


Version Control for PL/SQL

About Gerger

Formspider

Application development framework for Oracle PL/SQL developers. First Class Web and Mobile Applications with only PL/SQL as the programming language. www.theformspider.com

Hundreds of customers in 30+ countries.

- 
- What is the problem?
 - How did we solve it?
 - Implementation Strategies
 - Demo!!

Success Story: (In other words, this really works. :-))

Rhenus Logistics, leading logistics service company from Germany, uses our solution.

Manages over 20,000 database objects

Database objects are spread over 30+ schemas

Continuous delivery implementation



If you implement version control for DB Development:

You'll be one of the pioneers.

Ready to tackle continuous delivery.

You'll deploy to production twice as often.

Errors during deployment will reduce by 95%.

Facts:

Version Control is a solved problem in file-based languages.

The version control systems are almost exclusively concerned with file-based programming languages.

Other developers have private development environments.

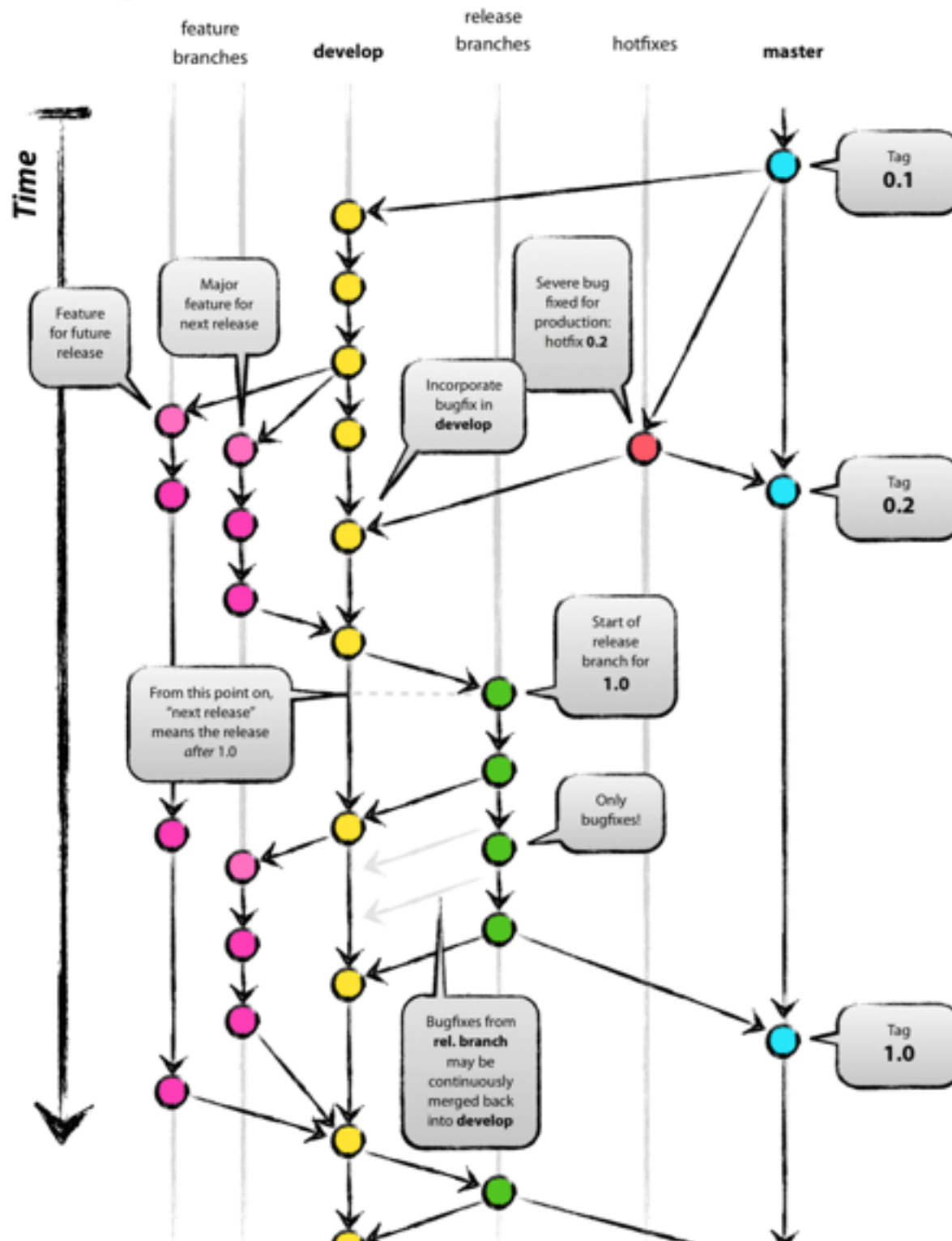
We need to find a way to bridge the database with version control systems.

