



DRIVOLUTION: RETHINKING THE DATABASE DRIVER LIFECYCLE

Emmanuel Cecchet (UMass Amherst)

Joint work with George Candea ()

THIS RESEARCH IS BASED ON TRUE EVENTS

UR

UNRESTRICTED

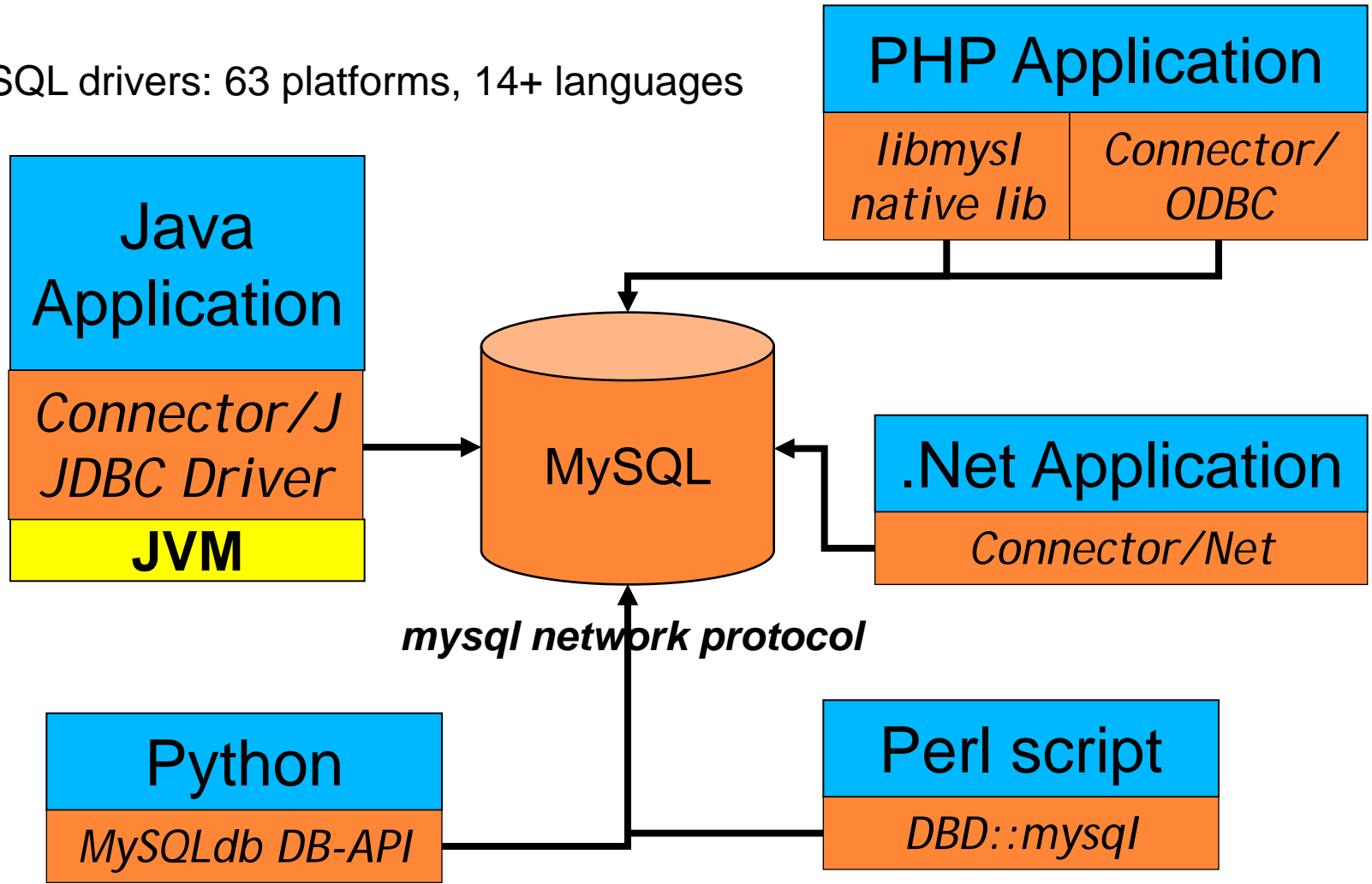


APPROPRIATE FOR BOTH ACADEMIC
AND INDUSTRIAL AUDIENCES



DATABASE DRIVERS

MySQL drivers: 63 platforms, 14+ languages



DATABASE DRIVER LIFECYCLE

- 1) Get an appropriate driver package from vendor
- 2) Install the driver on the client application machine
- 3) Configure the client application to use the driver
- 4) Start the application and load the database driver
- 5) Connect to database and check protocol compatibility
