



Research and practices on application server in SEI of Peking University

ZHOU Minghui

zhmh@sei.pku.edu.cn

*School of Electronics Engineering and
Computer Science Peking University, Beijing,
100871, China*



北京大學

- Vision of Peking University Software Engineering Institute
- Practices on middleware
 - A pluggable framework for application server construction
 - Architecture-based middleware management
- Practice on software engineering
- Summary: what's next?



Vision

- Establishing the fundamental of software
- Promoting software engineering technologies in software enterprises



- Practice
- Experience
- Open view and creative mind



A pluggable framework for application server construction



Middleware sits "in the middle"

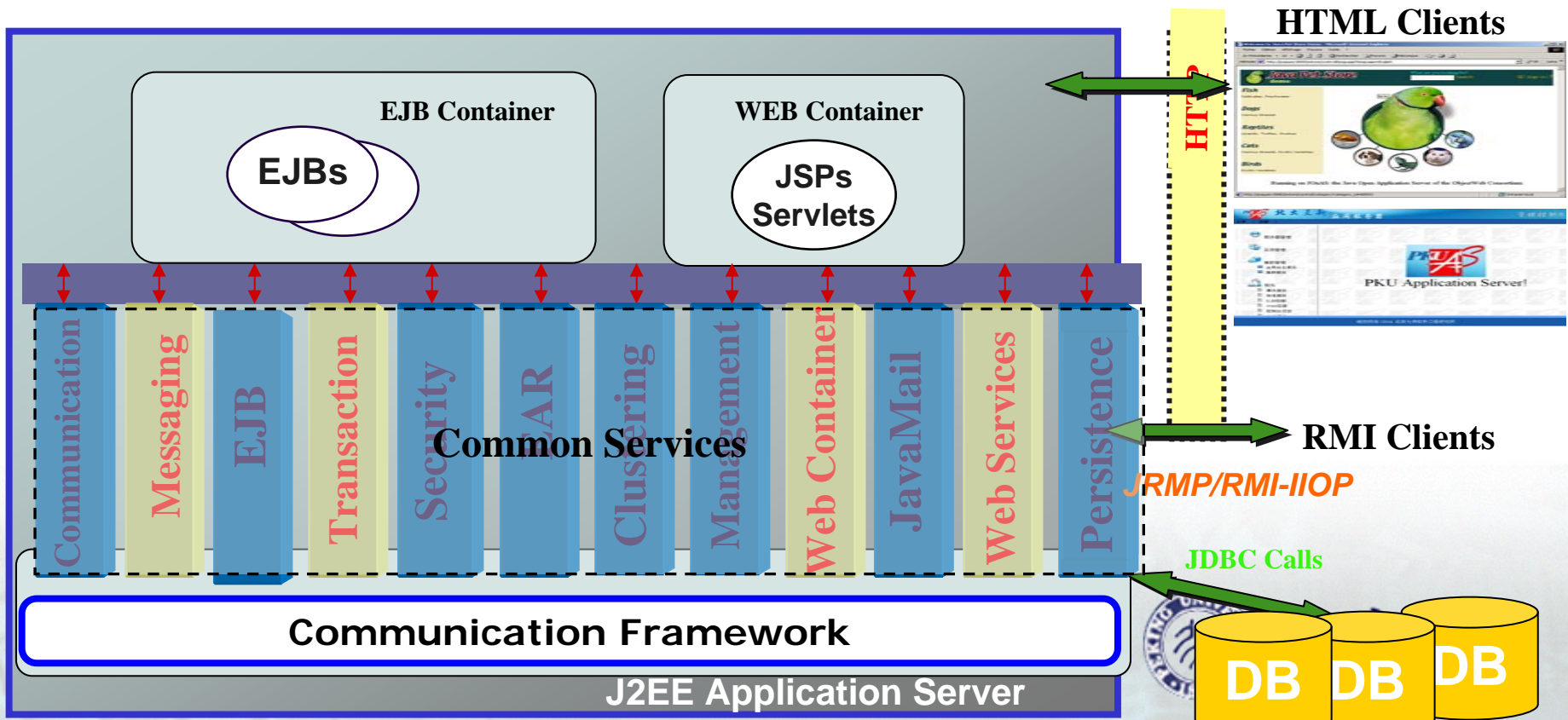
- between application and operating system;
- between application software working on different operating systems