Facts about Git:

Git is a distributed version control system (unlike SVN)

Best path forward

This is how best developers work with Git



Facts about database development:

Two types of objects:

CREATE

CREATE OR REPLACE

Code is 90% of the problem.

Constraints we live in:

Very little change to the workflow

DBA

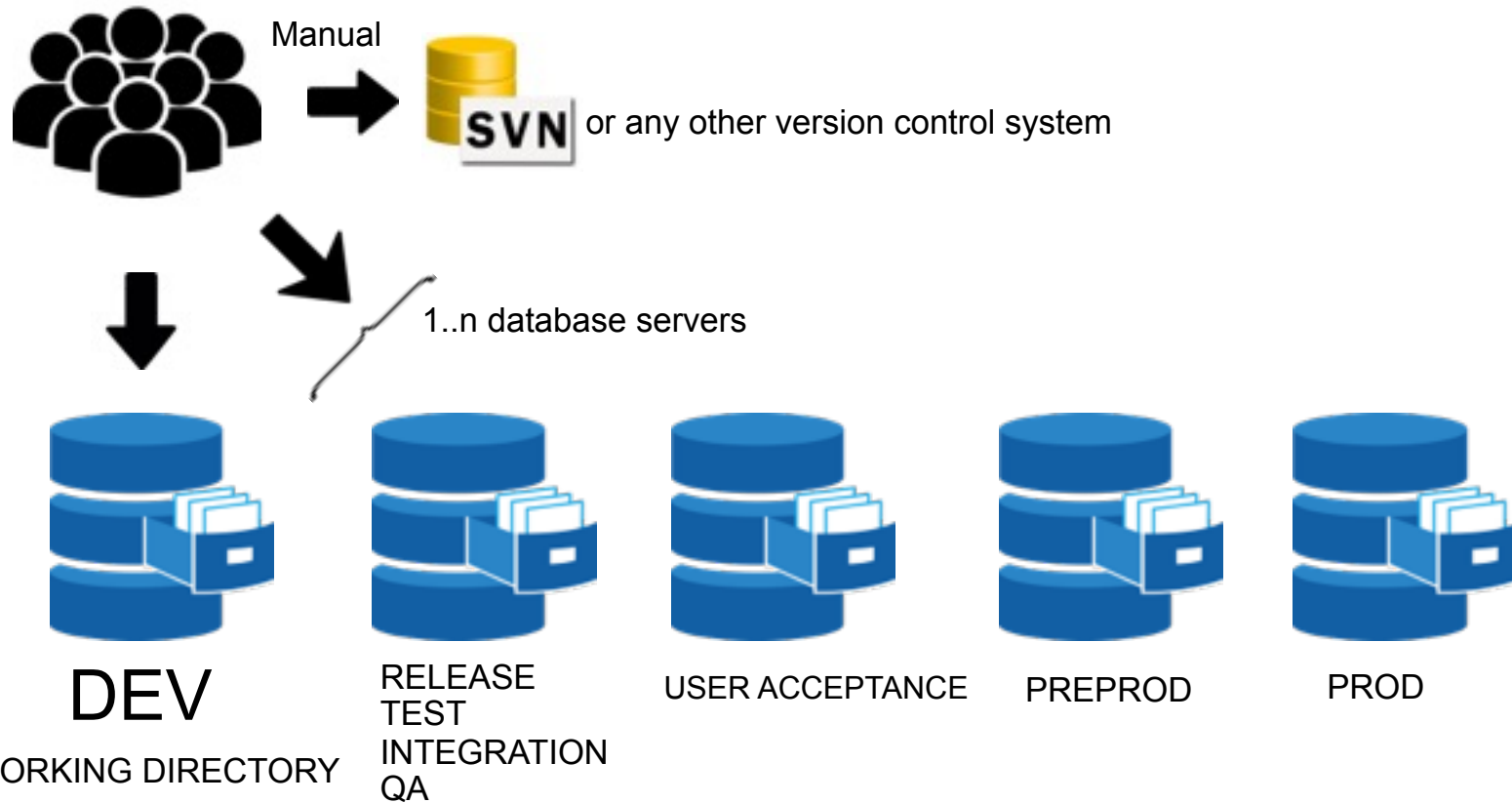
Cannot enforce a development tool

No private environment on day one

Minimal intrusion