- 6) Authenticate
- 7) Execute requests

Driver update:

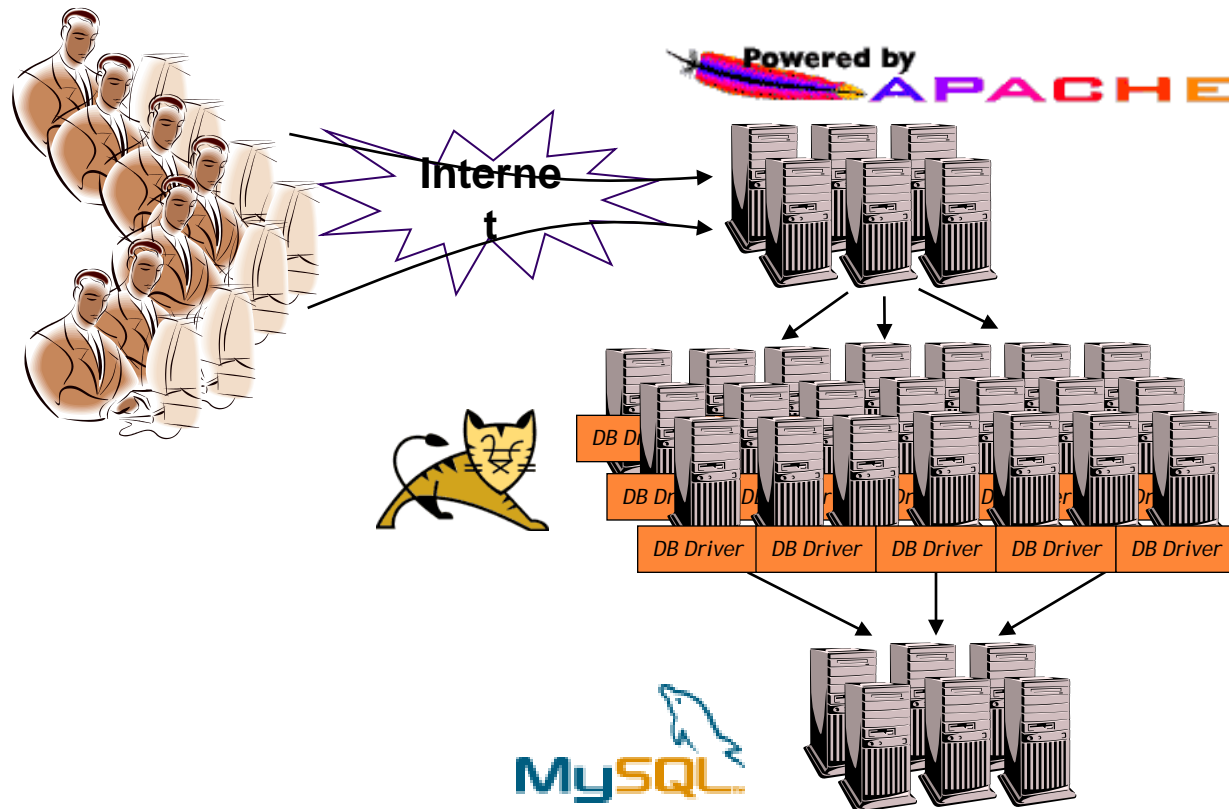
- 8) Stop the application
- 9) Uninstall old driver
- 10) Repeat all steps 1 through 7

STATE OF THE ART IN DATABASE DRIVERS

- Drivers are not sexy...
- Driver distribution separate from database engine
- Driver installation manual process on client machine
- Driver upgrades are disruptive (client application reconfiguration + restart)
- No protection against malicious drivers

A REAL LIFE EXAMPLE

- 200 application servers accessing a cluster of 4 DBs
- Driver upgrade more complex than database upgrade
- Online upgrades?



