In-Field Healing of Integration Problems with COTS Components

Hervé Chang, Leonardo Mariani, Mauro Pezzè
University of Milano Bicocca
University of Lugano

COTS products

Software elements

commercial or free

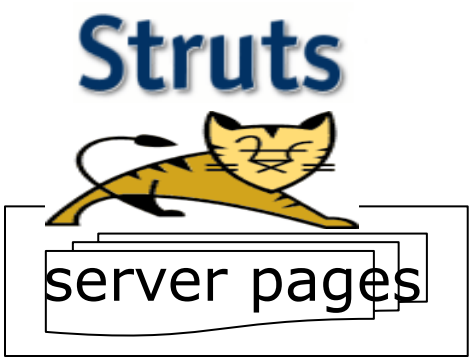
big or small

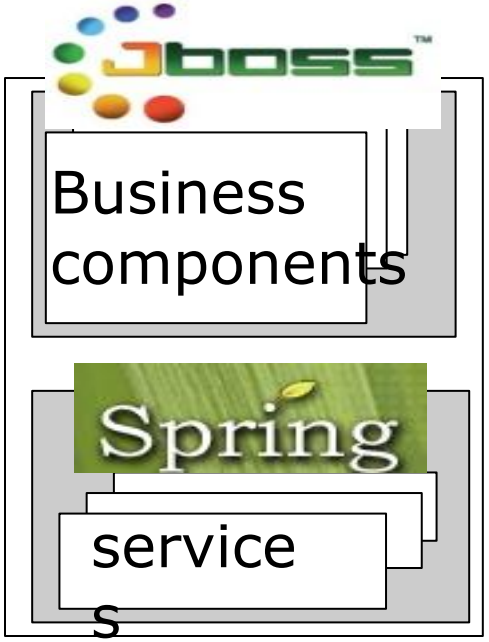
reused in many contexts

complex integration

partial documentation

no access to source code





Problems in integrating COTS products

misuse of COTS products

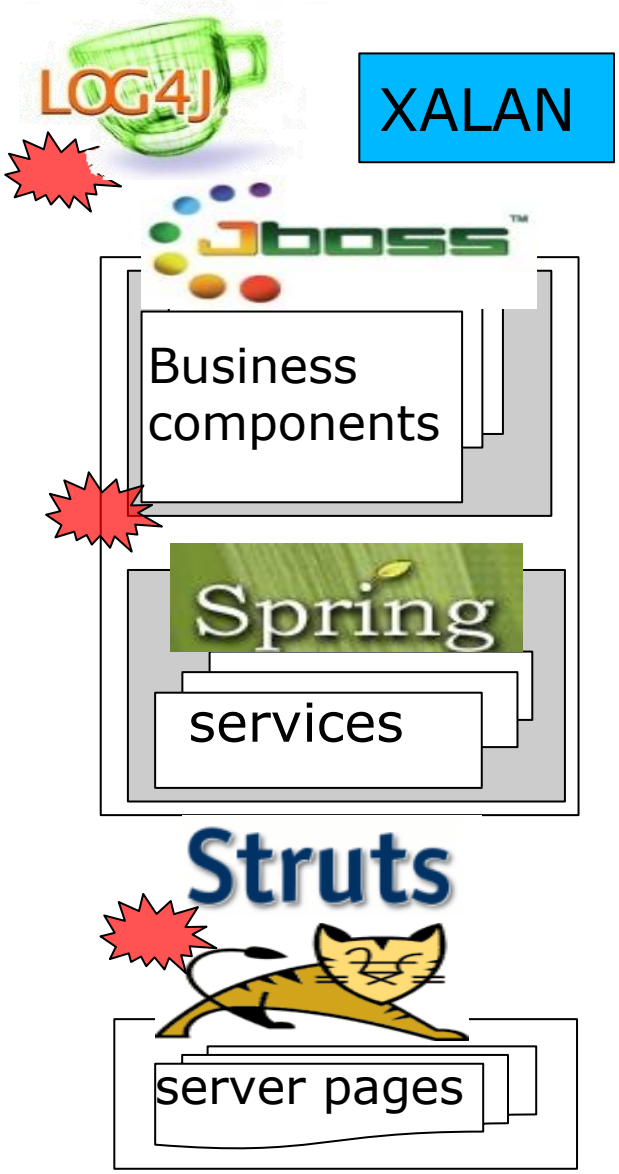
hard to fix when

we need to know both