We need to find a way to bridge the database to existing version control software.

Traditional PL/SQL Development



Queued access

Manual deployment

Version control repo is a last resort back up

Blocking

Intermingled projects, features, hot fixes

No automated testing

Interrupting

Practically impossible to branch, merge

Developers are expected to follow version management rules manually.

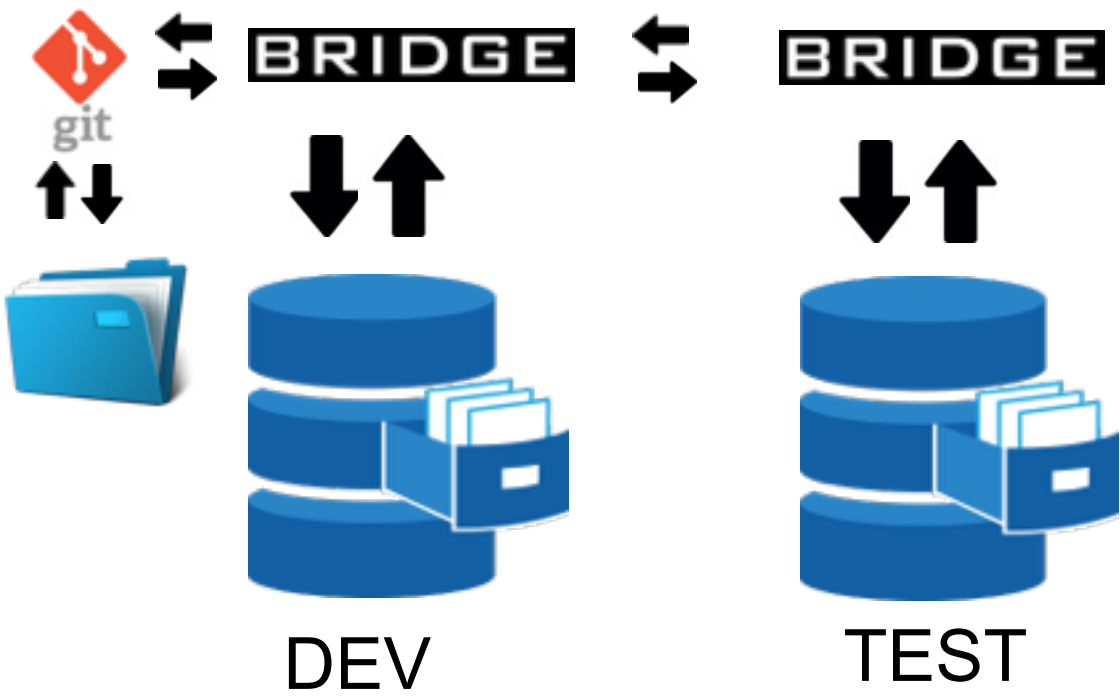
Create a Working Directory in the OS that is synced with the code in the Database.



