

# D2R Server

Publishing databases on the Web  
as SPARQL endpoints

Richard Cyganiak, Chris Bizer

# The problem

- Have data in a database
- Want to access it from the RDF world

# 2002: No good tools

- little or no customization
- auto-generated schemas

# D2R Map (2002): customizable

# D2R Map

- dumps database to RDF file
- you load it into RDF triple store
- problem: RDF data becomes stale

# D2RQ (2004): Live queries

# D2RQ

- access database through Jena API
- rewrite queries on the fly into SQL
- problem: limited to Jena-based clients
- problem: remote access?

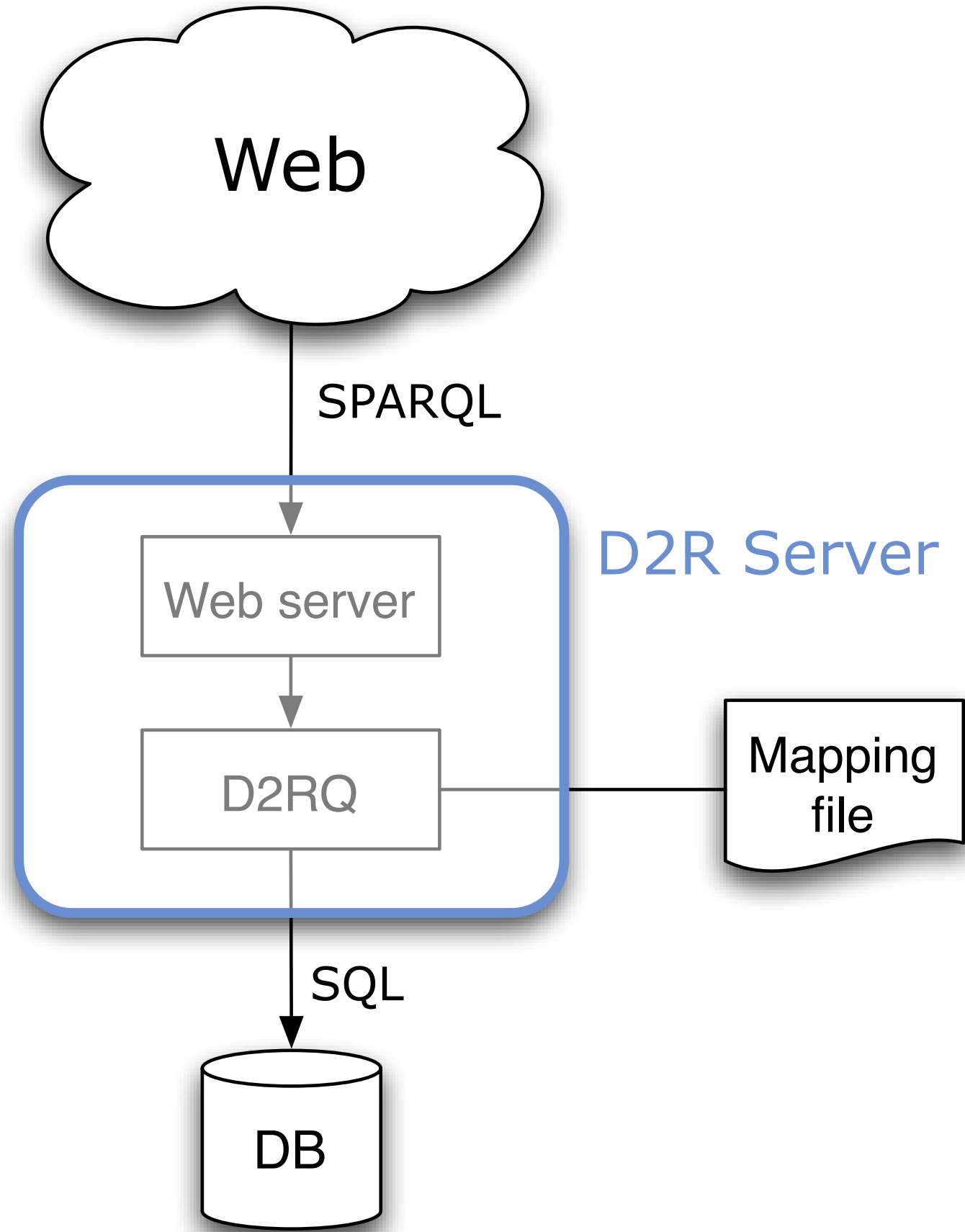
# SPARQL



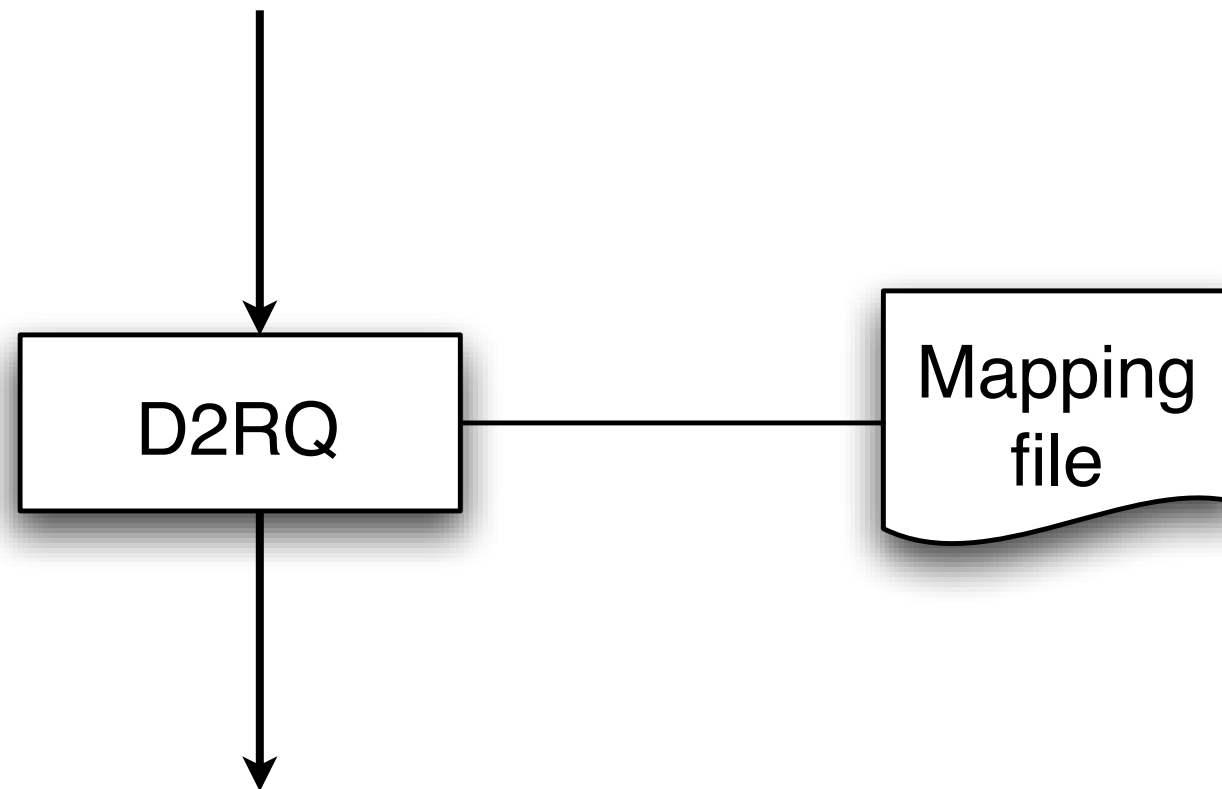
# D2R Server (2006): Over the Web

# D2R Server

- SPARQL protocol server based on D2RQ
- Turns a database into “SPARQL endpoint”



```
SELECT * WHERE {  
  ?x foaf:name "Chris" .  
  ?x foaf:mbox ?mbox .  
}
```



```
SELECT People.ID, People.email  
FROM People  
WHERE People.name="Chris";
```