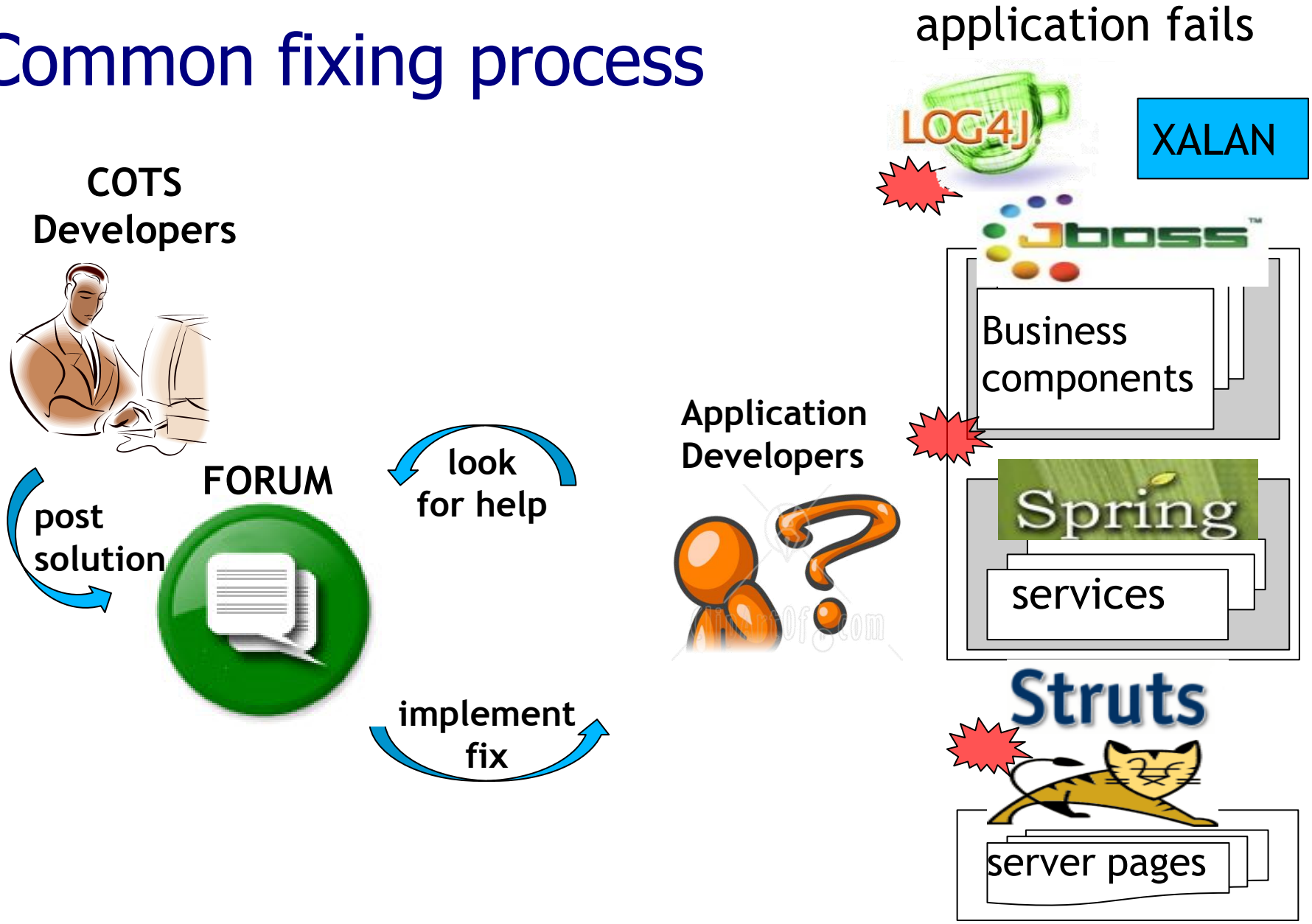
the application

and

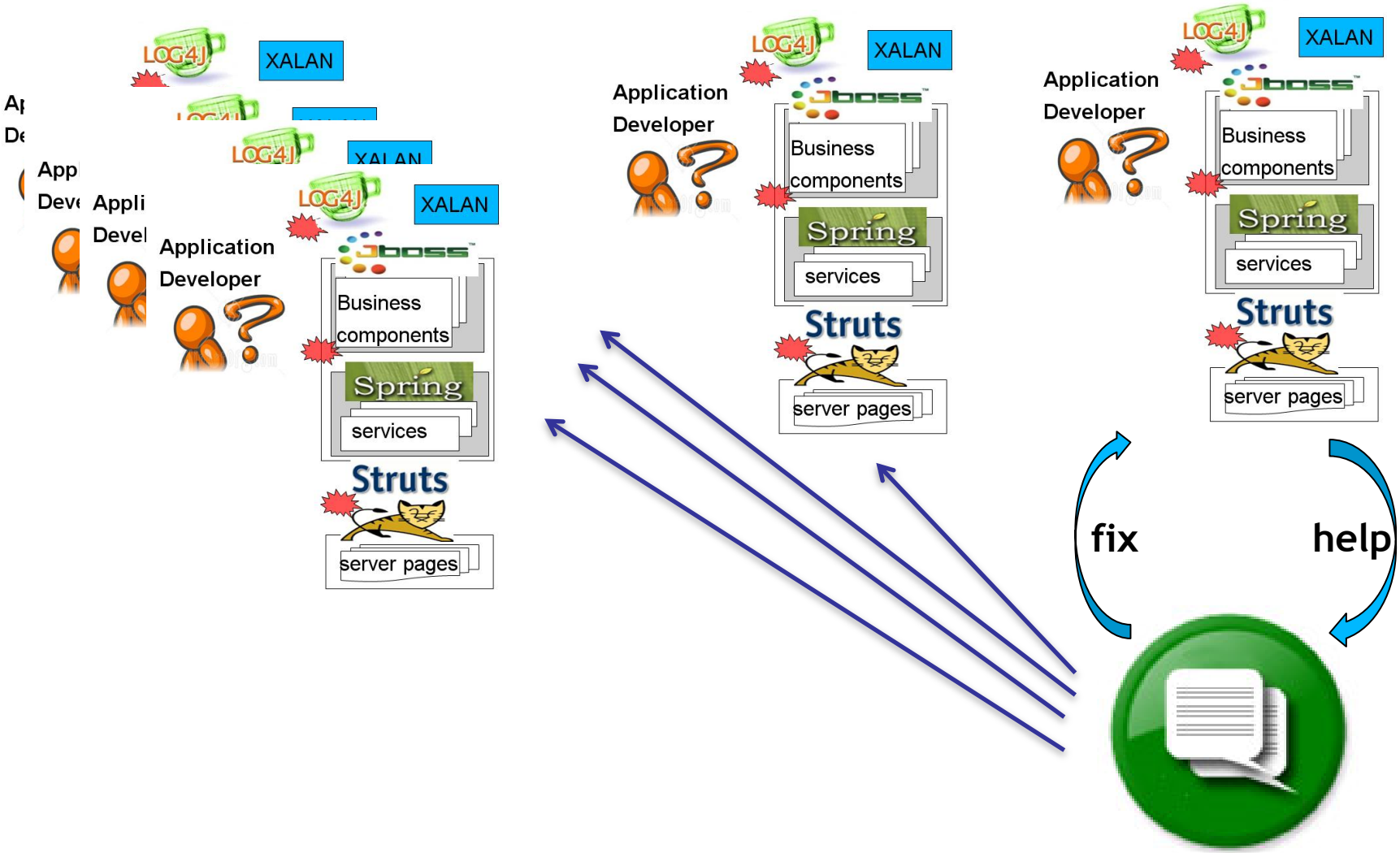
the COTS product



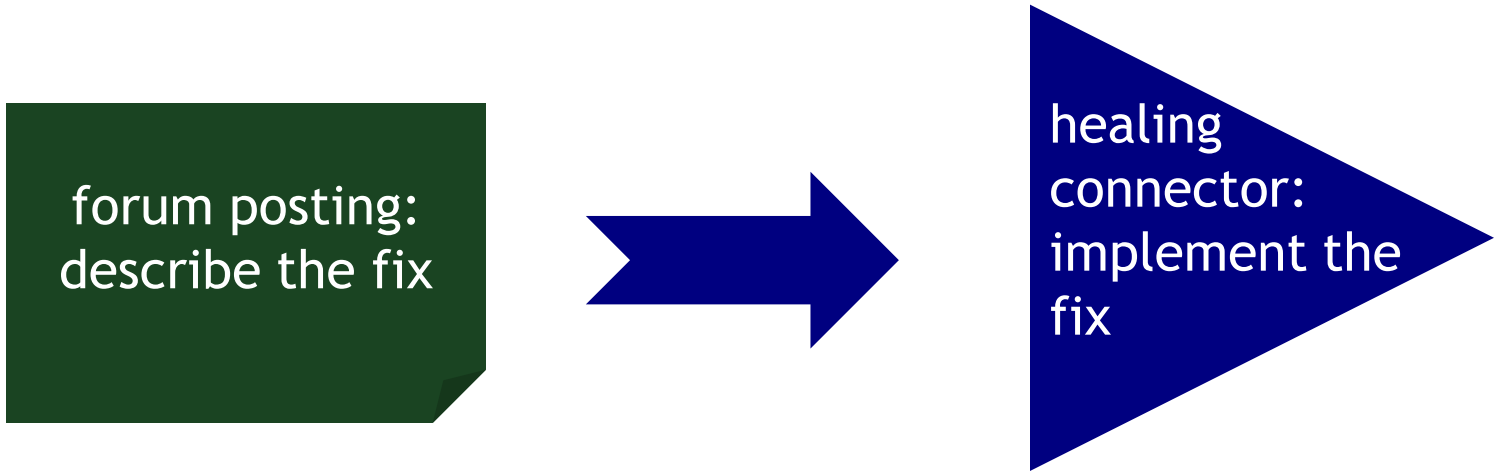
Common fixing process



Reuse fixes


LOG4J
XALAN
Application
Developer
Business
components
Spring
services
Struts
server pages
fix
help