Edit PL/SQL Objects as usual in your favorite editor.

Log in to your own bridge in the editor (can be done automatically in most cases).

Use bridge API's or bridge GUI to perform version control tasks.



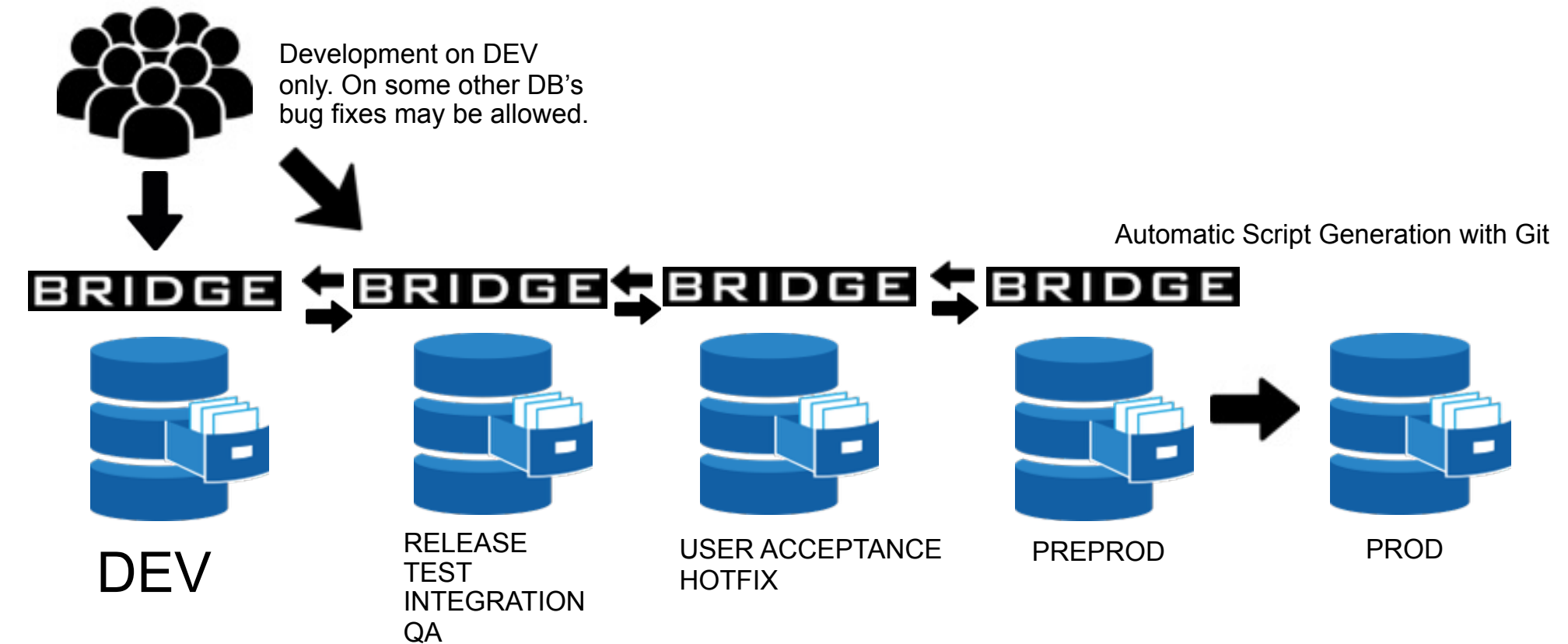
A database trigger listens to DDL events.

Version control commands update the database.

Bridge consists of a DB schema and some Java.

Only executes DDL's for changed objects. (Very fast!)