# D2RQ mapping language

# ClassMaps

## People

<u>ID</u>	name	email	orgID

```
:PeopleClassMap a d2rq:ClassMap;  
  d2rq:class foaf:Person;  
  d2rq:uriPattern  
    "http://example/person@@People.ID@@".
```

# PropertyBridges

## People

<u>ID</u>	name	email	orgID

```
:PeopleEmailColumn a d2rq:PropertyBridge;  
  d2rq:belongsToClassMap :PeopleClass;  
  d2rq:property foaf:mbox;  
  d2rq:uriPattern "mailto:@@People.email@@".
```

# Joins

## People

<u>ID</u>	name	email	orgID

```
:PeopleOrgColumn a d2rq:PropertyBridge;  
  d2rq:belongsToClassMap :PeopleClass;  
  d2rq:property ex:worksFor;  
  d2rq:refersToClassMap :OrganizationsClass;  
  d2rq:join "People.orgID=Organizations.ID".
```



# ... and more

- can map messy database schema to clean RDF

# Making it easy

# Mapping generator

- Auto-generates mapping file from database schema
- Then customize manually
- Less boilerplate typing

# SPARQL explorer

- Browse mapped RDF
- Instant feedback from incremental changes to mapping file

SPARQL Explorer - D2R Server at http://localhost:2020/sparql

http://localhost:2020/ Google

## D2R Server at http://localhost:2020/sparql

**SPARQL:**

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>
PREFIX db: <jdbc:mysql://127.0.0.1/wordpress#>

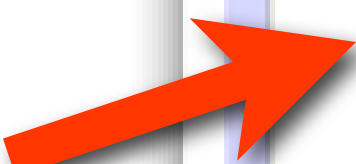
SELECT * WHERE {
...
}
```

**Browse:**

- [Classes](#)
- [Properties](#)

Results:

Powered by [D2R Server](#)



## D2R Server at http://localhost:2020/sparql

### SPARQL:

```
PREFIX rdf: <http://www.w3.org/1999/02/22-rdf-syntax-ns#>  
PREFIX xsd: <http://www.w3.org/2001/XMLSchema#>  
PREFIX db: <jdbc:mysql://127.0.0.1/wordpress#>
```

```
SELECT ?property ?hasValue ?isValueOf  
WHERE {  
  { <jdbc:mysql://127.0.0.1/wordpress#users1> ?property ?hasValue }  
  UNION  
  { ?isValueOf ?property <jdbc:mysql://127.0.0.1/wordpress#users1> }  
}  
ORDER BY (IBOUND(?hasValue)) ?property ?hasValue ?isValueOf
```

### Browse:

- [Classes](#)
- [Properties](#)

Results:

### Description of jdbc:mysql://127.0.0.1/wordpress#users1:

property	hasValue	isValueOf
<a href="#">rdf:type</a>	<a href="#">db:users</a>	-
<a href="#">db:users_ID</a>	1	-
<a href="#">db:users_display_name</a>	"Richard Cyganiak"	-
<a href="#">db:users_user_activation_key</a>	""	-
<a href="#">db:users_user_email</a>	"richard@cyganiak.de"	-
<a href="#">db:users_user_login</a>	"admin"	-
<a href="#">db:users_user_nicename</a>	"site-admin"	-
<a href="#">db:users_user_pass</a>	"83afbb3c2fddb610ad445f4bdcf2f119"	-
<a href="#">db:users_user_status</a>	0	-
<a href="#">db:users_user_url</a>	"http://richard.cyganiak.de"	-

Powered by [D2R Server](#)



# SPARQL explorer

- Web 2.0 buzzword compliant
- AJAX
- SPARQL results over JSON

# Summary

- Access database as SPARQL endpoint
- Live queries
- Easy setup
- Flexible, customizable mapping
- Download, docs and source code:
  - <http://www.wiwiss.fu-berlin.de/suhl/bizer/d2r-server>



Thank you!