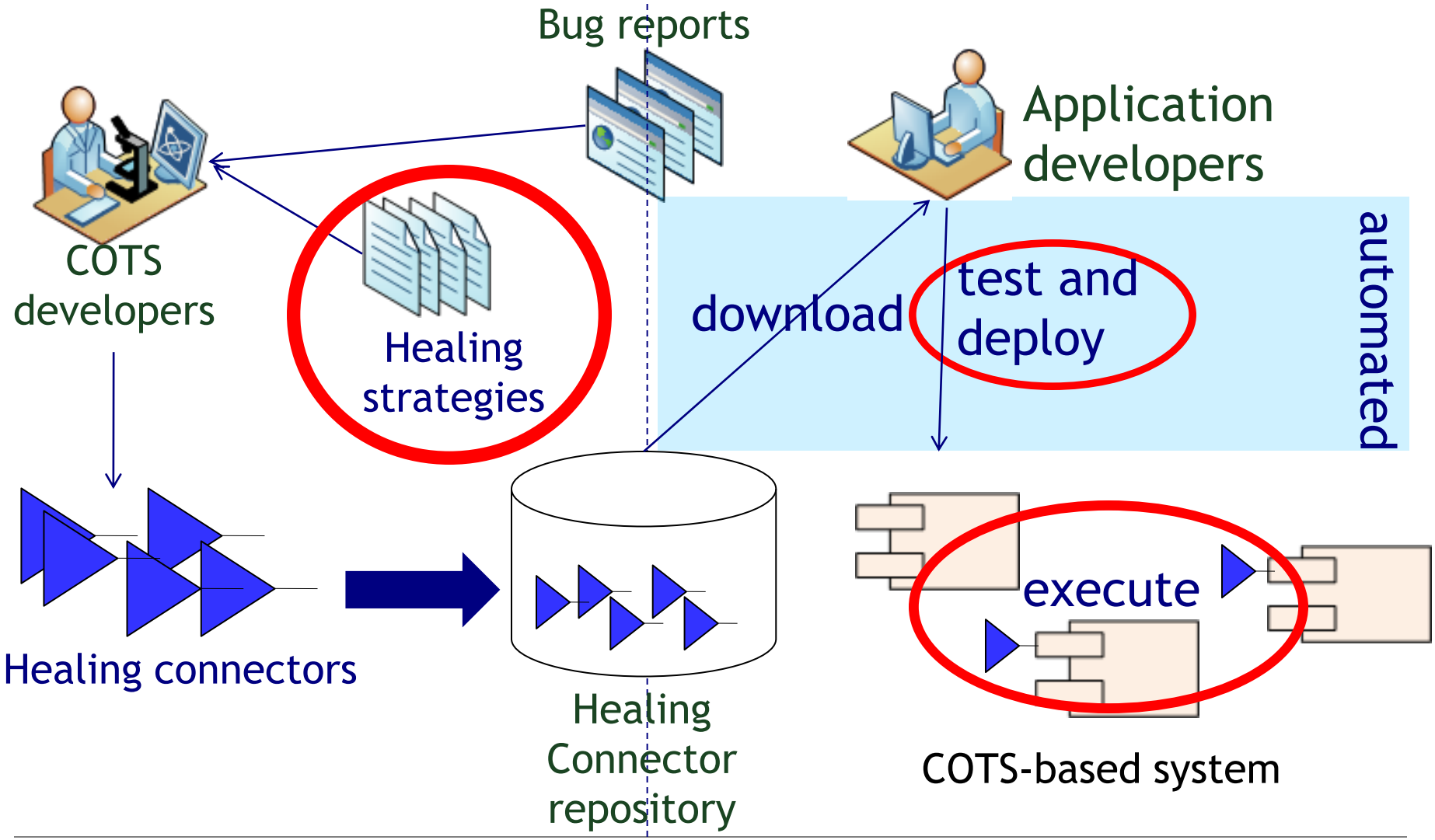
Healing connectors



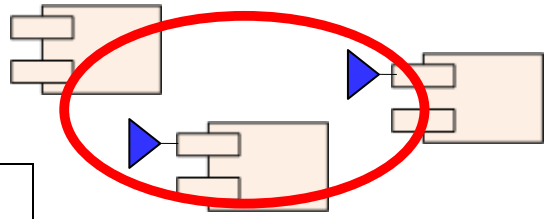
- distribution: manual on the Web
  deployment: manual after failure
- activation: off-line