PL/SQL Development with Git Step 1 (One Repository)



Queued access

Automatic deployment

Source code is in Git

Blocking

Intermingled projects, features

Faster hot fixes

Better automatic testing

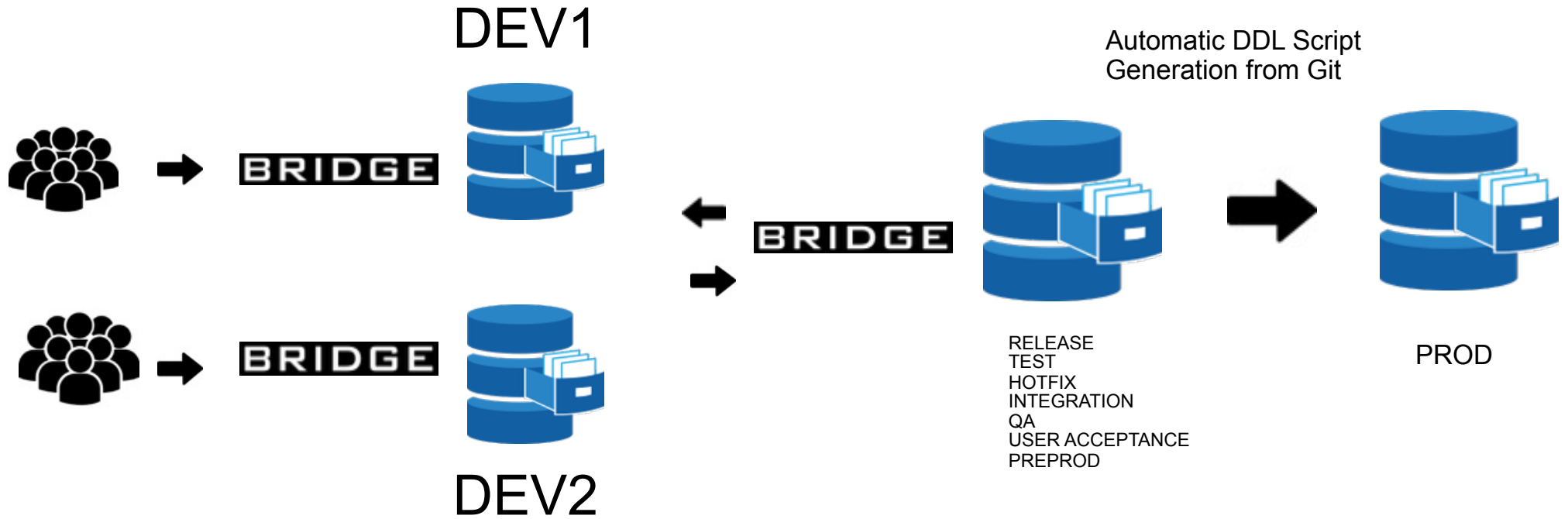
Interrupting

Basic branching and merging is possible for hot fixes

Version control rules are enforced at the database level.

PL/SQL Development Step 2 with Git (One Repository)

