



**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP  
BERLIN, 26-28 SEPTEMBER 2012

**APACHE SLING & RDBMS**

**Stefan Seifert**

## Stefan Seifert

Founder of pro!vision GmbH

### pro!vision GmbH

Wilmerdorfer Str. 50-51

10627 Berlin

<http://www.pro-vision.de>



**PRO!VISION**  
SOFTWARE CRAFTSMANSHIP



Enterprise Solution Partner  
**BUSINESS**

Spezialized on Adobe CQ and it's technology stack since 2003

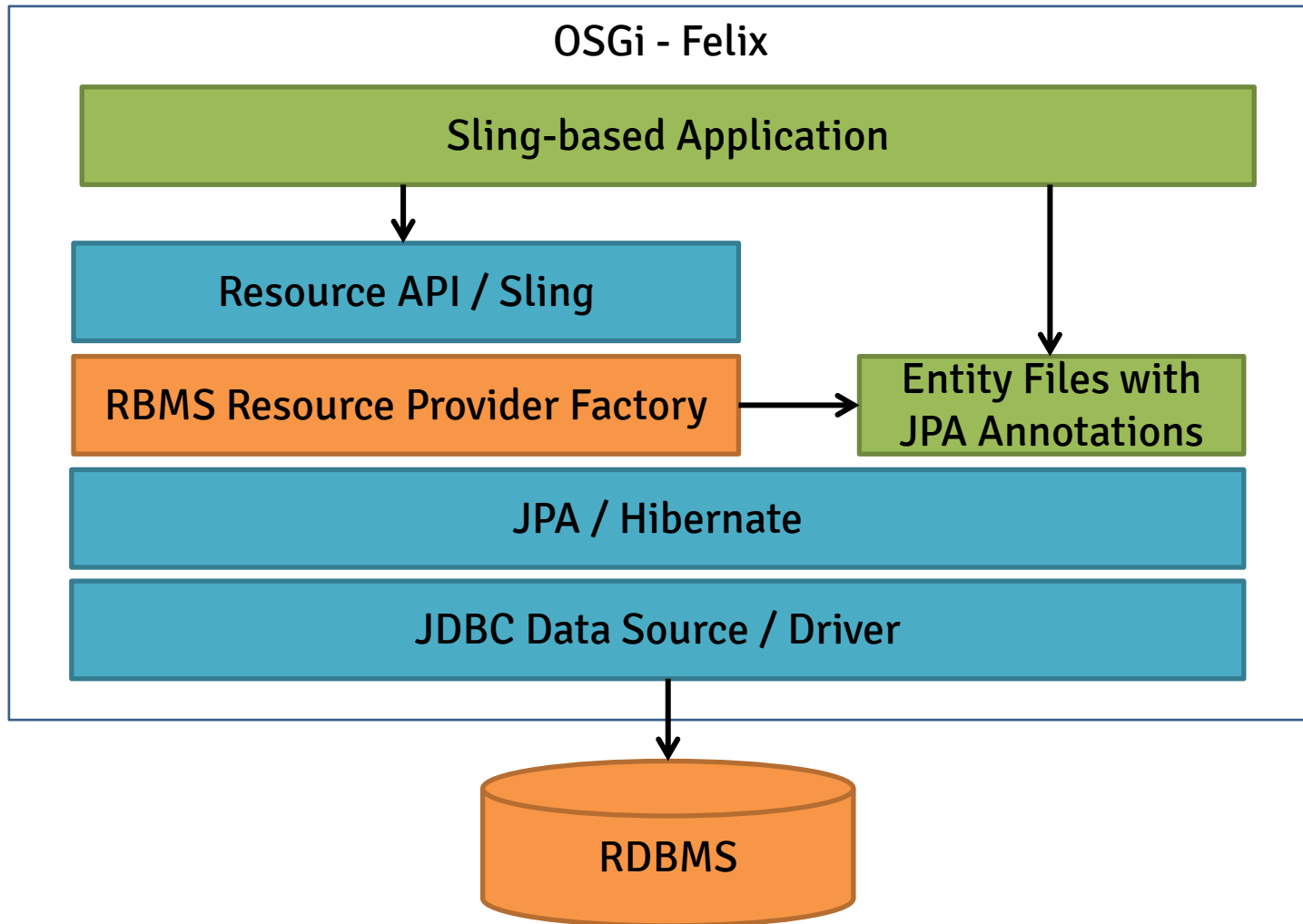
- **Goals of this presentation**
  - Evaluate integration of RDBMS data into Sling Resource API and mapping concepts
  - Evaluate integration of Hibernate into OSGi Context (Felix, Sling, CQ5)
  - Evaluate new Sling Resource API CRUD Features
  