DRIVOLUTION

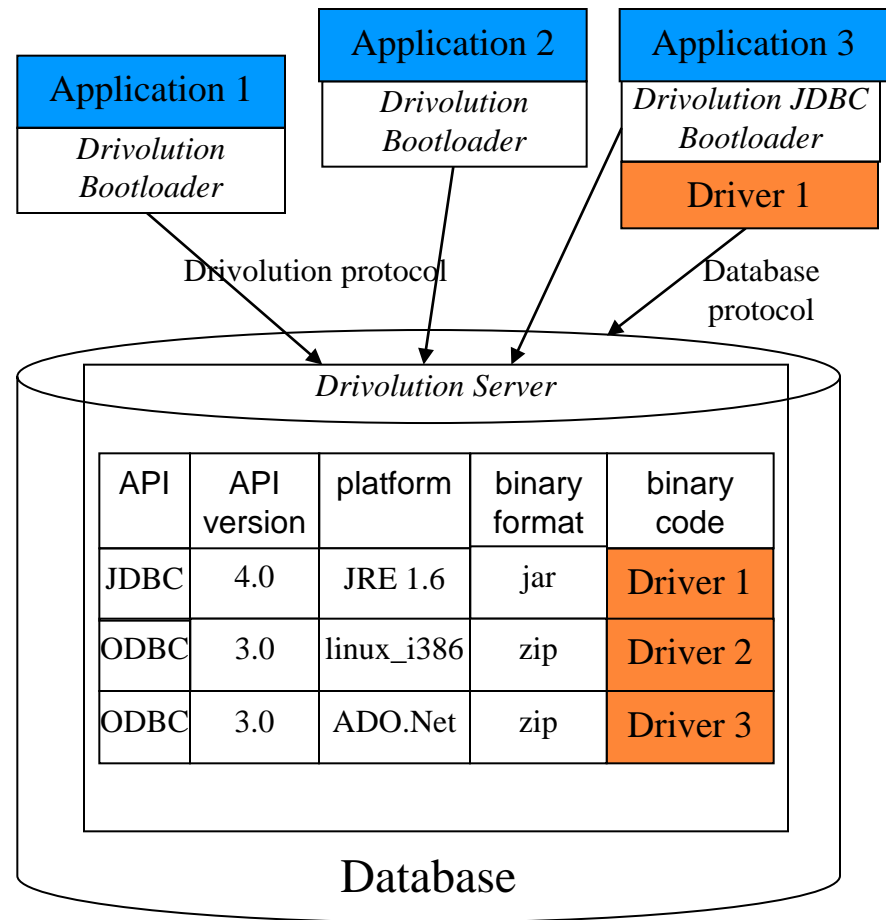
Think different

- Concepts
- Implementation
- Use cases

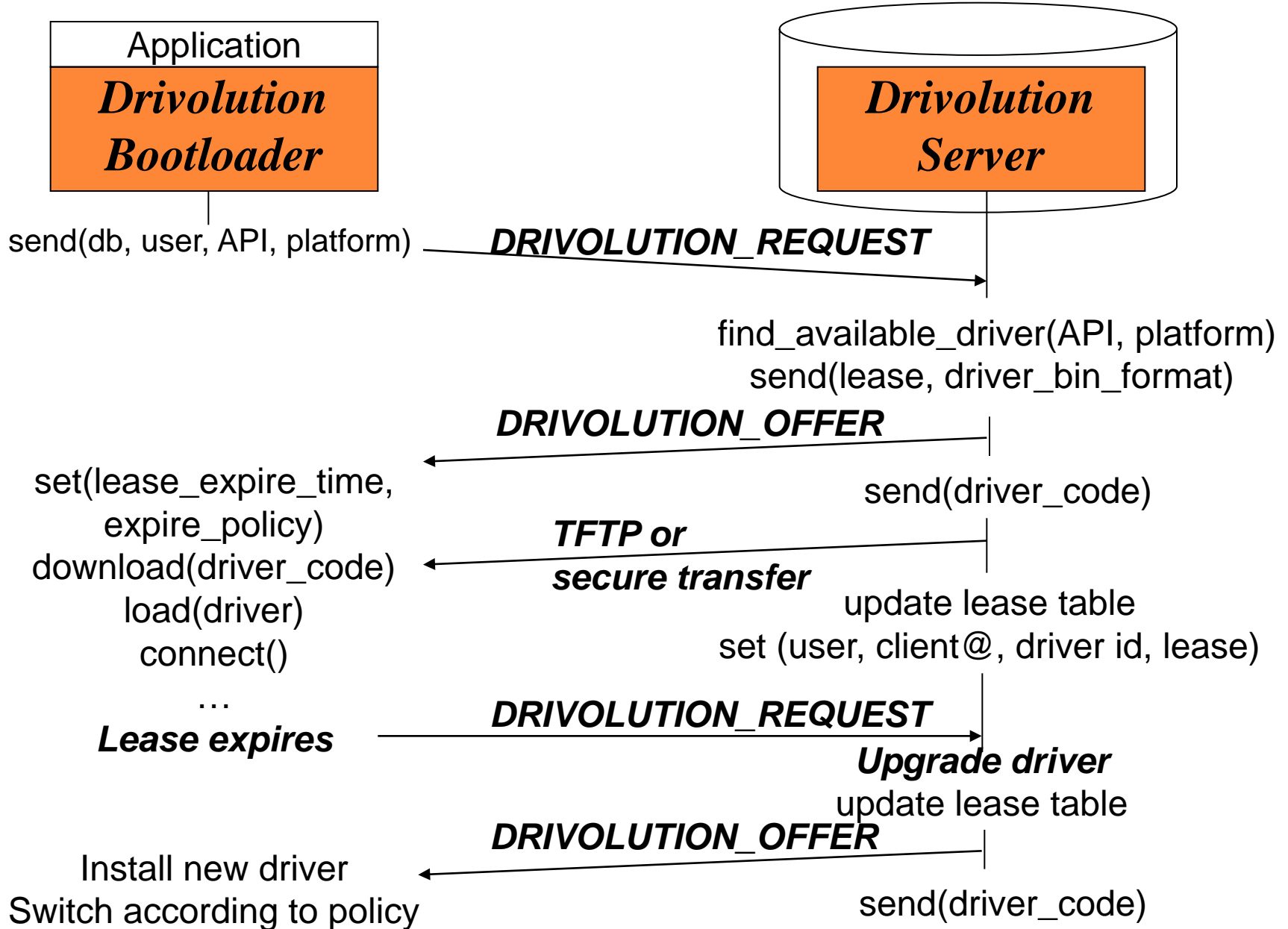


DRIVOLUTION CONCEPTS

- 2 Components: Bootloader and Server
- Drivers are stored in the database information schema
- Generic bootstrap receives driver a la DHCP from Drivolution server
- Lease associated to driver with various renewal policies