N DEV servers
DEV1, DEV2,
DEV3,...DEVN



Fewer Queues

Automatic deployment

Source code is in Git

Less Blocking

Less intermingled projects, features

Hot fixes delivered faster

Less interruption

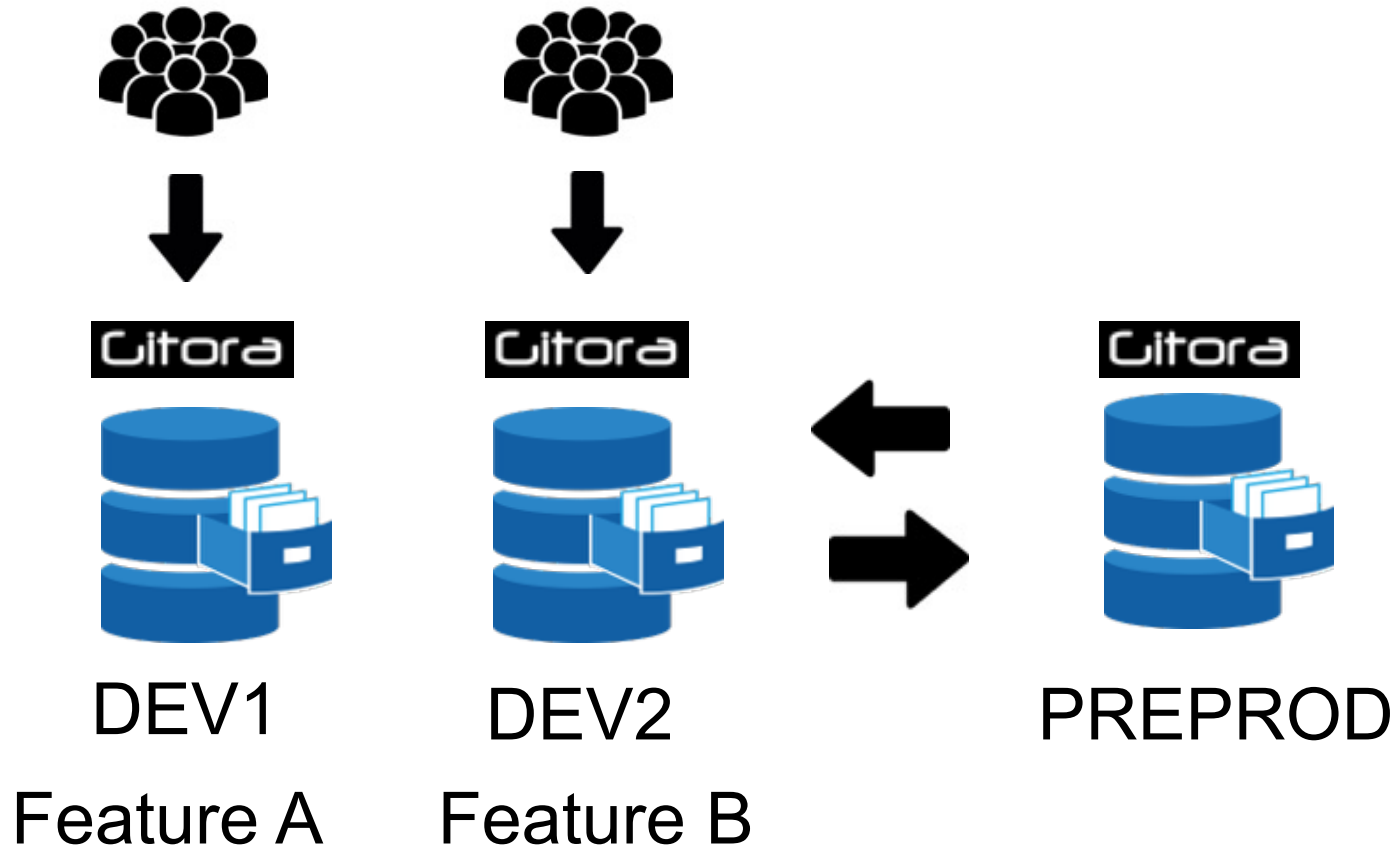
Basic branching and merging is possible

Better automatic testing

Version control rules are enforced at the DB level and managed by a release manager.

Demo!!!!

Demo Setup (One Repository)



About Gerger

Hundreds of customers in 30+ countries.