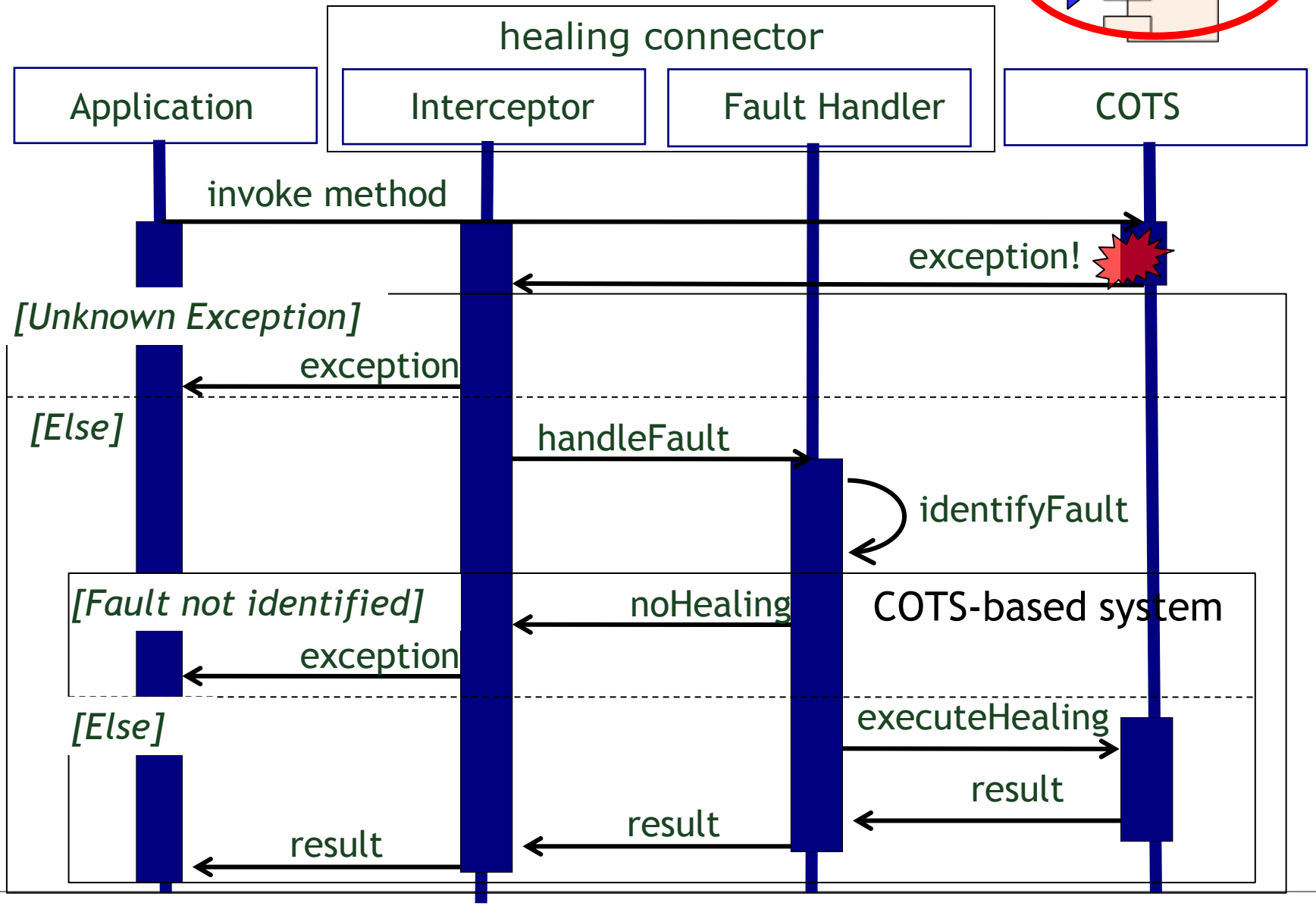
- distribution: automatic
- deployment: automatic before failure
- activation: on-the-fly and in-the-field

