

# Trends in the Java Community

as interpreted by Callista Enterprise  
Johan Eltes och Mikael Hammarlund

# What drive trends?

Examples

Problems	Needs	Innovations
Development with EJB is complex and time-consuming	Better mechanisms for working with EJB or alternatives	Stay tuned....!
Too many documents and too much formalities in development	Good, light-weight methods for small to mid-size projects	
Integration testing too late, too difficult	Tools and techniques for early integration testing	
HTML user interfaces are not responsive, slow, and have limited functionality	Responsive client applications with more functionality	
Difficult to share services between applications	Standards and architectures for sharing services	

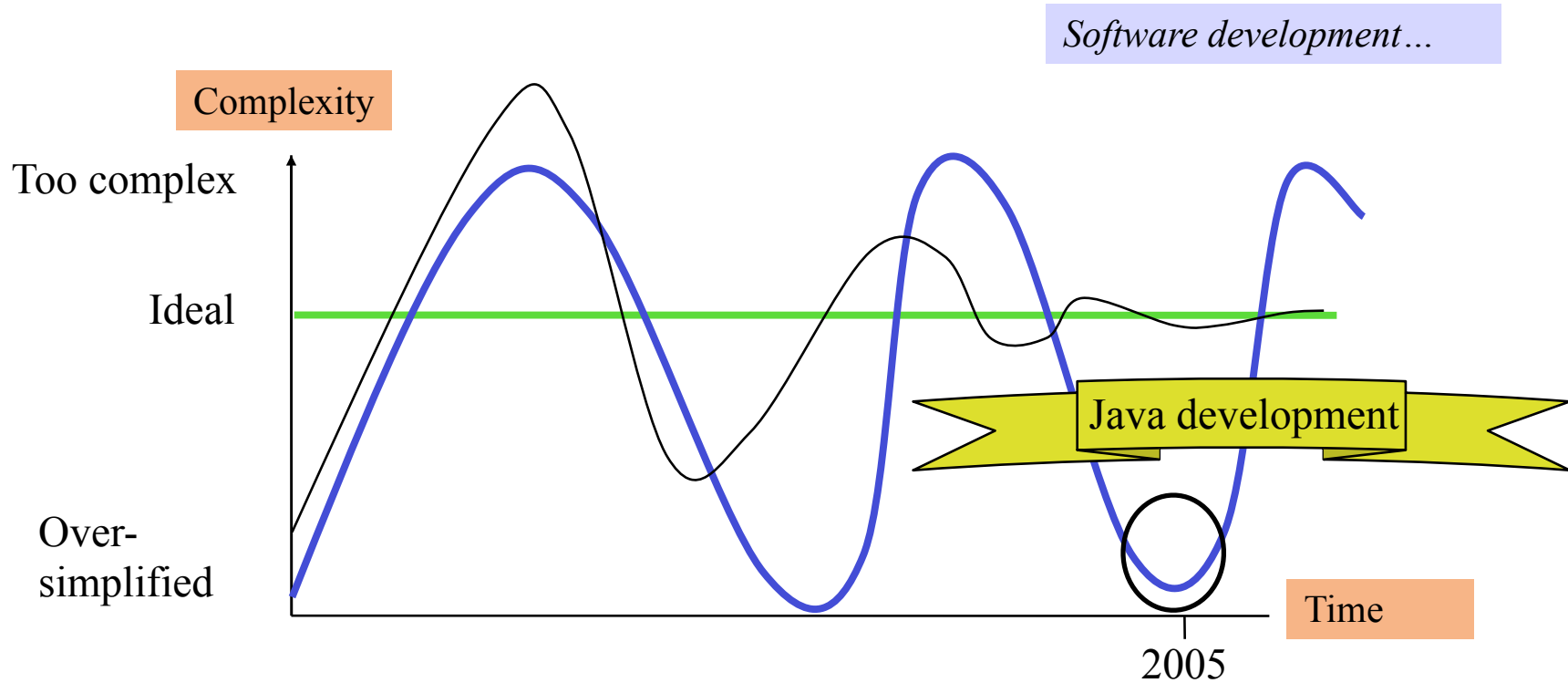
# Trends

- *Architecture trends*
  - "Elephant on Diet"
- *Software Development Processes*
  - "The flying elephant"
- *Development tools*
  - "Notepad is back"
- *User Interfaces*
  - "The Return of the Users"
- *Open Source*
  - "R&D For free"

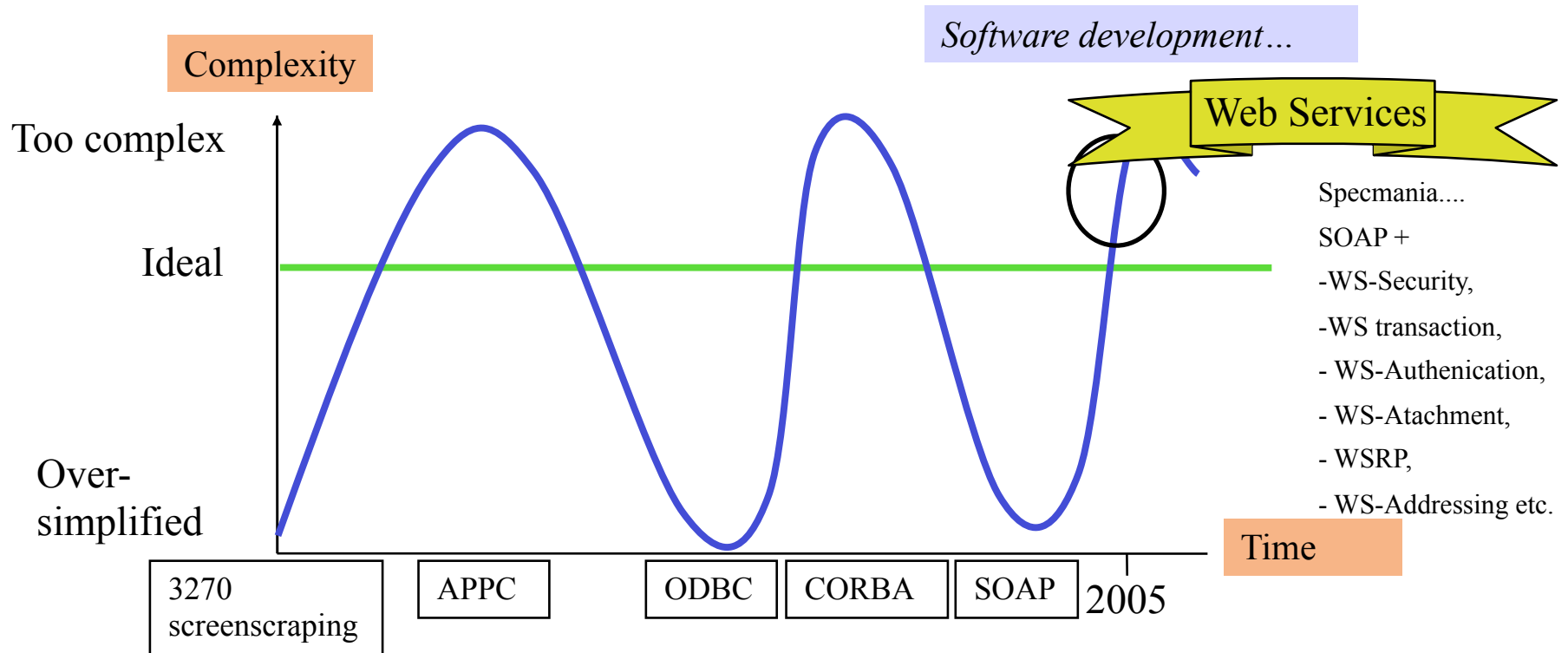
## ***Architecture Trend - Elephant on Diet***