北京大學

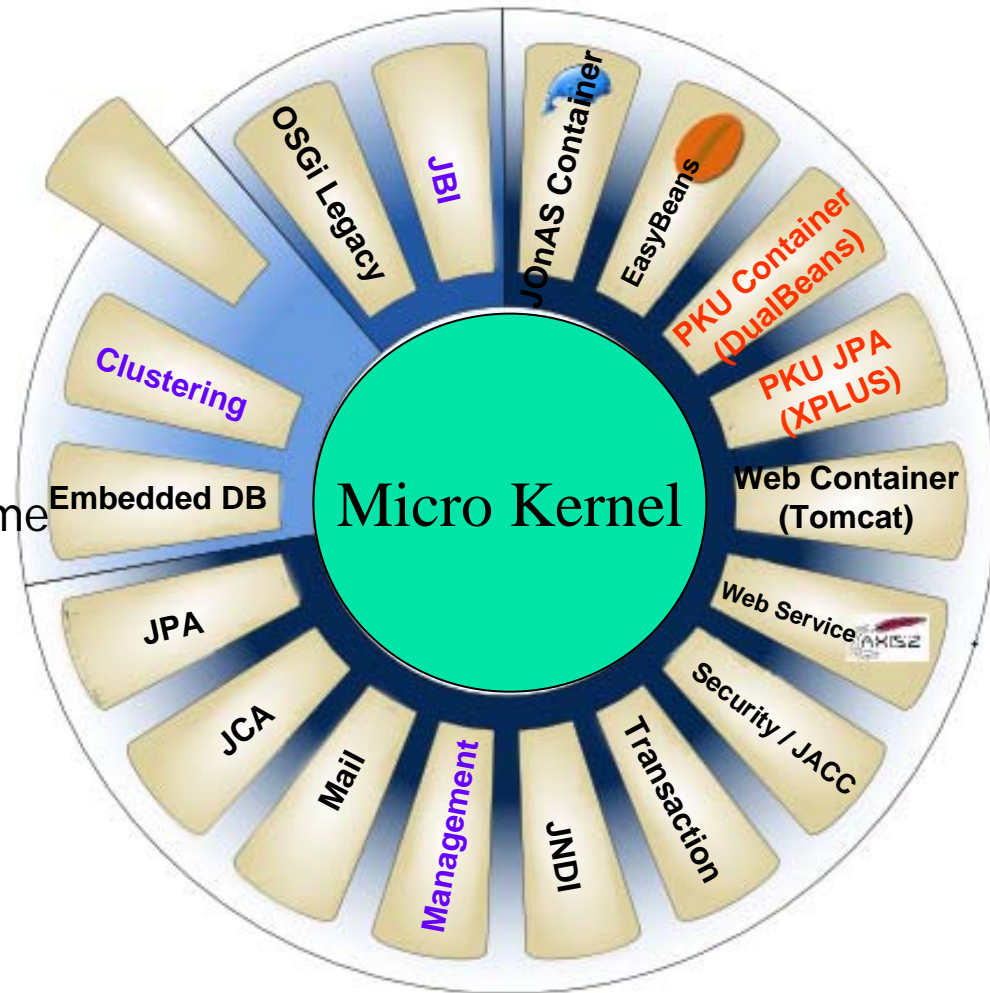
Motivation

- Application server is becoming complex never before
 - Middleware services (modules) have reached to 20, 23 specifications in JavaEE 5
 - Architecture is clear, but dependency among modules is out of control
- Vendors are inclined to selectively implement some modules, and to integrate some reliable Off-the-Shelf modules
 - Dependency among modules is difficult to handle, tight coupling, without flexibility



Motivation

- We have an application server named PKUAS, mergence between JOnAS and PKUAS is difficult
 - Modification should be minimum to the existing modules
 - Modules should be pluggable
- Goal
 - Scalable : add/delete modules without influencing others
 - Customizable: switch (same) modules easily
 - Dynamic: configure easily at runtime
 - Maintainable: upgrade easily



A pluggable framework

■ Principle

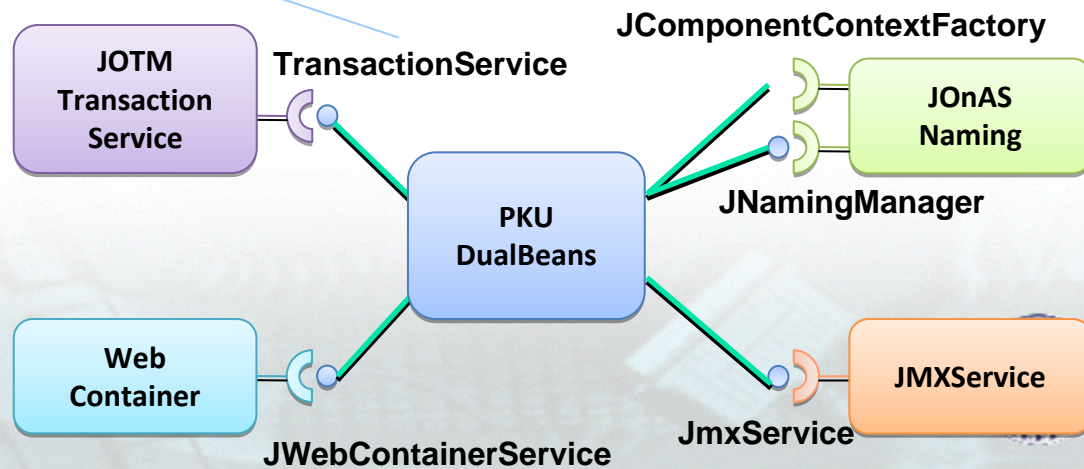
- Greatest common divisor: extraction of overlap
- Least common multiple: abstraction of difference (same entrance)
- Separation between configuration and code

```
public Xid getXidByTransaction(Transaction tx);
```

```
public Transaction getTransactionByGlobalId(Xid xid);
```