A methodology



Execute



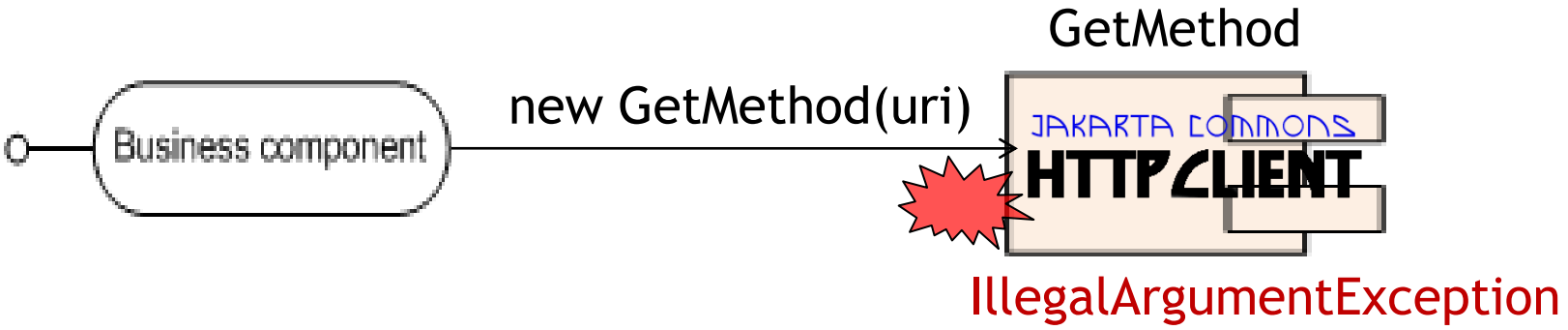

healing connector
Application
Interceptor
Fault Handler
COTS
invoke method
exception!
[Unknown Exception]
exception
[Else]
handleFault
identifyFault
[Fault not identified]
noHealing
COTS-based system
exception
[Else]
executeHealing
result
result
result

Healing strategies

- from experience
  (Sun JDK, JBoss, Spring, Apache,…)

classes of faults	healing strategies
invalid parameter	translate parameter
wrong use of the method interface	prepare component
faulty method	alternative operation
environment fault	change environment

Invalid parameter

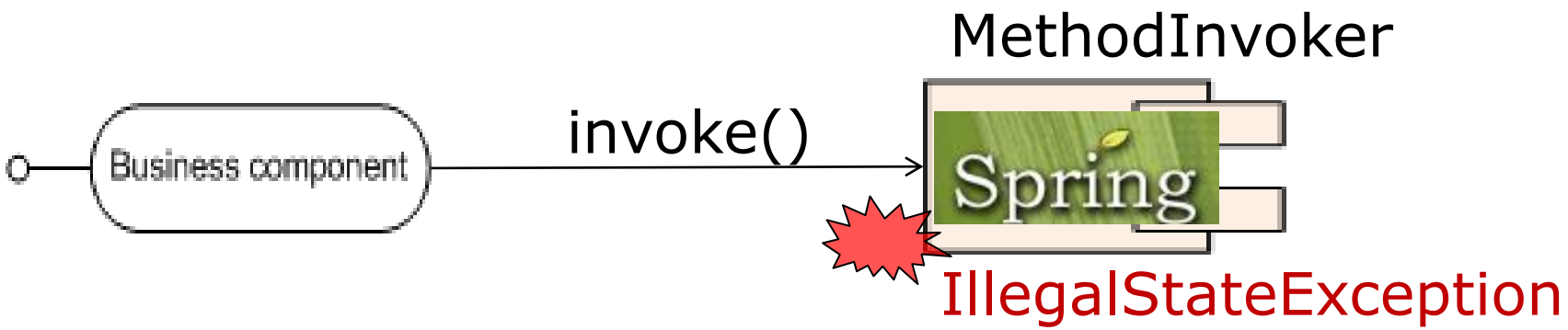


uri=``http://web/images/logo[1].gif``

translate parameter

check invalid characters in uri
(uri=«http://web/images/logo**%5B1%5D**.gif»)

Wrong usage of the method interface



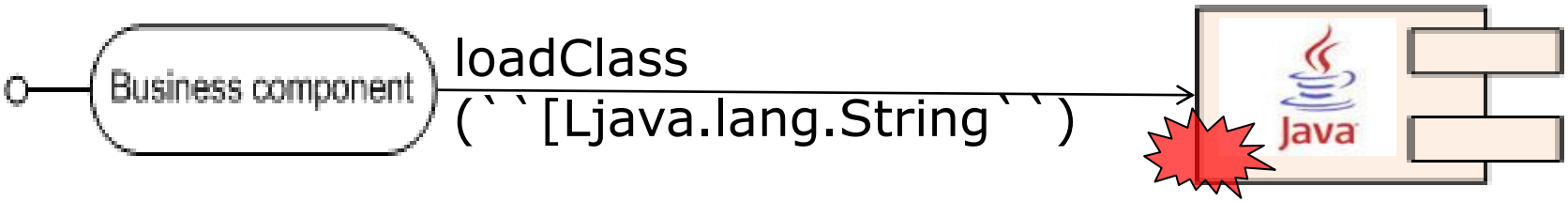
prepare component

isPrepared()?