Gitora

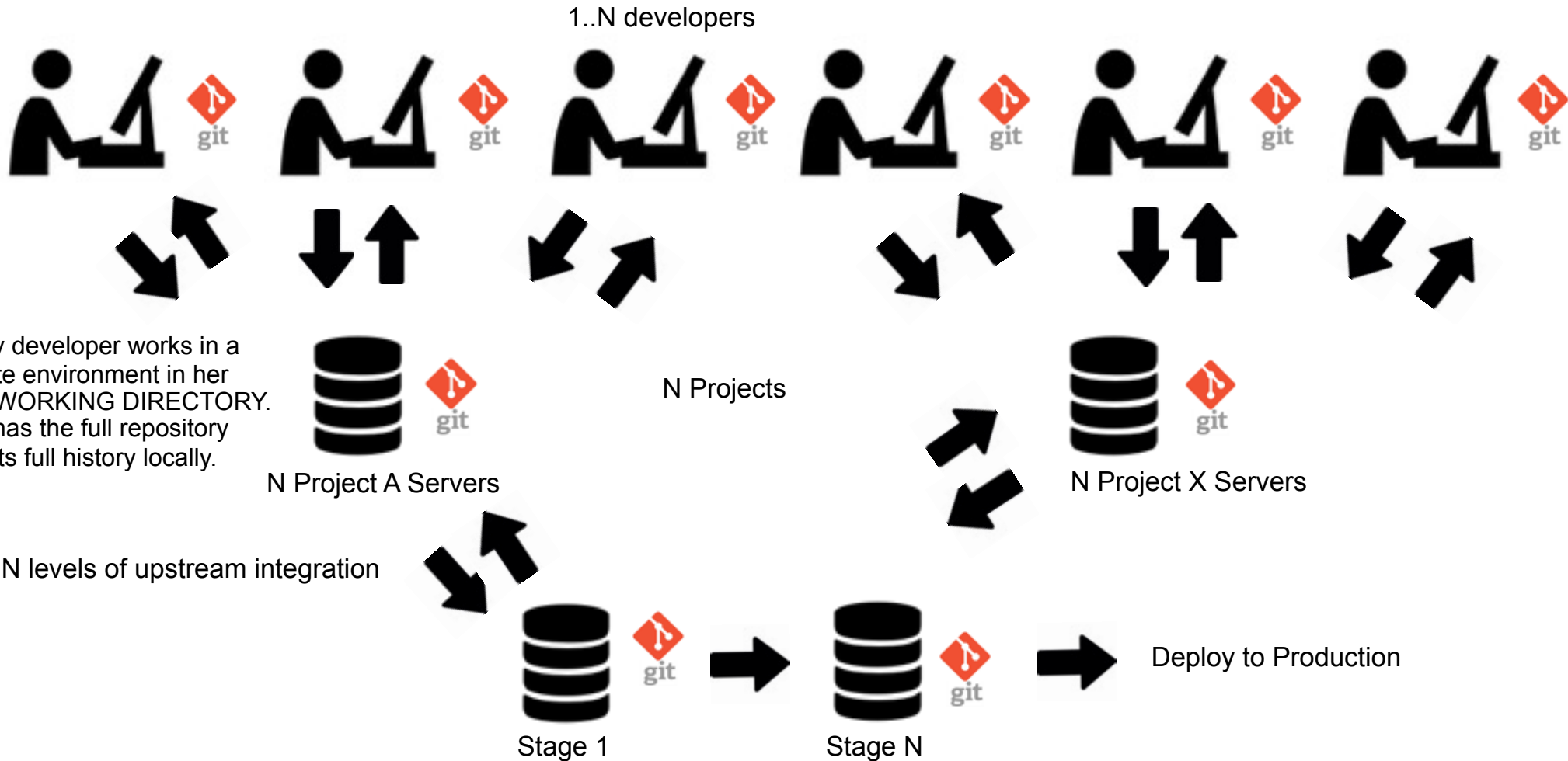
Manage your PL/SQL code with Git

<http://www.gitora.com>

Contact me for a free consultation:

yalim.gerger@gerger.co

This is how best developers work with Git



No Queues

Automatic deployment

Source code is in Git

No Blocking

No blocking between projects, features

Faster hot fixes

No Interruptions

Branching and merging is possible

Great automated testing

Version control rules are enforced by the deployment pipeline