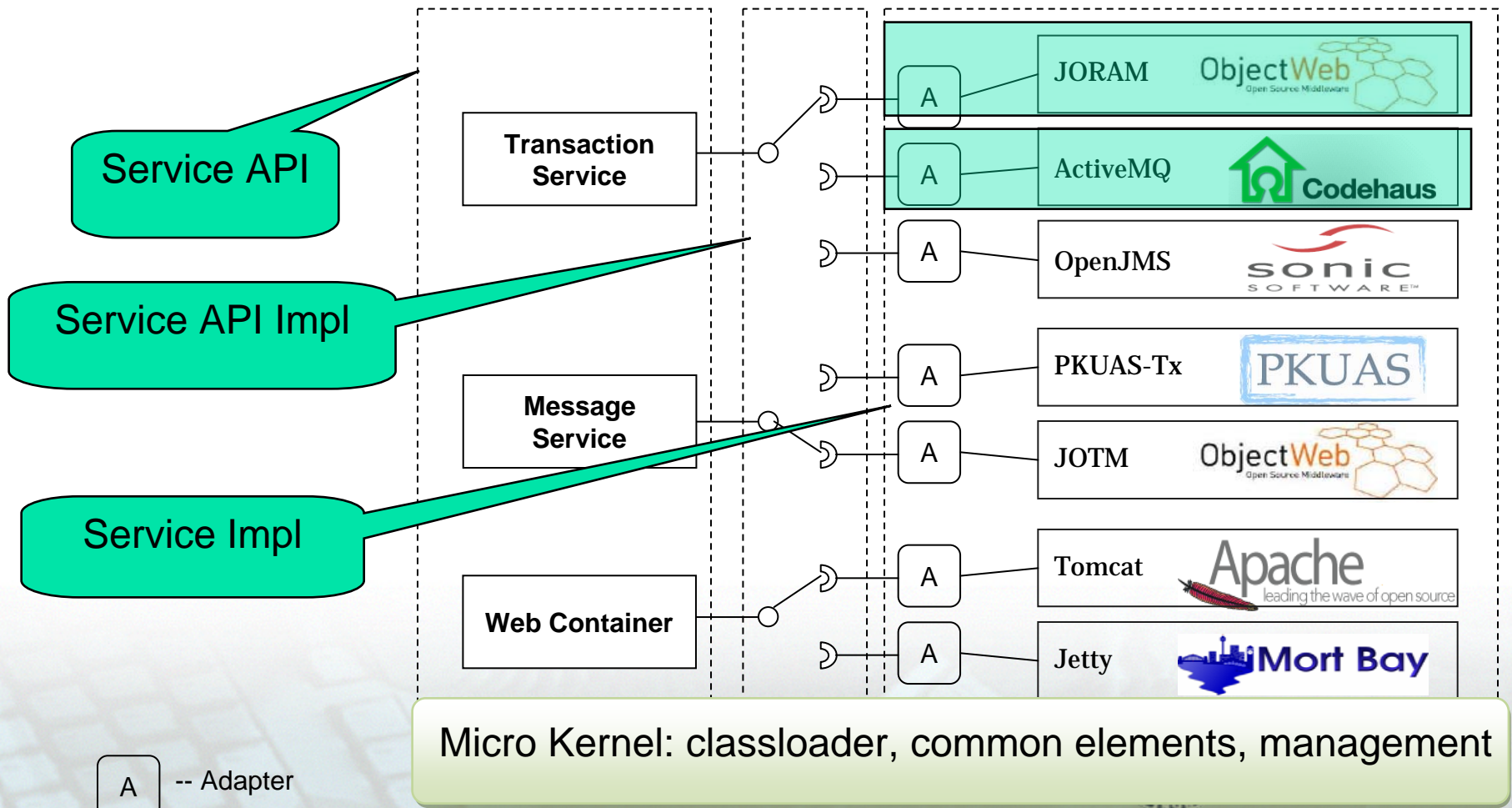
Add 2 Methods for DualBeans

TransactionManager



A pluggable framework

- Modularity and Flexibility: Easy to change module implementation even at runtime
- Configuration and customization is becoming simpler





Architecture-based middleware management



北京大學

Motivation

- In order to deploy and configure application on AS in real environment, too much effort is needed
 - Boring work
 - Build a configuration is quite difficult
 - Heterogeneous files (properties, XML, ...)
 - Numerous files and parameters
 - JOnAS = 46 files, Apache configuration file = 160 lines
 - Computing environments are becoming increasingly sophisticated
 - Numerous complex software
 - Heterogeneous programming models
 - Configuration facilities are generally proprietary
 - Skillful person
 - Technical knowledge
 - Domain knowledge
- Why don't make the application environment-sensitive itself?



Management Process

■ Monitoring

- Measuring parameters of the controlled system that are relevant for the function under control

■ Decision

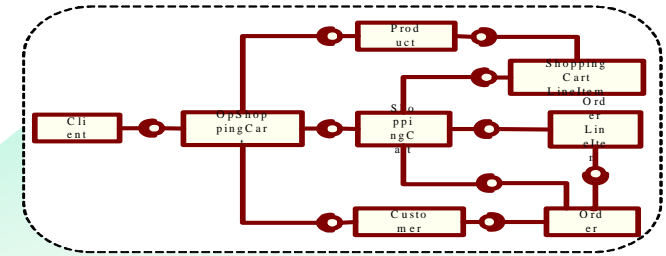
- Deciding the corrective action to be taken.
- Analyzing and planning

■ Execution

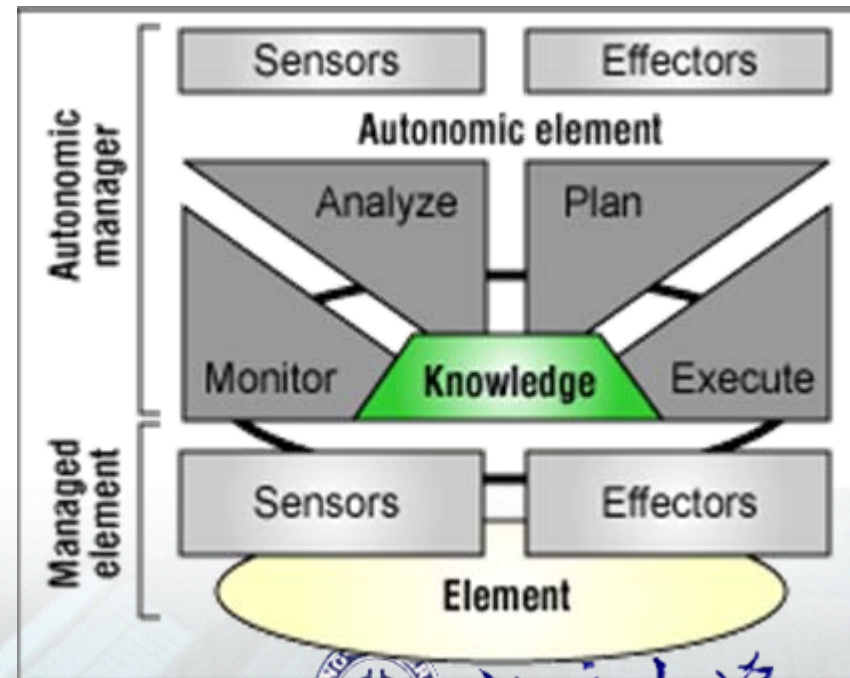
- Actually performing the actions on the controlled system.

■ Knowledge base

- Represents information on the controlled system.