Who is the Elephant?

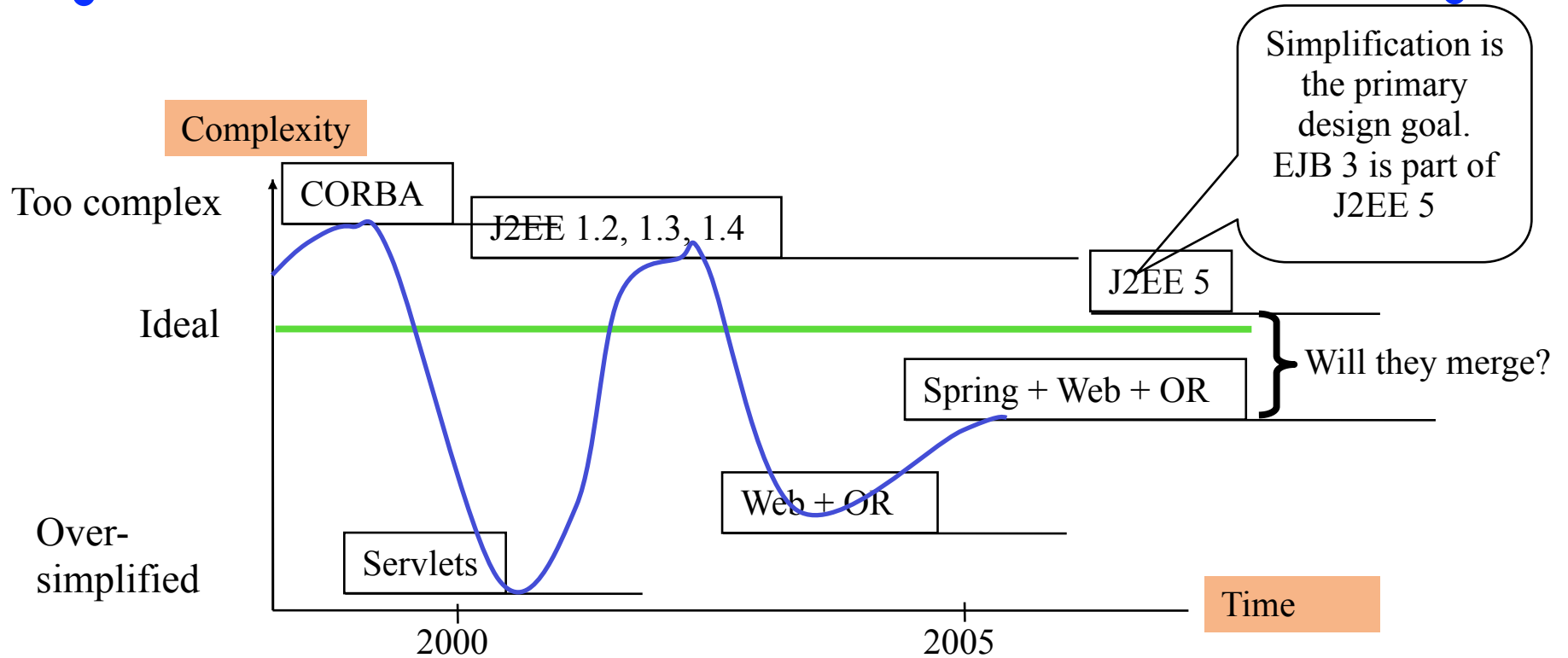
# Technology evolution...