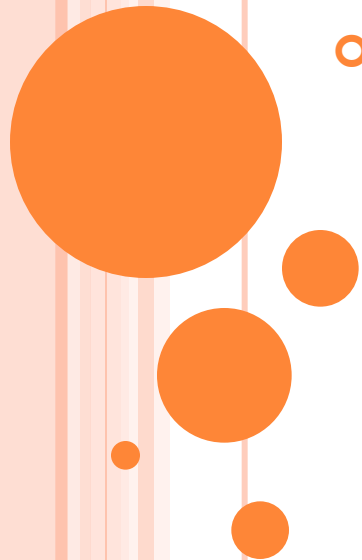
DRIVOLUTION CONCEPTS



DRIVOLUTION

Think different

- Concepts
- **Implementation**
- Use cases

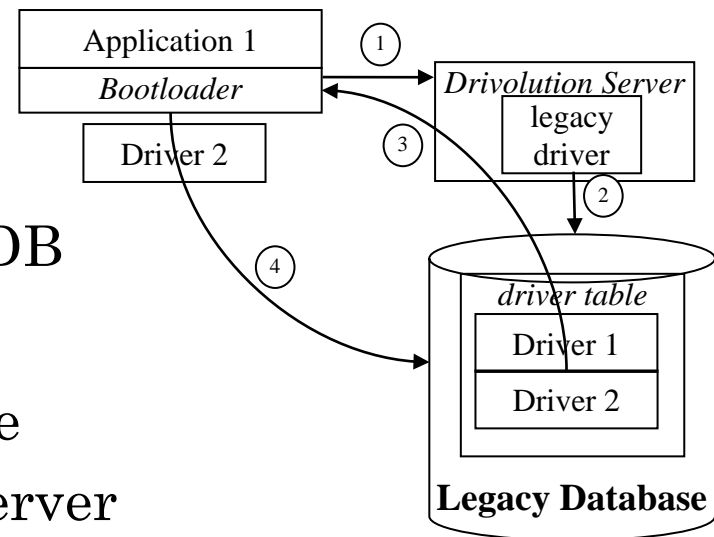


DRIVOLUTION BOOTLOADER

- Database driver agnostic but API[/Platform] specific
- Implements the Drivolution protocol
- Can load multiple drivers and switch from 1 version to the other
- Does not need to be upgraded