Application SA



北京大學

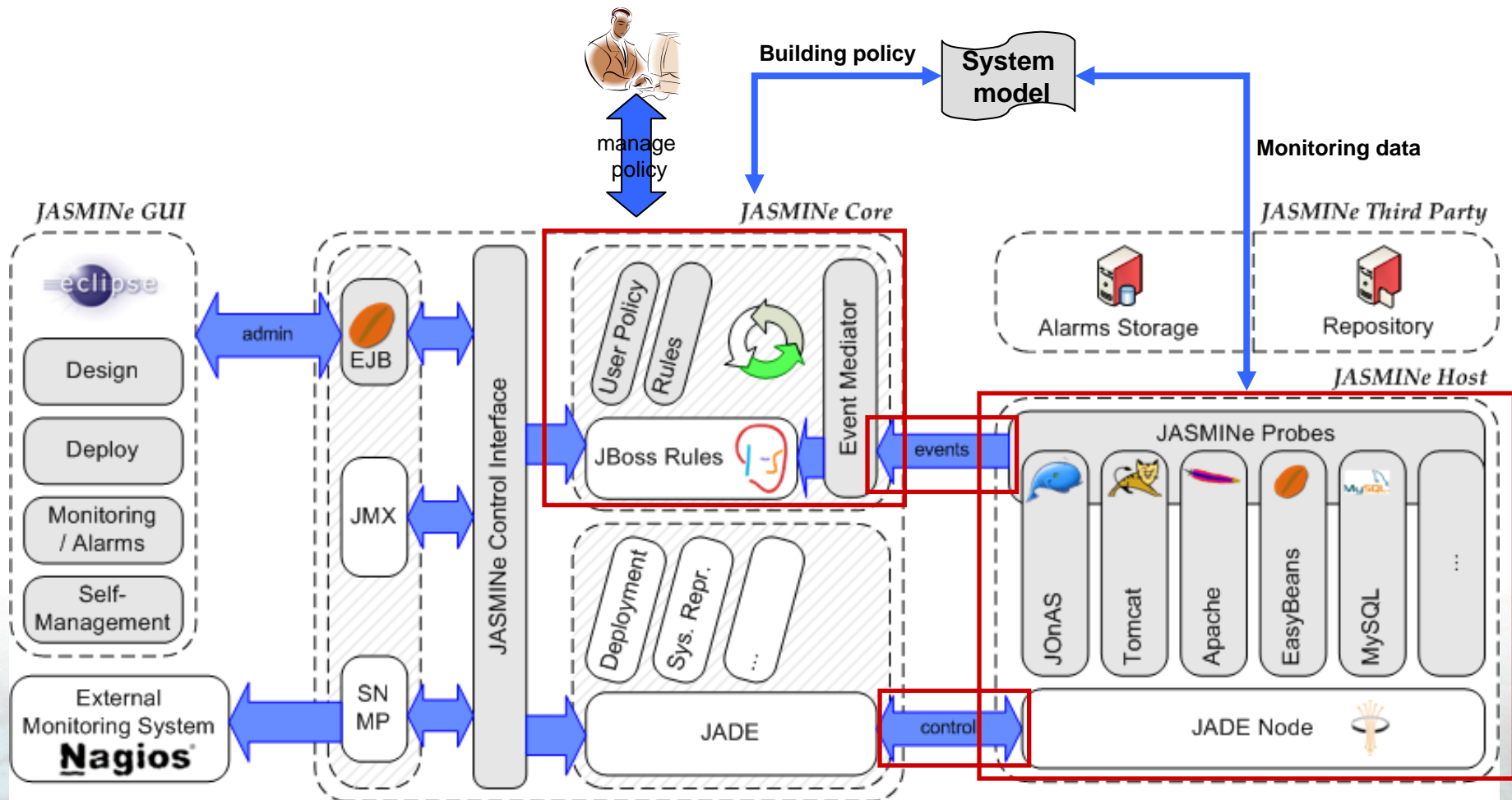


Challenges

- Mechanisms, to support the complexity of the management process
 - Management process is general
 - There are too many facilities need to be handled whose relationship are too complex
- Policies, to tune the system status automatically
 - Source status -> destination status
 - Decision policies decide parameter tuning for status migration