# Technology evolution...



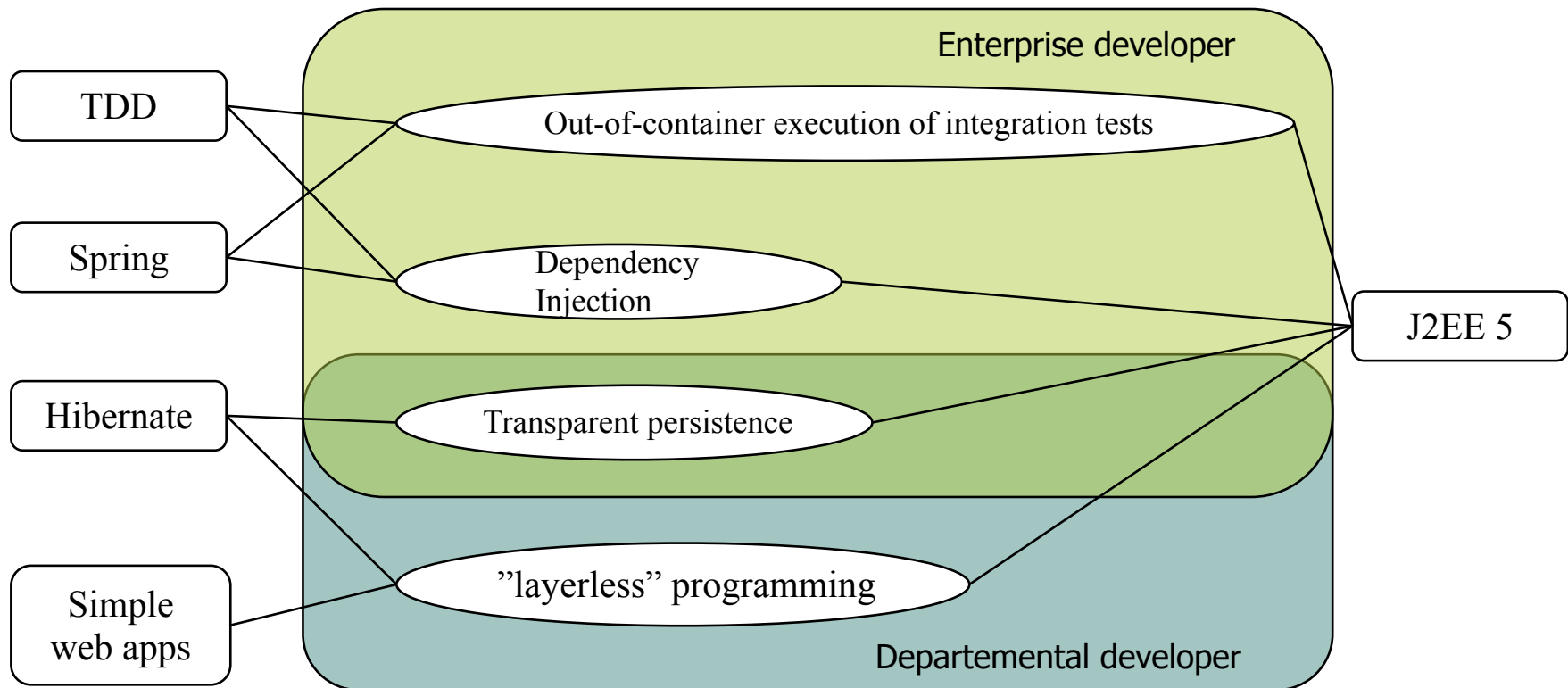
# Java Development



Context: *Enterprise Software Projects*

# Will J2EE 5 succeed?

## Major influences on J2EE 5

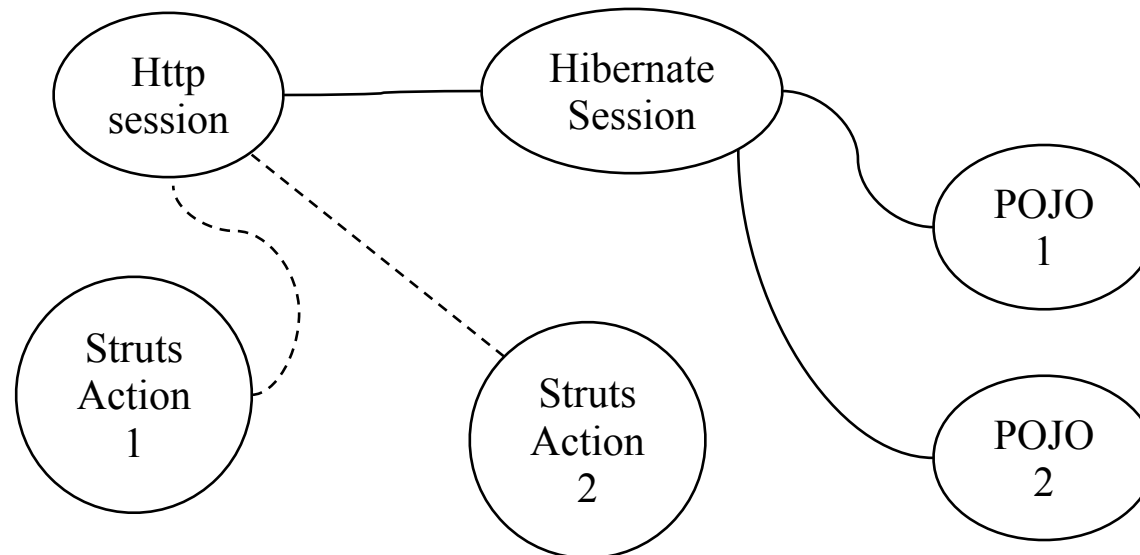


## Will J2EE 5 succeed...

- Two categories of developers need to be supported:
  - *Departemental developer*
  - *Enterprise developer*
- Out-of-container execution needs to be supported
  - *EJB 3 persistence solution must be usable outside of container*
  - *EJB 3.0 persistence engine usable in stand-alone mode for IDE-execution of integration tests*
  - *Dependency injection*
- Pluggable persistence frameworks
  - *Combine App server from vendor X with peristence solution from vendor Y*
  - *Requires portable access to TransactionManager. Currently not possible.*

## Will J2EE 5 succeed...

- Support "layerless" programming
  - *Made popular by Hibernate*
  - *Makes web-app database programming extremely efficient*
- What is "layerless" programming with Hibernate?



## Will J2EE 5 succeed...

- Most of this is technically possible
- But how to make money if the code runs well also without an EJB container?
  - *This is the toughest challenge for the spec writers.*

# Trends

- *Architecture trends*
  - "Elephant on Diet"
- *Software Development Processes*
  - "The flying elephant"
- *Development tools*
  - "Notepad is back"
- *User Interfaces*
  - "The Return of the Users"
- *Open Source*
  - "R&D For free"

# Software Development Processes – The Flying Elephant

This is about lightweight development processes that focus on what's really important.



## Agile

- ❑ Iterative, incremental development has proved to efficient
- ❑ but, methods like RUP are to complex and often require too much documentation and administration for small to mid-size projects
- ❑ In 2001 discussions among agile methods resulted in the Agile Alliance

# Agile

- What is Agile Software Development?
  - *Manifesto for software development, 2001*

**Individuals and interactions** over processes and tools  
**Working software** over comprehensive documentation  
**Customer collaboration** over contract negotiation  
**Responding to change** over following a plan

“That is, while there is value in the items on the right, we value the items on the left more”

## Agile Methods

- AD - Agile Database Techniques
- AM - Agile Modeling
- ASD - Adaptive Software Development
- Crystal
- FDD - Feature Driven Development
- DSDM - Dynamic Systems Development Method
- Lean Software Development
- Scrum
- TDD - Test-Driven Design
- XBreed
- XP - eXtreme Programming

## Agile Trends

- Agile methods are hot
  - *but not in Sweden - yet*
- Practices from Agile methods are taken into projects that use RUP etc
  - *Example, Test-driven development, continuous integration*

# Resources

- Agile Alliance : <http://www.agilealliance.org>

# Trends

- *Architecture trends*
  - "Elephant on Diet"
- *Software Development Processes*
  - "The flying elephant"
- *Development tools*
  - "Notepad is back"
- *User Interfaces*
  - "The Return of the Users"
- *Open Source*
  - "R&D For free"

## ***Development Tools Trend - Notepad is back***

This is about high-end  
IDEs becoming out of  
fashion

## ***Notepad is back - Why?***

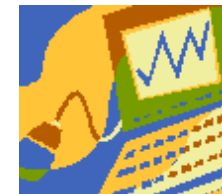
- ❑ Large, complex IDE:s require a lot of individual set-up.
- ❑ The set-ups are fragile. Too often, a set-up ends up in an inconsistent state, requiring the user to rework the full set-up.
- ❑ IDE-centric development usually delays integration until code stop, leading to manual, weakly documented and error-prone integration process.
- ❑ Delays in integration means less time for code reviews, design reviews, unit test reviews, writing unit tests....



