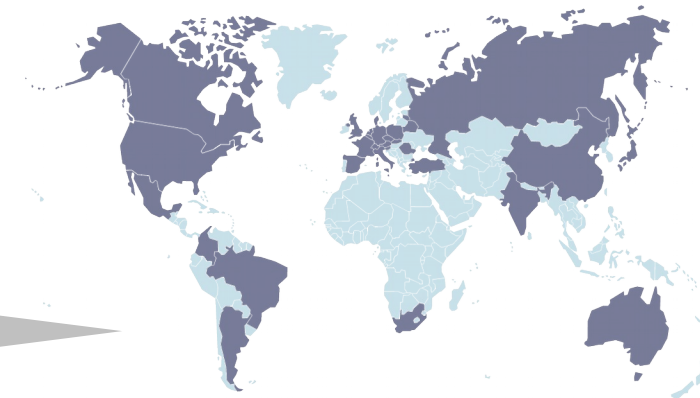
- Rolling-wave planning: obvious and intuitive
- Rolling-wave approach applied to everything:
 - architecture, schema, features, release scope
- Create architecture that can survive decades
 - But do NOT implement everything
 - Implement only what you need now
- Design 1-3 years ahead
 - But do NOT implement what you don't need now
 - Data model (schema), DB model, interfaces
- Implement only what you need

Questions you wanted to ask at the beginning

- Java? Really?
 - Really. And we use checked exceptions!
 - But no Java EE. We are not that crazy.
 - Compiler saves huge amount of time (remember: code generated from data model)
 - Old language +1: libraries for everything
 - Old language -1: you need to avoid landmines
 - OpenJDK
 - Hindsight: Java is lesser evil

Questions you wanted to ask at the beginning

- Self-funded? And still alive?
 - Alive and well. 
 - Bootstrapped (FFF). No venture capital.
 - Beginnings were hard. Very hard.
 - Persistence pays off.
- Business model?
 - Subscription: support + new feature development
 - Trainings, PoCs, Architecture reviews
 - Professional services, projects (minimal) → partners



Summary

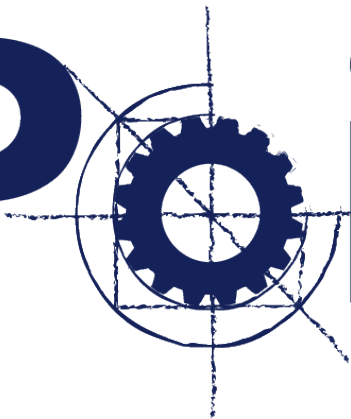
midPoint

- Million lines of Java code, 7 years, small team ... and still going fast and strong.
- Open source, self-funded ... and survived!
- Good architecture, rolling-wave design
- Schema-aware from bottom to top
- Not entirely normal project

Join the Team

- Java developers, IDM engineers, ... marketing
- Bratislava, Košice
... or anywhere (remote work)
- Join the team
... if you are up to the challenge

midPoint



Conditions REST Metaroles Lifecycle Extensibility Workflow Template
Connectors Matching rules Caching Parametric roles **Policy rules**
Role catalog **Identity Management** Schema Expressions
Correlation **Synchronization** Organizational Structure Entitlements
Localization Validity constraints
Scripting **Self-service Governance** RBAC LDAP Consistency
Sequences Approval Import SoD **Data Protection** LiveSync
Reporting Notifications Constants
Mappings XML/JSON/YAML Recertification Function libraries Personas
Audit Reconciliation ITSM integration **Authorization** Meta-data
Manual provisioning Deputy
Password management Bulk actions Dependencies Administration Web UI

For more information please visit
www.evolveum.com