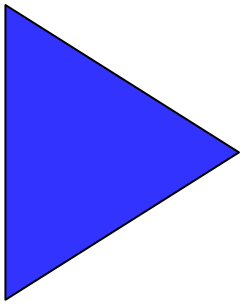
prepare(); invoke()

Faulty method

ClassLoader under Java 6



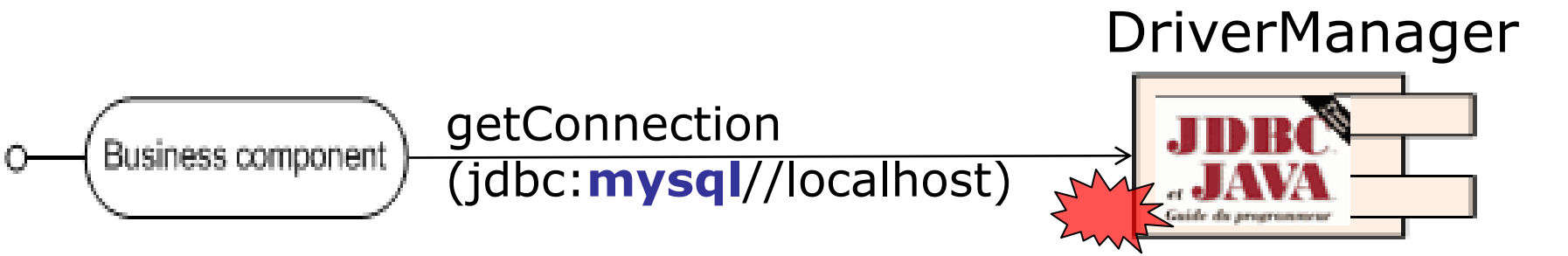
ClassNotFoundException

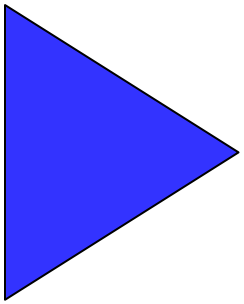


alternative operation

if array syntax
forName()