*6 weeks development*

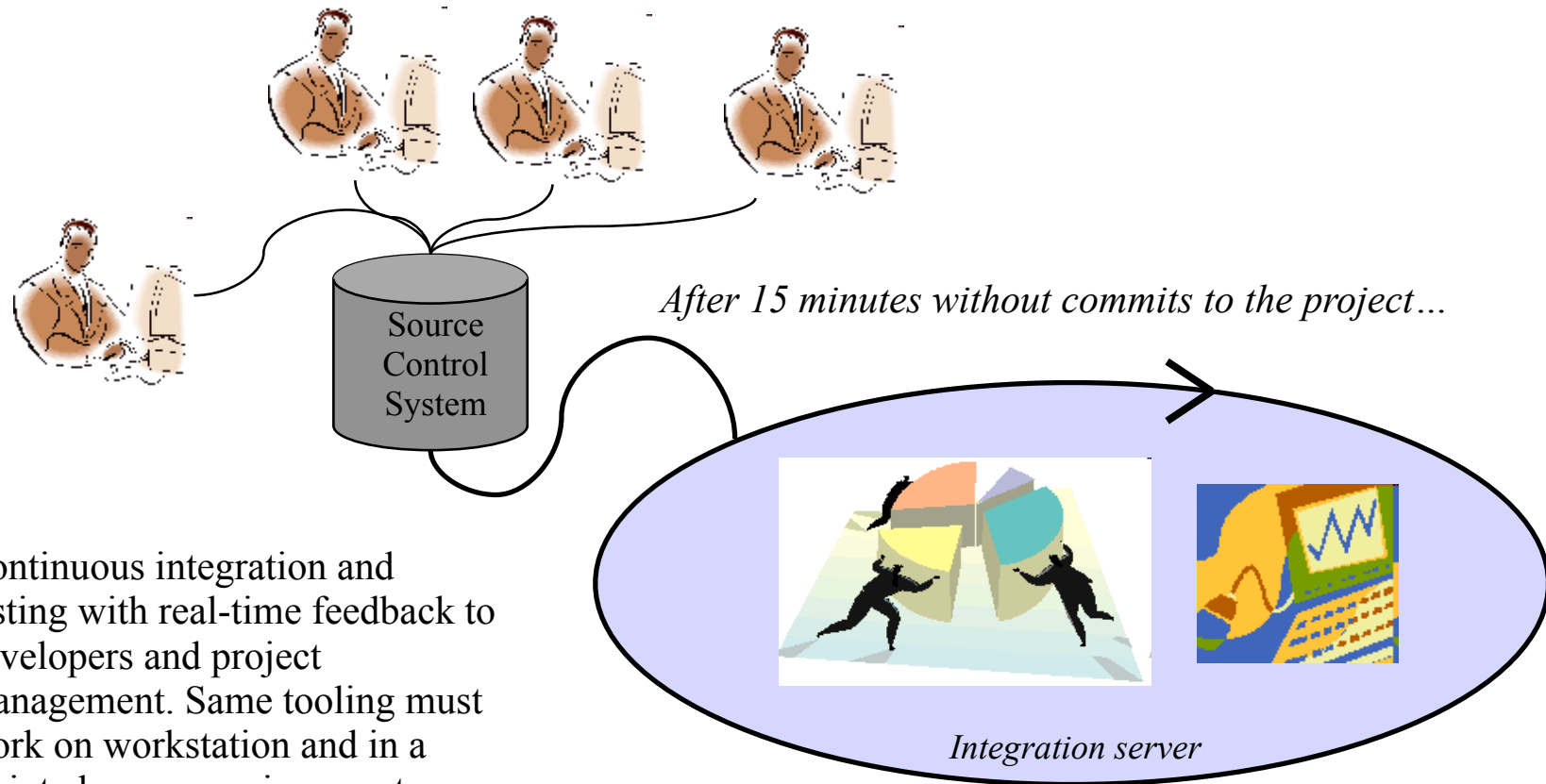


*1 week integration*



*3 weeks testing and bug-fixing*

# Continuous Integration



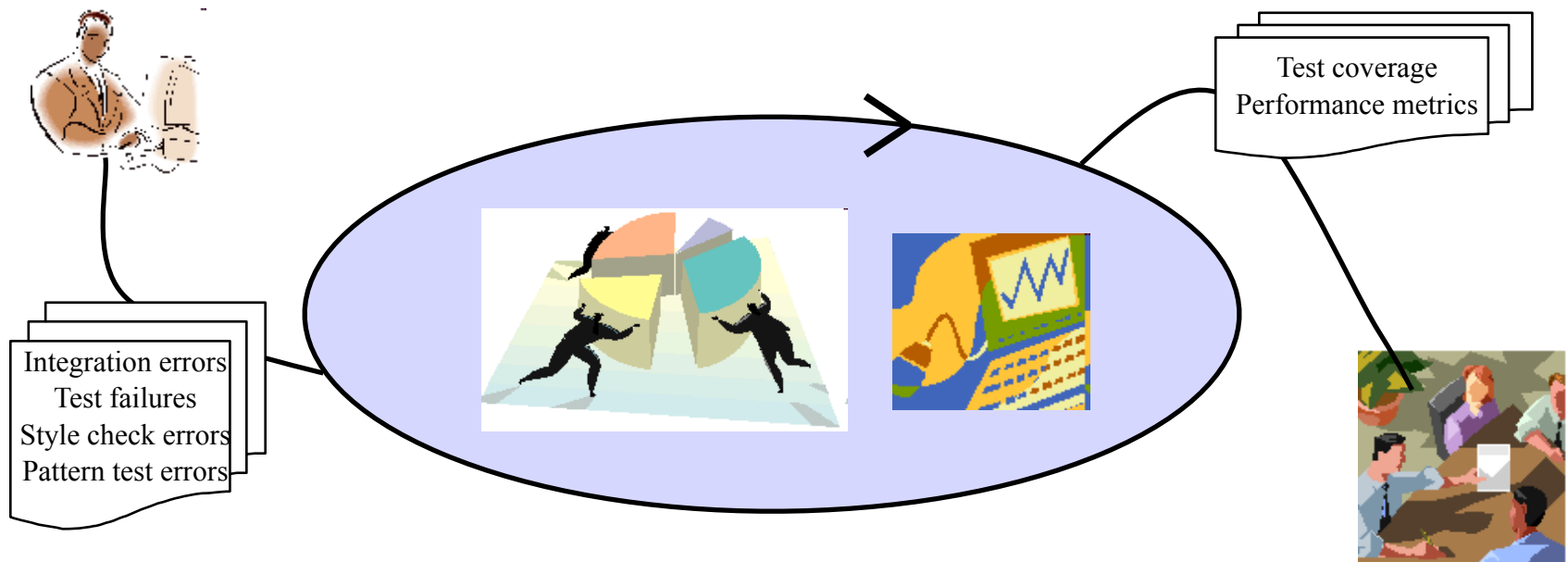
Continuous integration and testing with real-time feedback to developers and project management. Same tooling must work on workstation and in a scripted server environment.