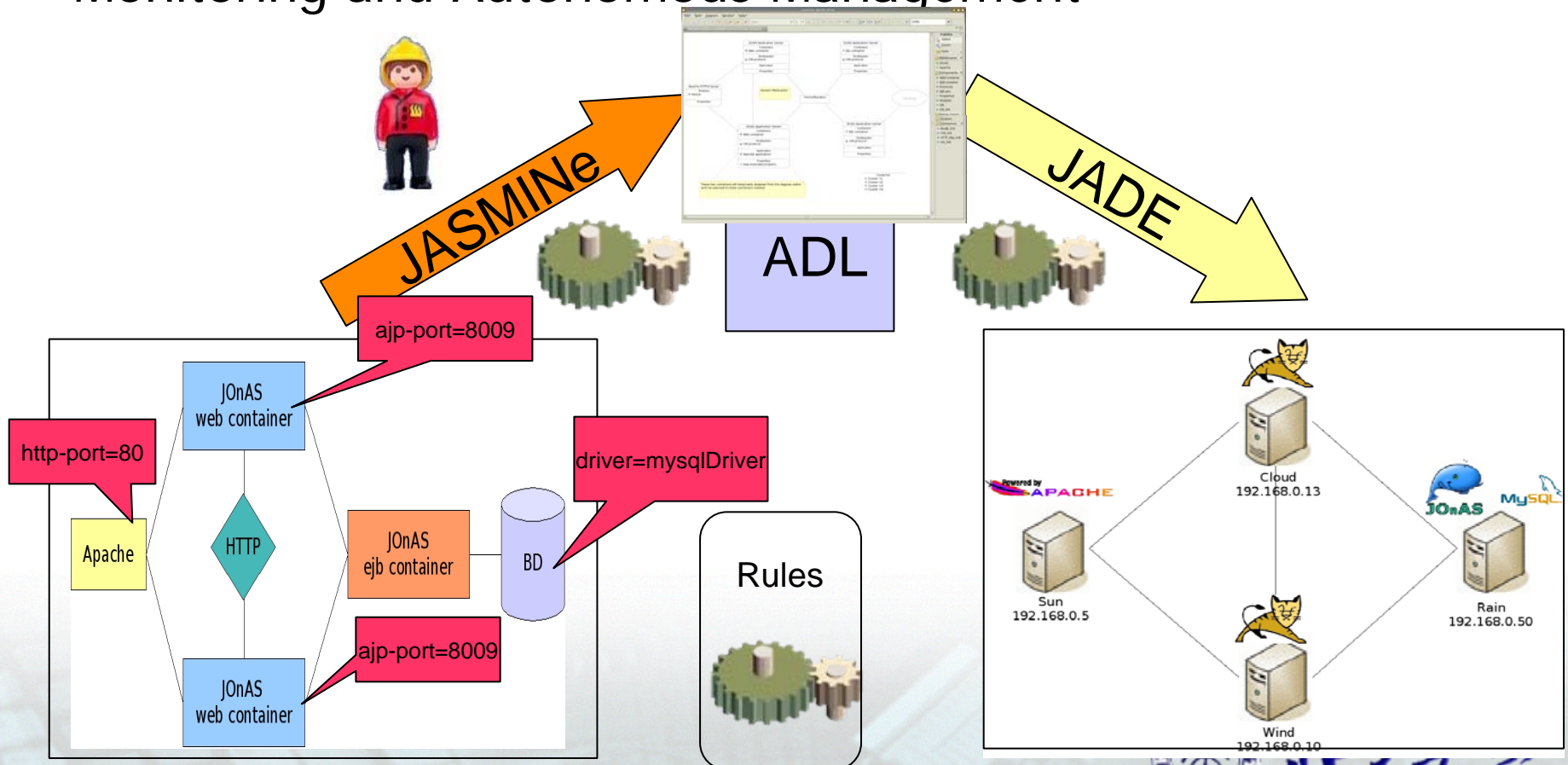


- Environment-driven configuration, deployment and adjustment framework



JASMINe

- Design GUI: Eclipse EMF/GMF for architecture description
- Configuration and Deployment: Logical configuration TO physical configuration
- Monitoring and Autonomous Management





Data driven measures of software development



北京大學



Motivation

- Use data to understand software development and help tradeoff between the main issues in software projects
 - Schedule
 - Quality
 - Cost
- Understand the foundation in software engineering
 - What makes programmers programming different?
 - Individual differences
 - Cognitive science
 -