- **Disclaimer: Code is not Production ready, not feature complete! It's a technical prototype.**

- Status of Hibernate and OSGi
  - Hibernate 4 was split up for better OSGi compatibility
  - But ([cite from Steve Ebersole](#), Jan 2012):  
*“... The thing to keep in mind is that no one on the Hibernate team is an OSGi expert. This is something that someone who is fluent in authoring OSGi metadata and has a vested interest should work on.”*
  - So there is still some work to do...

- **Hibernate and OSGi**
  - ...fortunately someone had done the hard part already:
    - [Everit OSGi Tools](#)
    - [Post from Balázs Zsoldos on Aries Mailing List](#)
  - But this does not work out-of-the-box in Sling/Felix environment
  - And adds much more dependencies required for our use case

- **Hibernate in the Sample Project**
  - Stripped down and adapted version of the Everit OSGi Hibernate Adapter with a subset of dependencies
  - Works in Sling Standalone and CQ5.5
  - Still has some issues (reports some warnings/errors in logs), not production ready
  - If you want to know more how to get non OSGI-compatible Libraries to work in an OSGI-Environment see my talk from last year [“Apache CXF and Sling”](#) (in short: it’s an ungrateful task)

# Sling & RDBMS – Integration overview



- **RdbmsResourceProviderFactory**
  - OSGi Factory Configuration
  - One Instance per Entity Class and Content Path
  - Maps the table content to a content path's children
  - Creates ResourceProvider instances on a per-use basis
  - For each request a JPA EntityManager instance is created and closed
  - JDBC Connections from JDBC Pool



- **RdbmsResourceProvider**
  - Implements `ModifyingResourceProvider`
  - The new API for CRUD inside Sling
  - Maps each table row to a resource
  - Provides a custom `ModifiableValueMap` that transparently reads/writes properties from the JPA entity instance
  - Maps `create/delete`, `commit/revert` to the underlying JPA API

- **RdbmsResourceProvider**
  - All data changes (updates, inserts, removes) are persisted in the Hibernate session, but not in the database
  - This happens in a bulk operation when `ResourceResolver.commit()` is called
  - If not `commit()` is executed all changes are rolled back when the request ends

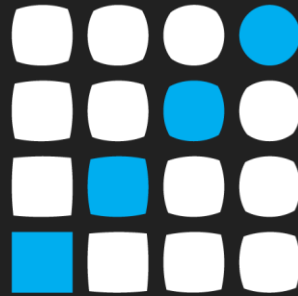
- **RdbmsResourceProvider**
  - Implements `QueryableResourceProvider`
  - Supports passing through of JPA QL Queries via Sling Resource API to Hibernate and iterating over the results.
  - **Issues:**
    - Distinction between multiple `RdbmsResourceProviderFactory` instances (different JPA entities) not solved yet
    - Sling API allows only single string for Query  
→ High risk for SQL injection vulnerability

- Demo of Sling Standalone sample implementation
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
  
- There's a variant for CQ5.5 included as well
  - Beware: it will break your CQ5.5 instance due to incompatible Sling Snapshot bundles, but the Demo will work

- How to handle Master/Detail-Mappings, Join Object Attribute references
- Path mapping to a flat list best solution?
- hasChanges not implemented
- Query-Support only rudimentary
- Not working yet with CRUD-enabled SlingPostServlet (which is <24h old currently)
- Hibernate Integration not production-ready

# Conclusion

- Simple RDBMS Mapping works very well with Sling Resource API
- Especially the JCR-like concept of implicit transactions and transient changes can be directly mapped to Hibernate concepts
- Not yet proven if this suits well for complex use cases/database schemas
- But it's always possible to use the JPA API directly from a Sling Application's code



**adaptTo()**

APACHE SLING & FRIENDS TECH MEETUP  
BERLIN, 26-28 SEPTEMBER 2012

**Thank you for your attention!**

**Any questions?**