*WSAD wizards are not applicable*

...

## CI Benefits

- ❑ All developers build their code exactly the same way
- ❑ The solution is continuously integrated. Each integration build gives immediate feedback:



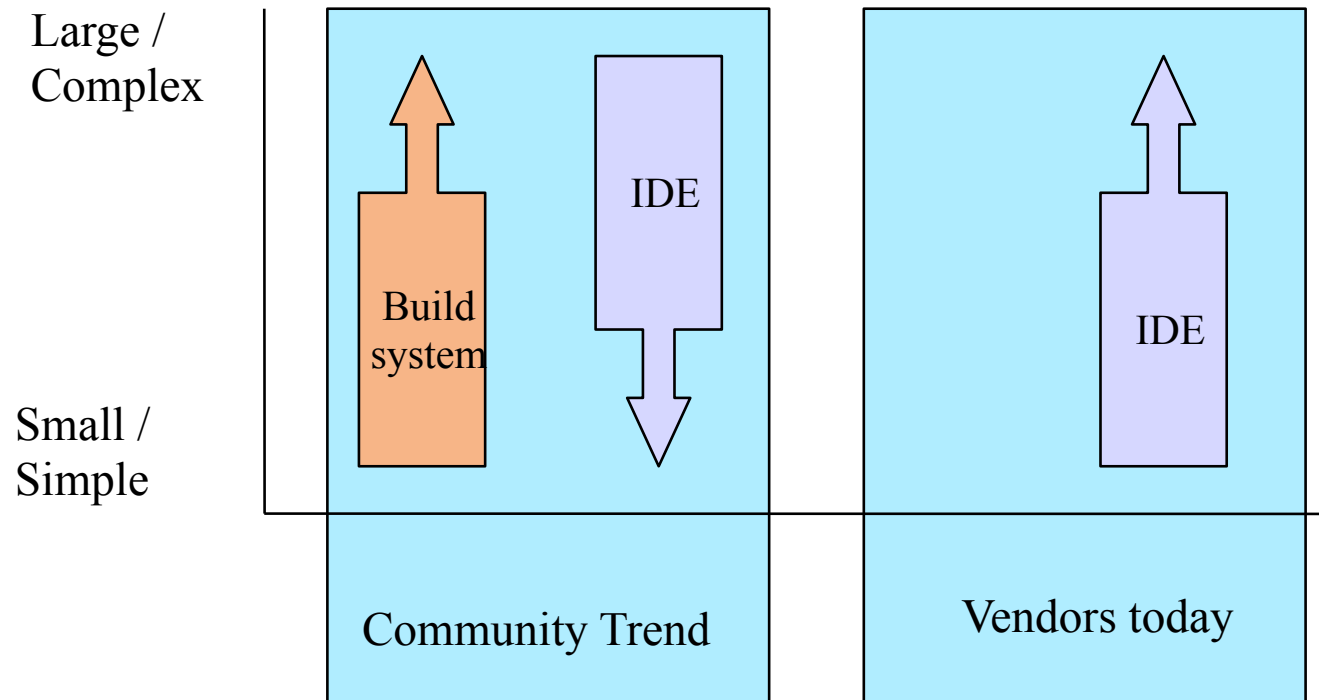
## CI Tooling

- Tooling for CI is mostly Java-based open source
- Tooling covers features also included in complex IDEs, in the following areas:
  - *Unit Test execution*
  - *Integration test execution, including test data set-up and tear-down*
  - *Code generators*
  - *Profiling*
  - *Performance analysis*
- The result of a build is reported as a web-based dashboard:
  - Test results
  - Test coverage
  - Performance metrics
  - Generated code documentation
  - Compliance to design- and coding guidelines

## CI consequences

- ❑ We are spending a lot of money on overly complex IDEs.
- ❑ Choice of IDE isn't strategic.
- ❑ The value is in continuous integration and test automation practices.
- ❑ Developers must have support with all build set-up - clear role split is essential

## Trend summary



*Corporate Standards prescribe WSAD (3000 \$).  
Community trend is to use Eclipse (0 \$),*

## Resources

- Lightweight IDEs
  - <http://www.eclipse.org>
  - <http://www.jetbrains.com/idea/>
- Continuous integration job control
  - *Cruise Control*
  - *Anthill*
- Enterprise build system
  - *Maven*
- Automated design guideline reviews
  - *Pattern testing based on AspectJ*