- Version control and problem tracking data are the most promising sources for measures
 - Necessary for development
 - Can be automatically collected

```
407 -----
408
409 f723 | wangzy06 | 2008-06-25 12:08:51 +0800 (Wed, 25 Jun 2008) | 1 line
410 Changed paths:
411   D /branches/TRY-MODULES/conformance-test/libs/client.jar
412   M /branches/TRY-MODULES/conformance-test/libs/jonas_log.jar
413 -----
414
415
416 f722 | zhangyf05 | 2008-06-25 11:24:37 +0800 (Wed, 25 Jun 2008) | 1 line
417 Changed paths:
418   M /branches/TRY-MODULES/modules/remoting/src/main/resources/META-INF/pku
419   M /branches/TRY-MODULES/modules/services/transaction/src/main/resources/
420 -----
421
422 make Remoting module can transport Kid
423 -----
424
425 f721 | wangzy06 | 2008-06-23 19:48:21 +0800 (Mon, 23 Jun 2008) | 1 line
426 Changed paths:
427   A /branches/TRY-MODULES/conformance-test/libs/client.jar
428   M /branches/TRY-MODULES/conformance-test/libs/jonas_log.jar
429 -----
430
431
432 f720 | wangzy06 | 2008-06-23 11:06:14 +0800 (Mon, 23 Jun 2008) | 1 line
433 Changed paths:
434   M /branches/TRY-MODULES/conformance-test/jndi.properties
435 -----
436
437
438 f719 | liyan05 | 2008-06-21 14:33:56 +0800 (Sat, 21 Jun 2008) | 1 line
439 Changed paths:
440   M /branches/TRY-MODULES/modules/services/security/src/main/java/pku/as/s
441 -----
442
443
444 f718 | liyan05 | 2008-06-21 14:33:42 +0800 (Sat, 21 Jun 2008) | 1 line
445 Changed paths:
446   M /branches/TRY-MODULES/modules/services/security/src/main/java/pku/as/s
447 -----
448
449
450 modify the getRolesOfUser method, when the roles collection is null, return
451 -----
452
453
454 f717 | liyan05 | 2008-06-21 14:31:12 +0800 (Sat, 21 Jun 2008) | 1 line
455 Changed paths:
456   M /branches/TRY-MODULES/modules/launchers/felix-launcher/pom.xml
457 -----
458
459
460 f716 | liyan05 | 2008-06-21 14:30:52 +0800 (Sat, 21 Jun 2008) | 1 line
461 Changed paths:
462   M /branches/TRY-MODULES/modules/container/ejbcontainer/src/main/java/pku
```

```
11042 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/libraries/management/tools/src/main/java/org/ow2/jonas/lib/manages
11043 11044 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/libraries/management/tools/src/main/java/org/ow2/jonas/lib/manages
11044 11045 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/libraries/management/tools/src/main/java/org/ow2/jonas/lib/manages
11045 11046 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/services
11046 11047 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/services/ejb/pom.xml
11047 11048 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/services/jonas-ear/src/main/java/org/ow2/jonas/ear/internal/AppCl
11048 11049 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/services/jonas-ear/src/main/java/org/ow2/jonas/ear/internal/EarDep
11049 11050 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/services/jonas-ear/src/main/java/org/ow2/jonas/ear/internal/JNDIac
11050 11051 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/services/jonas-ear/src/main/resources/betadata.xml
11051 11052 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/services/pom.xml
11052 11053 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/services/Versioning/src/main/java/org/ow2/jonas/Versioning/Interna
11053 11054 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/services/web-container/base/src/main/java/org/ow2/jonas/web/base/F
11054 11055 e13434:alltokens:2008-03-28 07:00:00 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/branches/Versioning-eb/modules/tools/jonas-ant-task/src/main/java/org/ow2/jonas/ant/cluster/Clus
11055 11056 e13433:liyan:2008-03-28 06:01:47 -0400 (Fri, 28 Mar 2008) |2:M:/sandbox/pkuas/jonas/modules/services/pkuas-container/src/main/java/pku/as/application/Application.java
11056 11057 e13433:liyan:2008-03-28 06:01:47 -0400 (Fri, 28 Mar 2008) |2:M:/sandbox/pkuas/jonas/modules/services/pkuas-container/src/main/java/pku/as/actdata/ApplicationInfo.java
11057 11058 e13433:liyan:2008-03-28 06:01:47 -0400 (Fri, 28 Mar 2008) |2:M:/sandbox/pkuas/jonas/modules/services/pkuas-container/src/main/java/pku/as/security/SecurityManager.java
11058 11059 e13433:liyan:2008-03-28 06:01:47 -0400 (Fri, 28 Mar 2008) |2:M:/sandbox/pkuas/jonas/modules/services/pkuas-container/src/main/java/pku/as/security/SecurityRealm.java
11059 11060 e13432:duisuep:2008-03-28 05:58:19 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-en/ejb2/programmer_guide.xml
11060 11061 e13432:duisuep:2008-03-28 05:58:19 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-en/services/ejb2/mp2_usage.xml
11061 11062 e13432:duisuep:2008-03-28 05:58:19 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-en/services/ejb2/db_access_exp.xml
11062 11063 e13432:duisuep:2008-03-28 05:58:19 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-en/services/ejb2/db_deploy.xml
11063 11064 e13432:duisuep:2008-03-28 05:58:19 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-en/services/ejb2/deploy.xml
11064 11065 e13432:duisuep:2008-03-28 05:58:19 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-en/services/ejb2/entity.xml
11065 11066 e13432:duisuep:2008-03-28 05:58:19 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-en/services/ejb2/entity_tuning.xml
11066 11067 e13432:duisuep:2008-03-28 05:58:19 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-en/services/ejb2/wdl.xml
11067 11068 e13432:duisuep:2008-03-28 05:58:19 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-en/services/ejb2/packaging.xml
11068 11069 e13432:duisuep:2008-03-28 05:58:19 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-en/services/ejb2/packaging.xml
11069 11070 e13432:duisuep:2008-03-28 05:42:16 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-en/esnat.xml
11070 11071 e13428:duisuep:2008-03-28 05:40:38 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/resources/commen.css
11071 11072 e13428:duisuep:2008-03-28 05:40:38 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/resources/html.xml
11072 11073 e13427:formaeif:2008-03-27 15:53:41 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas/assemblees/jonas-configuration/src/main/resources/conf/jonas.properties
11073 11074 e13427:formaeif:2008-03-27 15:53:41 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas/assemblees/jonas-configuration/src/main/resources/conf/jonas.properties
11074 11075 e13425:duisuep:2008-03-28 05:06:53 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_doc/src/docbook/doc-tr/rtswal/rtswalDocBook.xml
11075 11076 e13424:eyindanga:2008-03-28 04:22:04 -0400 (Fri, 28 Mar 2008) |1:R:/jonas/trunk/jonas_tests/w/build.xml
11076 11077 e13423:duisuep:2008-03-28 04:06:47 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas_doc/build.xml
11077 11078 e13423:duisuep:2008-03-28 04:06:47 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas_tests/w/build.xml
11078 11079 e13421:duisuep:2008-03-28 03:55:43 -0400 (Thu, 27 Mar 2008) |1:R:/sub-projects/hta2book/trunk/hta2book/trunk/doc/html/doc/LogModules.html
11079 11080 e13407:benoitf:2008-03-27 15:53:41 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas_tests/build.xml
11080 11081 e13404:marainis:2008-03-27 14:07:01 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas/assemblees/jonas-osi/src/main/resources/bin/newjc.sh
11081 11082 e13403:benoitf:2008-03-27 12:57:49 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas/modules/service
11082 11083 e13401:marainis:2008-03-27 13:53:41 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas/modules/tools/jonas-ant-task/src/main/java/org/ow2/jonas/ant/cluster/Script.java
11083 11084 e13399:formaeif:2008-03-27 12:15:23 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas/assemblees/jonas-configuration/src/main/resources/conf/jonas.properties
11084 11085 e13396:formaeif:2008-03-27 11:53:27 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas_tests/examples/base/examples/sampleCluster2/newjc/build-jc.properties
11085 11086 e13396:formaeif:2008-03-27 11:53:27 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas_tests/examples/base/examples/sampleCluster2/newjc/build-jc.properties
11086 11087 e13396:formaeif:2008-03-27 11:53:27 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas_tests/examples/base/examples/sampleCluster2/newjc/build-jc.properties
11087 11088 e13396:formaeif:2008-03-27 11:53:27 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas_tests/examples/base/examples/sampleCluster2/newjc/build-jc.properties
11088 11089 e13396:formaeif:2008-03-27 11:53:27 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas_tests/examples/base/examples/sampleCluster2/newjc/build-jc.properties
11089 11090 e13396:formaeif:2008-03-27 11:53:27 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas_tests/examples/base/examples/sampleCluster2/newjc/build-jc.properties
11090 11091 e13396:formaeif:2008-03-27 11:53:27 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas_tests/examples/base/examples/sampleCluster2/newjc/build-jc.properties
11091 11092 e13396:formaeif:2008-03-27 11:53:27 -0400 (Thu, 27 Mar 2008) |1:R:/jonas/trunk/jonas_tests/examples/base/examples/sampleCluster2/newjc/build-jc.properties
```



Example: Expertise Browser

- Obtain and present relationships between code and people and organizations based on Experience Atoms (EAs) shared between code unit(CU) and a person/org
- User interface
 - Linked view paradigm (link by EAs)
 - Code, developer, organization, and detail views
 - Choosing CU shows people/orgs that are related
 - Selecting a person/org shows the fraction of work done on code modules and the persons contact info
 - Height – $\sqrt{\text{number of deltas}}$
 - Width – number of developers making changes



Trustie expertise: who does what?