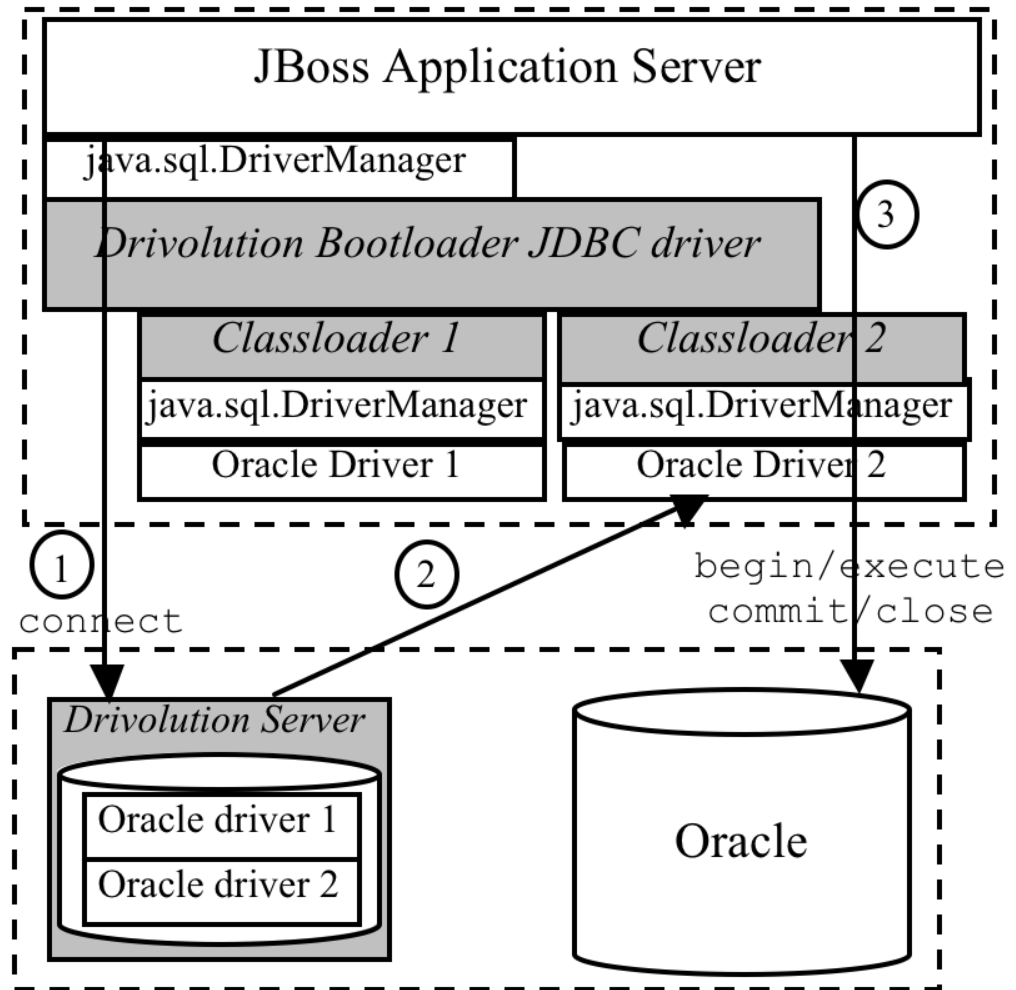
DRIVOLUTION SERVER

- Server logic uses regular SQL
 - re-uses all existing database mechanisms
 - logic can be implemented as a stored procedure
- In-core implementation
 - Drivolution server embedded in the database
 - can enforce connection closing on server side upon lease expiration
- External server
 - Server runs outside the DB
 - uses legacy driver to access DB
- Standalone server
 - database independent service
 - generic driver distribution server



DRIVOLUTION IN A JAVA WORLD

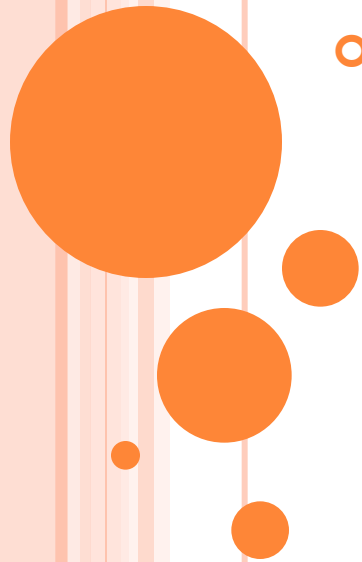
- Bootloader can be integrated in JVM runtime
- Very small footprint (8 KB)
- Drivers loaded in separate classloaders
- No runtime overhead after 1st connect (original driver code)