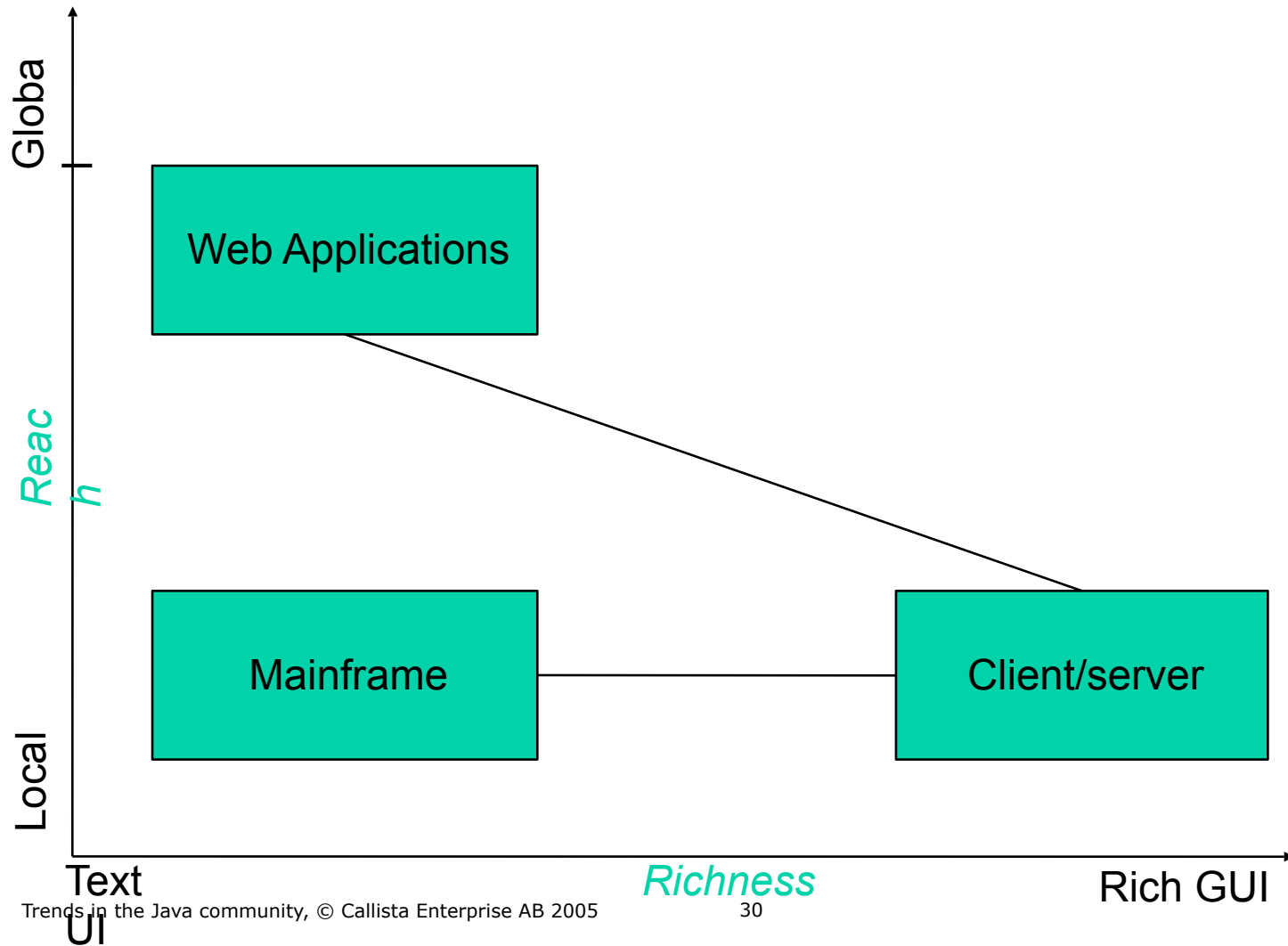
# Trends

- *Architecture trends*
  - "Elephant on Diet"
- *Software Development Processes*
  - "The flying elephant"
- *Development tools*
  - "Notepad is back"
- *User Interfaces*
  - "The Return of the Users"
- *Open Source*
  - "R&D For free"

## ***The Return Of The Users***

This is about business users raising their voices for responsive, supportive applications.