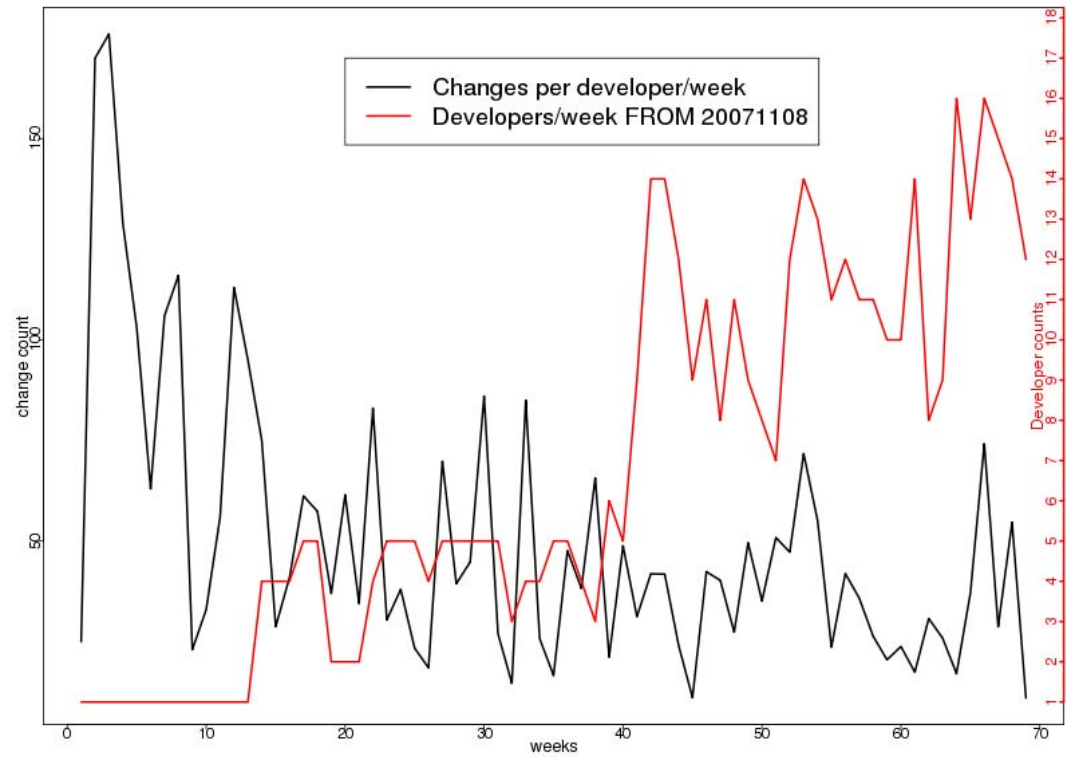
The screenshot shows the 'Who does what?' application interface. It is divided into several sections:

- Releases:** A list of release identifiers: unspec, CFBK64, CFBK97, CFBK46, CFBK51, CFBK61, CFBK91, CFBK85.
- Developer:** A list of developer names: liu_hao, yang zhen, li zhi, bian gjie, (没有作者信息), yuan zhengwen, wang ewu, han lun, deng xai, li shuqiang, wu yaoqian, li li, liang pfei, ren cje, xie ihui, cui liang, lian wuhai, zhang xbing.
- Module:** A tree view of modules including: trustie, com.cvicse.continuum.rap, forum, WebRoot, com.cvicse.im, com.cvicse.trustie.cms2, com.cvicse.trustie, com.cvicse, main, scarab, WebContent, com.cvicse.xplanner.rap, com.cvicse.xwiki.rap, xplanner, gamesproject, com.cvicse.subversion, com.cvicse.trustie.cms, statsvn_cvs, wiki1.4, xwiki, com.cvicse.im.rcp, com.cvicse.trustie.sysmanager, com.cvicse.xplanner.ui, com.cvicse.forum.rap, org.apache.maven.continuum, com.cvicse.trustie.project, com.cvicse.xwiki, continuum, maillist.
- Statistics for liu_hao:**