Environment fault





change environment

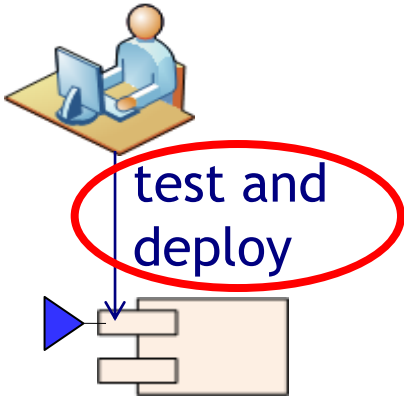
if MySQL driver not loaded

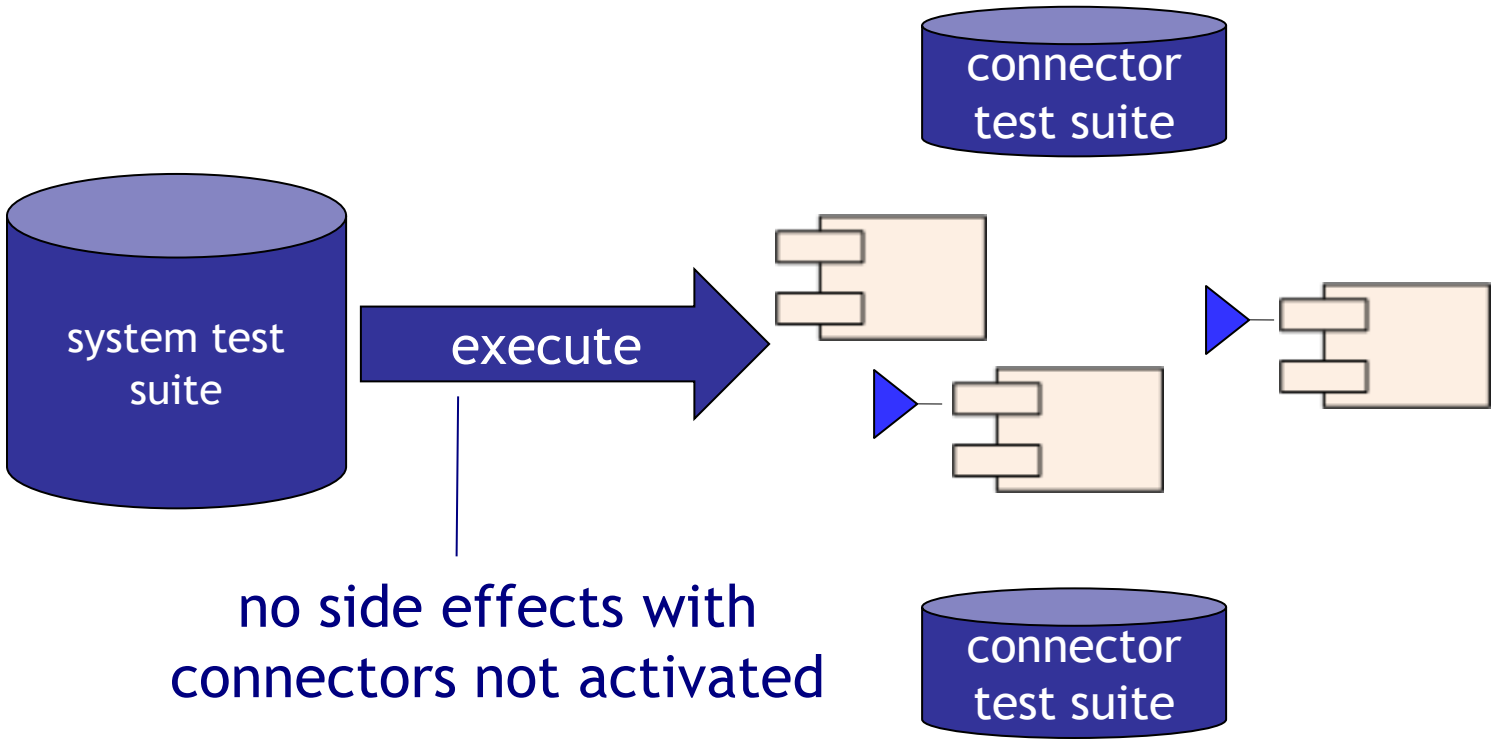
add the jar file of MySQL

register the driver

getConnection()

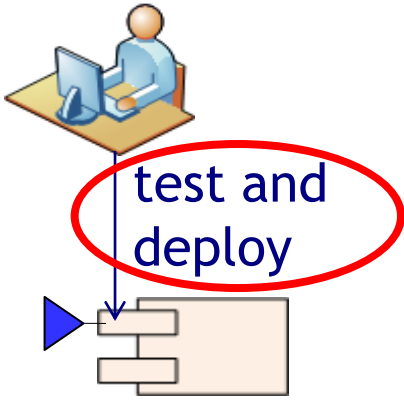
Test and deploy



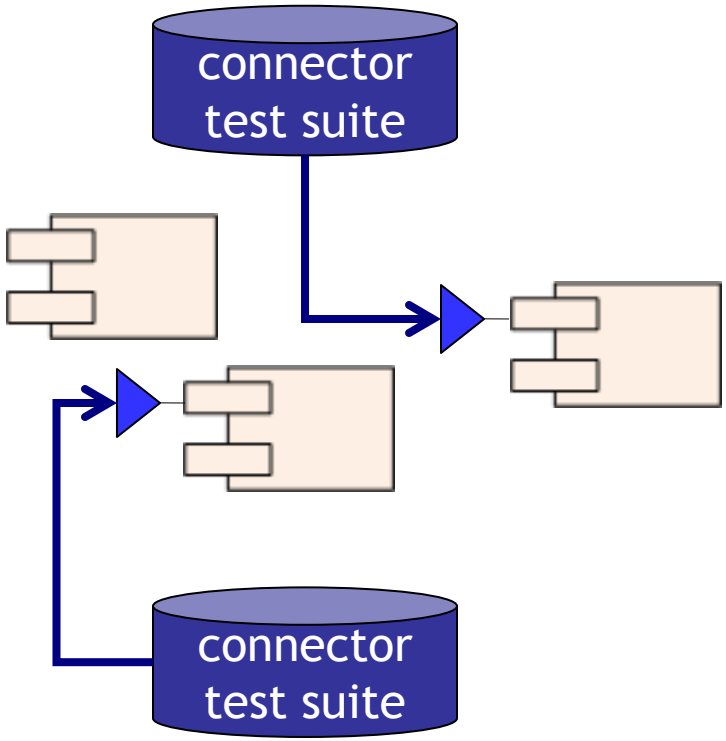


no side effects with
connectors not activated

Test and deploy



successful healing
and no connector interactions
when connectors are
activated



Validation

- reusing healing strategies
  - conceptually:
    same strategy for many cases
    (4 strategies for 18 cases)
  - by adapting:
    reconfigured healing connectors for different systems
    (2 connectors adapted to 5 cases)
  - with no changes:
    full reuse
    (4 connectors reused for 8 cases)

Validating overhead

Healing Connector	COTS component	Application	#Measures	Execution Time (ms)		
				no connec.	inact. heal.	active heal.
Change parameter and retry	Sun JRE Java Net	ActiveMQ	300	0.066	0.071	0.167
		ServiceMix	100	0.092	0.106	0.298
Call operation and retry	Geronimo Java Mail	J2EE Web Application	100	370.24	373.68	394.53
Replace Call	Sun JRE 1.6 Classloader	Geronimo	200	0.026	0.029	6.512
		JBoss	700	0.008	0.009	0.143
Change env and retry	Xalan	Magnolia	100	48.67	53.26	20.78

[illegible] ective than
[illegible] is difficult to be
[illegible] end user (<24ms)

Conclusion

Healing connectors
a simple but effective solution to a frequent and expensive problem