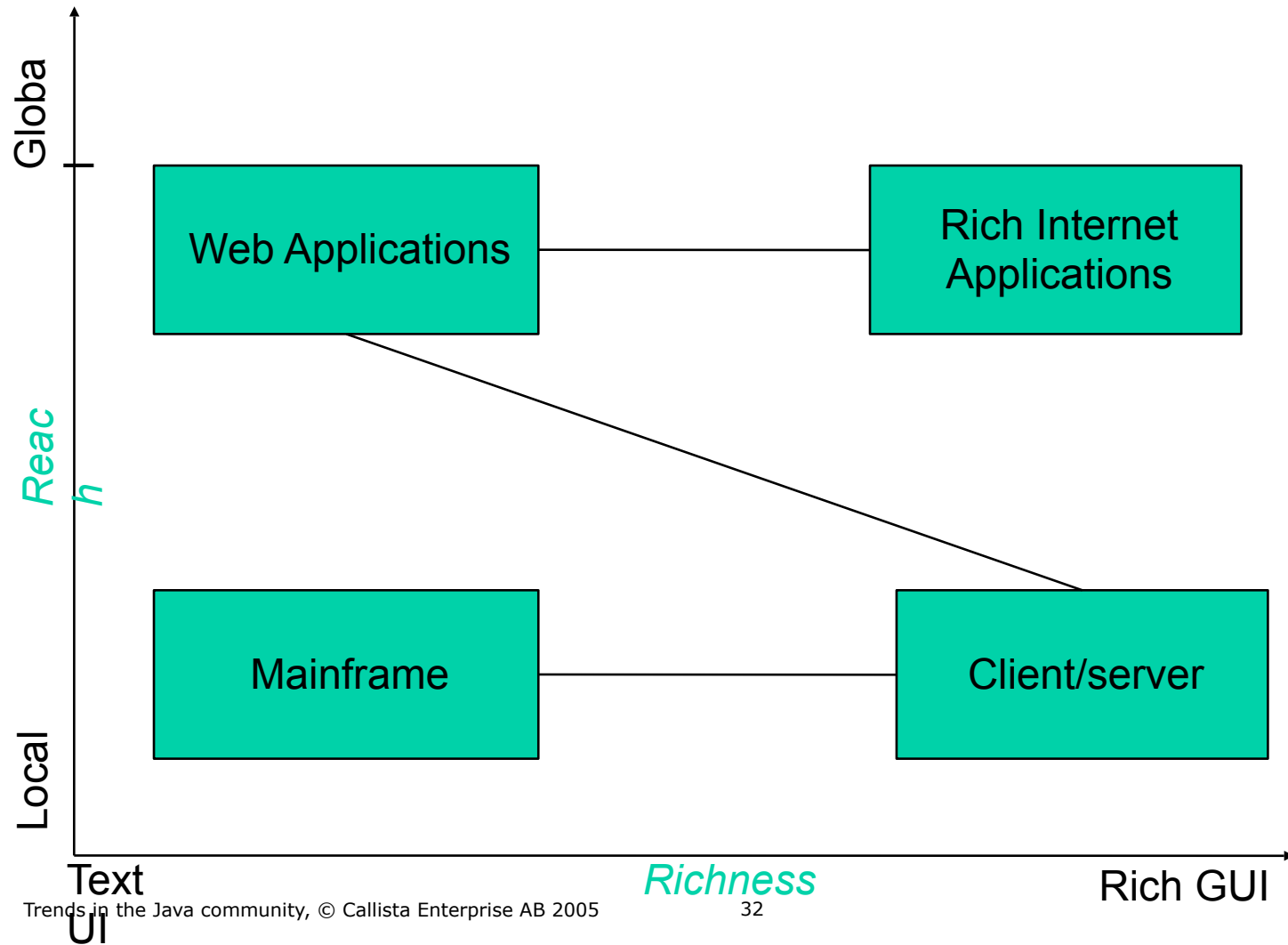
# User Experience Evolution



## Trends and Impact

Trend	Architectural Impact	Business Impact
J2EE and .Net in mainstream	Stable, flexible data access and business logic tiers	Chronic usability issues  Unrealized ROI
Service-oriented architectures emerge		
HTML limited for applications	Complicated, non-intuitive presentation tier	

# User Experience Evolution



# Rich Internet Applications

## □ Motivation

### □ *A need for:*

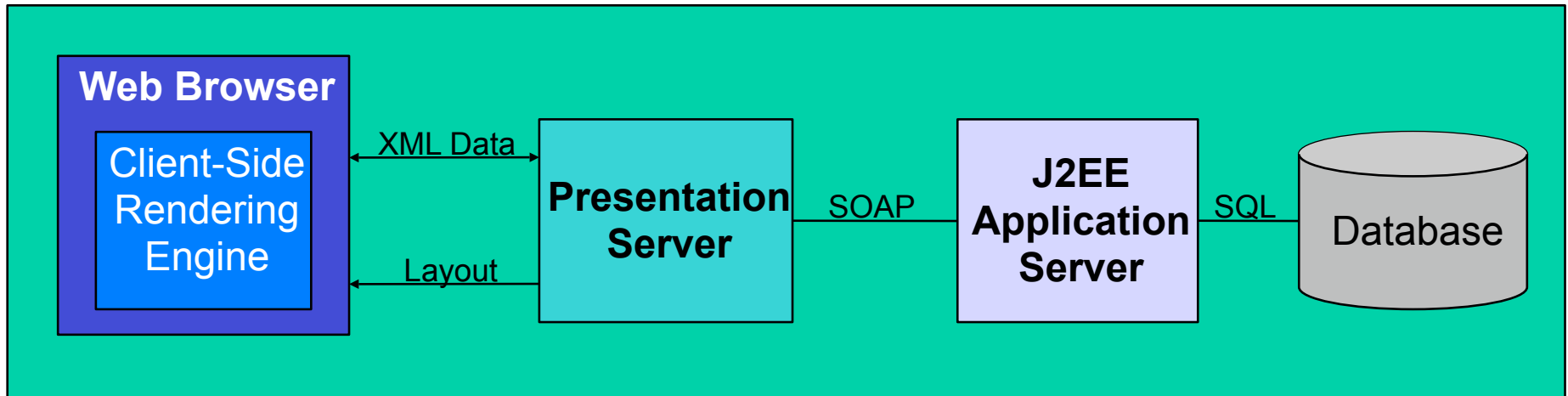
- more robust, responsive, and visually interesting applications than what can be achieved with HTML
- applications that can easily be deployed over the Internet
- applications that can be developed quickly

## The Return Of The User

Solutions to these problems are partly provided by

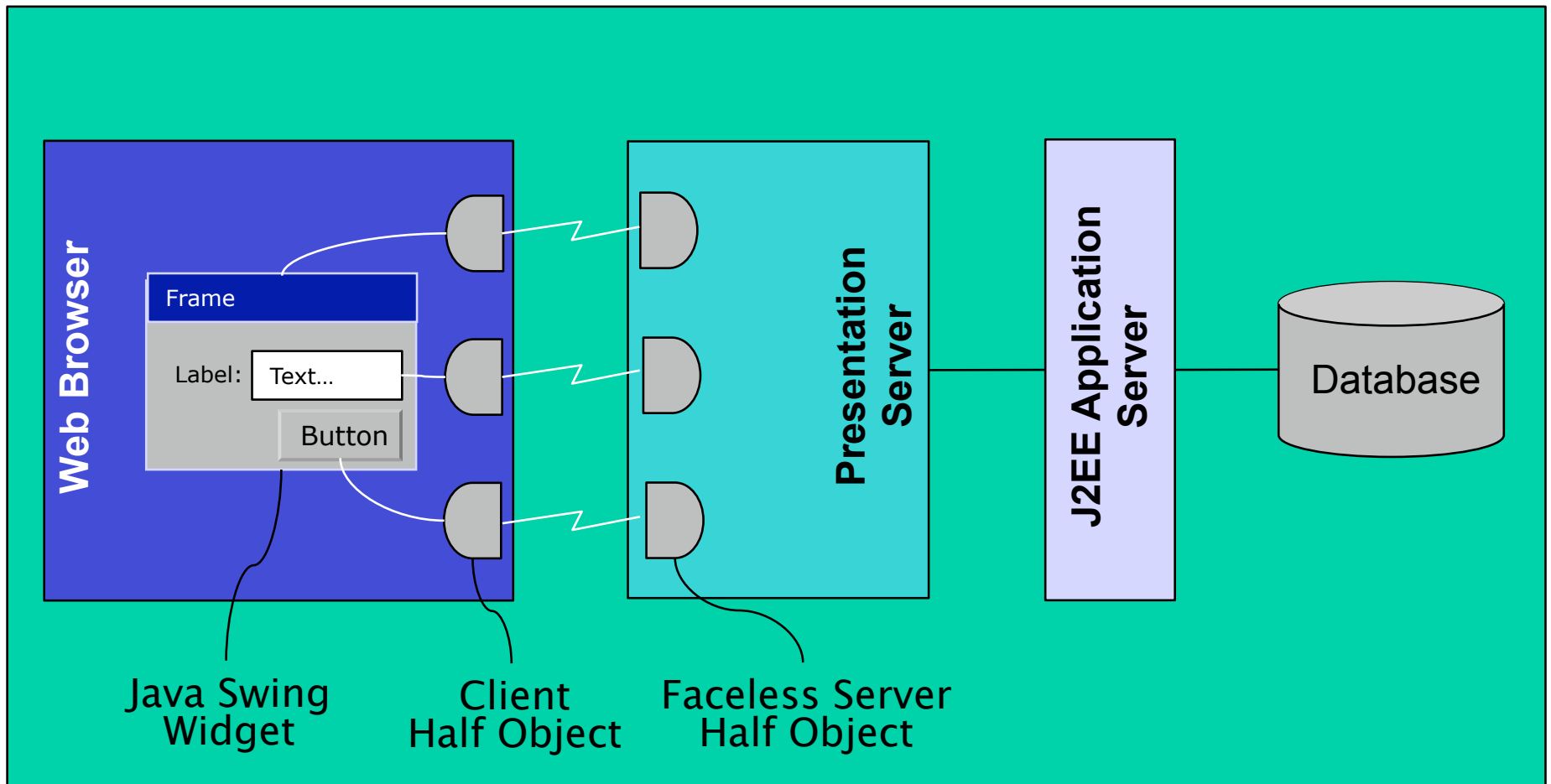
- *Struts*
- *Java Server Faces (JSF)*
- ....but primarily with
  - *Rich Internet Applications (RIA)*