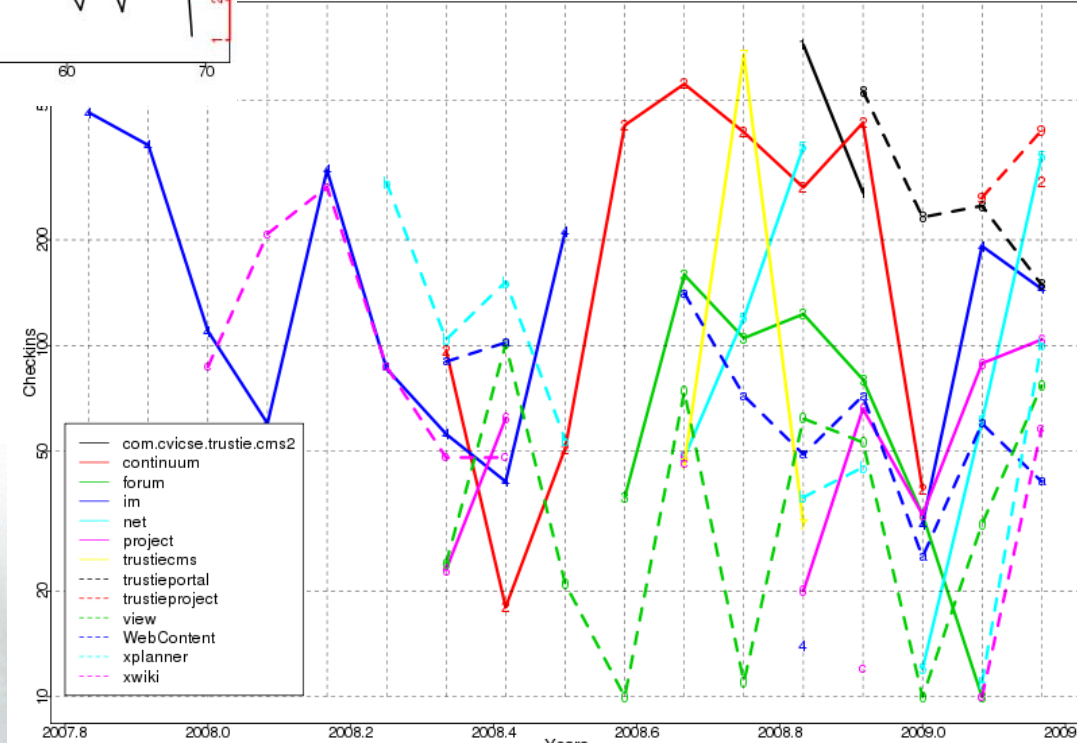
Number of Delta:	29313
Number of Fixes:	29313
Number of Developers:	18
Number of Fixers:	8

The application window title is 'Who does what?'. The Windows taskbar at the bottom shows the system tray with the time 11:25 and various icons.

Trustie: cost and output



Trustie month checkins by module



Summary: A perspective on middleware

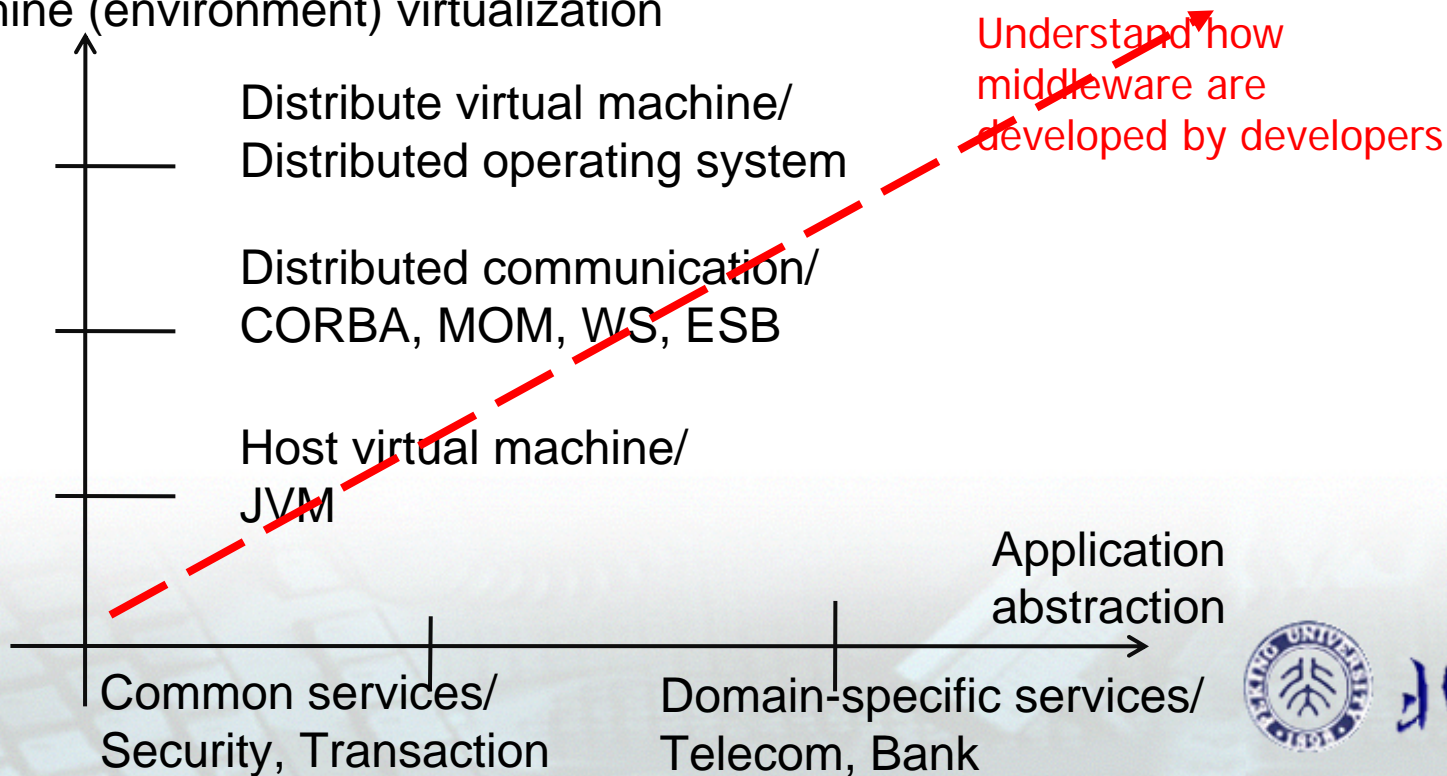
- Evolution principle

- Separation of concern, Abstraction of commonality

- Questions:

- What to abstract? How to construct? How to manage complexity?

Machine (environment) virtualization



北京大學



Summary: what's next?

- What commonalities can be abstracted next step?
 - Abstraction of reuse layer
- Can we build a general model for all the middleware, and customize the necessities to fit the applications when needed?
 - Application server's flexibility and customization
 - Middleware difference
- Can we build customizable, configurable, and flexible middleware frameworks for inherently heterogeneous environments?
 - Application server's adaptive management
 - Resource/environment difference
- Can we understand better how people develop middleware better?