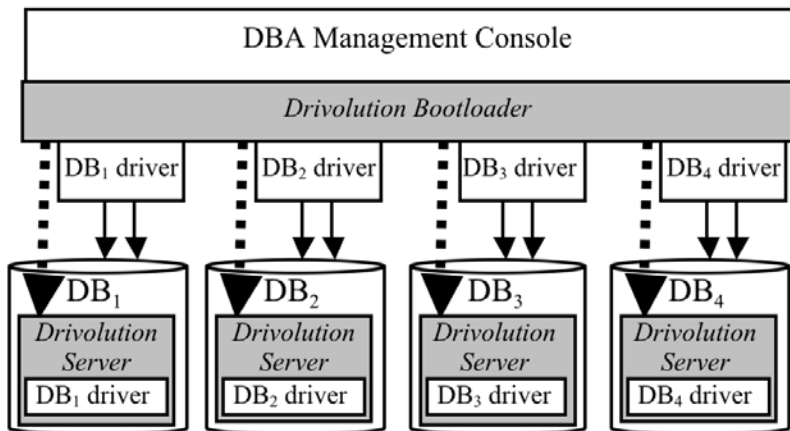
DRIVOLUTION

Think different

- Concepts
- Implementation
- Use cases



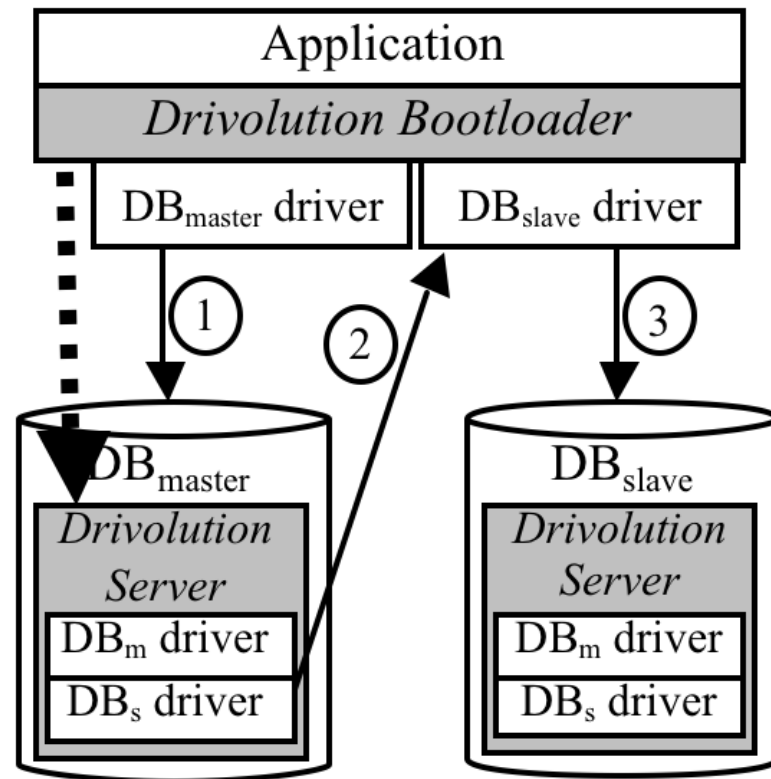
DRIVOLUTION USE CASES: HETEROGENEOUS DATABASE ADMIN



Tasks	Current State-of-the-Art	Drivolution
Accessing a new database	<ol style="list-style-type: none"> 1. Download drivers for DBA₁ platform 2. Configure DBA₁ console to find driver 3. DBA₁ connects to db 4. Download drivers for DBA₂ platform 5. Configure DBA₂ console to find driver 6. DBA₂ connects to db 	<ol style="list-style-type: none"> 1. DBA₁ connects to db 2. DBA₂ connects to db
Database driver upgrade	<ol style="list-style-type: none"> 1. Copy right driver for DBA₁ platform 2. Remove DBA₁ old driver 3. Restart DBA₁ console 4. Copy right driver for DBA₂ platform 5. Remove DBA₂ old driver 6. Restart DBA₂ console 	<ol style="list-style-type: none"> 1. Insert drivers in database 2. Revoke old driver