## Typical RIA Architecture



- Two principal approaches
  - *XML+Flash*
    - Macromedia Flex is hot
  - *Swing-based*
    - typically started with Java Web Start
    - simplified and enhanced
    - half-object pattern

# Swing Half-Object Pattern



## JDNC

- Java Desktop Network Components
- Developed by Sun, but open source project
  - *inspired by other opens source projects and proprietary applications (Canoo ULC, wingS, WebOnSwing, Wi.Ser)*
- Much improved Swing, with extensions
- Swing extensions is to be part of Mustang-Java6, but able to run on “older” v1.42 and v5 JVMs.

## The future of RIA

- New technology, not mature
- ....but very promising
- Analysts have estimated that by 2006, more than one third of all applications will be developed using RIA.

## Resources

- *RIA*
  - Java Desktop Network Components – JDNC, <https://jdnc.dev.java.net/>
  - Macromedia Flex, <http://www.macromedia.com/software/flex/> (XML, Flash)
  - Laszlo, <http://www.openlaszlo.com> (XML, Flash, Open source)
  - Canoo UltraLiteClient, <http://www.canoo.com> (Swing)
  - Thinlet: <http://www.thinlet.com> (XML, free)
  - Altio, <http://www.altio.com> (XML, Flash)
  - Droplets. <http://www.droplets.com>
  - InsiTech XTT, <http://www.insitechgroup.com/> (XML)
  - wingS, <http://wings.mercatis.de> (Swing-based, open source)
  - WebOnSwing, <http://webonswing.sourceforge.net> (Swing)
  - Wi.Ser, <http://wiser.dev.java.net> (Swing)
  - AppProjector, <http://www.asperon.com>
  - Classic Blend, [http://www.appliedreasoning.com/products\\_what\\_is\\_Classic\\_Blend.htm](http://www.appliedreasoning.com/products_what_is_Classic_Blend.htm)
  - RSWT, <http://rswt.sourceforge.net>
  - IAB Studio, [www.iabstudio.com](http://www.iabstudio.com) / [www.worcsnet.com](http://www.worcsnet.com)
- *Struts*
  - Struts 2.0/Shale, <http://wiki.apache.org/struts/StrutsShale>

# Trends

- *Architecture trends*
  - “Elephant on Diet”
- *Processes*
  - “Dumbo – the flying elephant”
- *Development tools*
  - “Notepad is back”
- *User Interfaces*
  - “The Return of the Users”
- *Open Source*
  - “R&D For free”

## ***Open Source Trends: R&D For Free***

- Java evolutions is supposed to be driven by Suns JCP
  - *Java Community Process*
  - *JSRs (Java Specification Requests) drive the JCP*
- Classic drivers for submitting a JSR
  - *Vendors to make THEIR api become the standard*
  - *Keep up with Microsoft*
- JSR downsides
  - *Politics*
  - *Often very slow process*

## Open Source as R&D for JCP

- The trend:
  - *Create JSRs from successful Open Source projects*
- (Successful) Open Source projects are different from JCP
  - *Starts small*
  - *Live in reality*
  - *Matures fast*
  - *Dies unless successful (this is important!)*
- JSRs created from or inspired by Open Source projects gets a jump start
- Downside...
  - *There must be a value for 100 000 developers to switch from open source solution to JSR counterpart*
  - *Compare to Log4J vs JDK 1.4 Logger.*

## Examples of "Open Source JSRs"

- JDK 1.4 Logger
  - *Log4J*
- J2EE 5 (not a JSR in itself, but still JCP)
  - *Dependency injection from IoC containers (Spring, Hivemind...)*
- EJB 3
  - *Transparent persistence from Hibernate (Yes, NextSTEP EOF was first, but not OpenSource...)*
  - *Hibernate founder is recruited to EJB 3 JCP expert group.*
- Java Server Faces (JSF)
  - *Set up as a "better Struts" framework*
- Annotations
  - *The annotations JSR picked up from the success fo Xdoclet (and got some additional fuel through DotNet Attributes)*

## ***Sun takes the next step...***

- Starts Open Source projects rather than JSRs, for new ideas to mature
  - *Java Desktop Network Components (JDNC) RIA architecture is started as Open Source by Sun*
- This may be a trend...

## What can we learn?

- Companies will get more conscious about Open Source projects.
  - *Knowledge about the status of various open source projects becomes strategic*
  - *Very powerful compared to inhouse framework development*
- Large IT organisations will recognize the need to maintain open source products
  - *Active participation is an excellent way to secure inhouse maintenance*
- Large IT organisations will adopt Open Source processes and practice for global, distributed projects
  - *For evolution and maintenance of internal frameworks.*

## Q&A

- *Architecture trends*
  - "Elephant on Diet"
- *Software Development Processes*
  - "The flying elephant"
- *Development tools*
  - "Notepad is back"
- *User Interfaces*
  - "The Return of the Users"
- *Open Source*
  - "R&D For free"



# Summary

Examples

Problems	Needs	Innovations
Development with EJB is complex and time-consuming	Better mechanisms for working with EJB or alternatives	Spring Hibernate EJB 3.0
Too many documents and too much formalities in development	Good, light-weight methods for small to mid-size projects	Agile
Integration testing too late, too difficult	Tools and techniques for early integration testing	Less focus on IDEs Continuous integration
HTML user interfaces are not responsive, slow, and have limited functionality	Responsive client applications with more functionality	Rich Internet Applications
Difficult to share services between applications	Standards and architectures for sharing services	SOA, WebServices