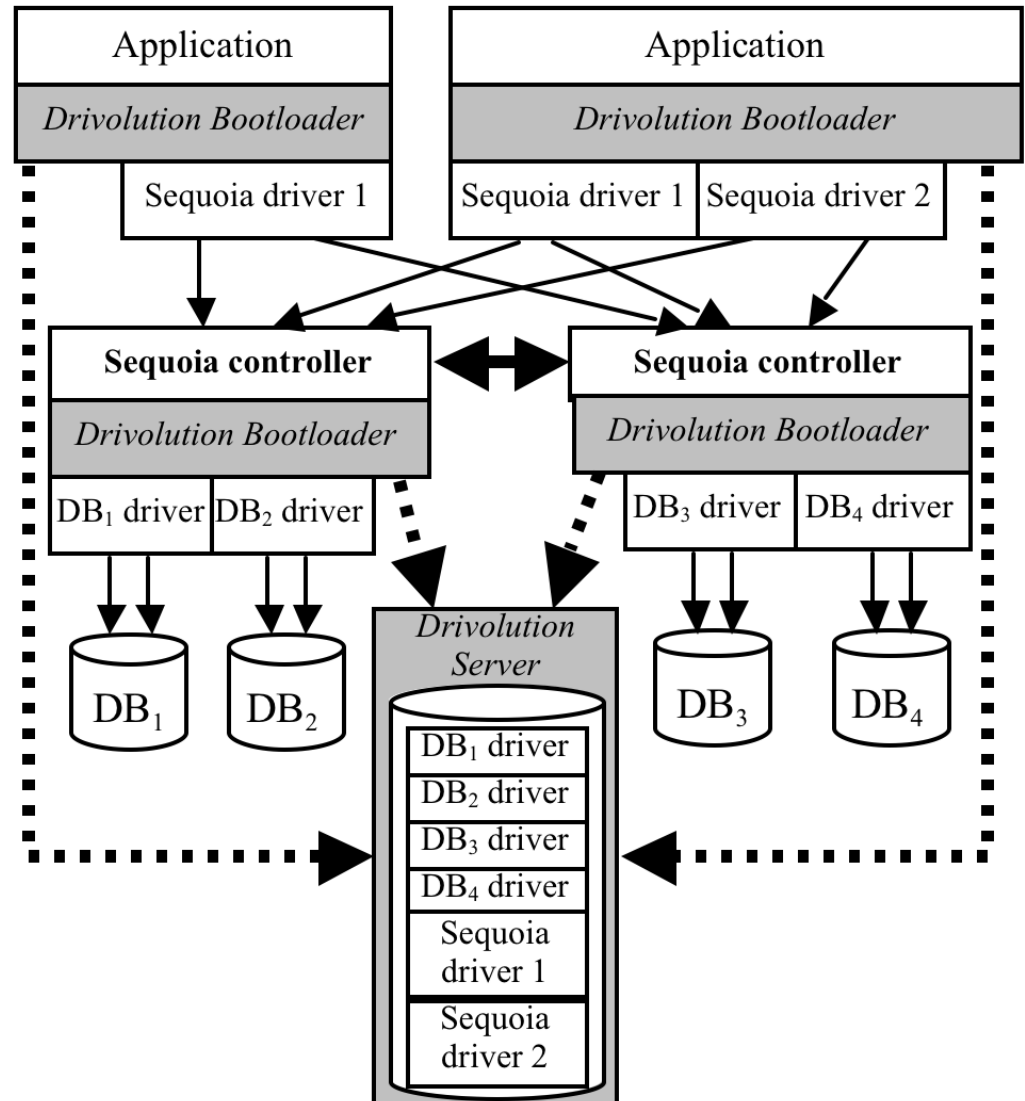
DRIVOLUTION USE CASES: DYNAMIC RECONFIGURATION OF CLIENTS

- Master/slave failover for planned maintenance
- 0-downtime operation
- Failback = downgrade



DRIVOLUTION USE CASES: DATABASE CLUSTERS

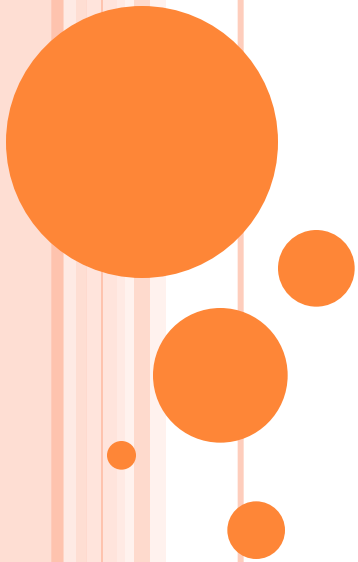
- Drivolution server as a standalone driver distribution service
- Useful for both legacy and homogeneous systems
- Can be replicated for HA



DRIVOLUTION: OTHER USE CASES

- Assembling drivers on-demand
 - extensions on-demand (NLS, GIS, Kerberos...)
 - driver code signing to identify malicious drivers
 - dynamic generation of preconfigured drivers
- License server
 - license keys in separate files
 - static or dynamic distribution
- Integration of Bootloaders in runtime libraries

**WHAT IS THE
MESSAGE TO
TAKE BACK
HOME?**



CONCLUSION

- Drivers are not sexy... but they deserve our attention

<https://sourceforge.net/projects/drivolution/>

- More open issues in Sigmod'08 paper
**MIDDLEWARE-BASED DATABASE REPLICATION:
THE GAPS BETWEEN THEORY AND PRACTICE**

QUESTIONS?



Contact: cecchet@cs.umass.edu