JAVA DAYS ´99



JavaBeans™ Component Architecture

Onno Kluyt
Sr. Product Manager
Sun Microsystems, Inc

Sun microsystems

In This Session...

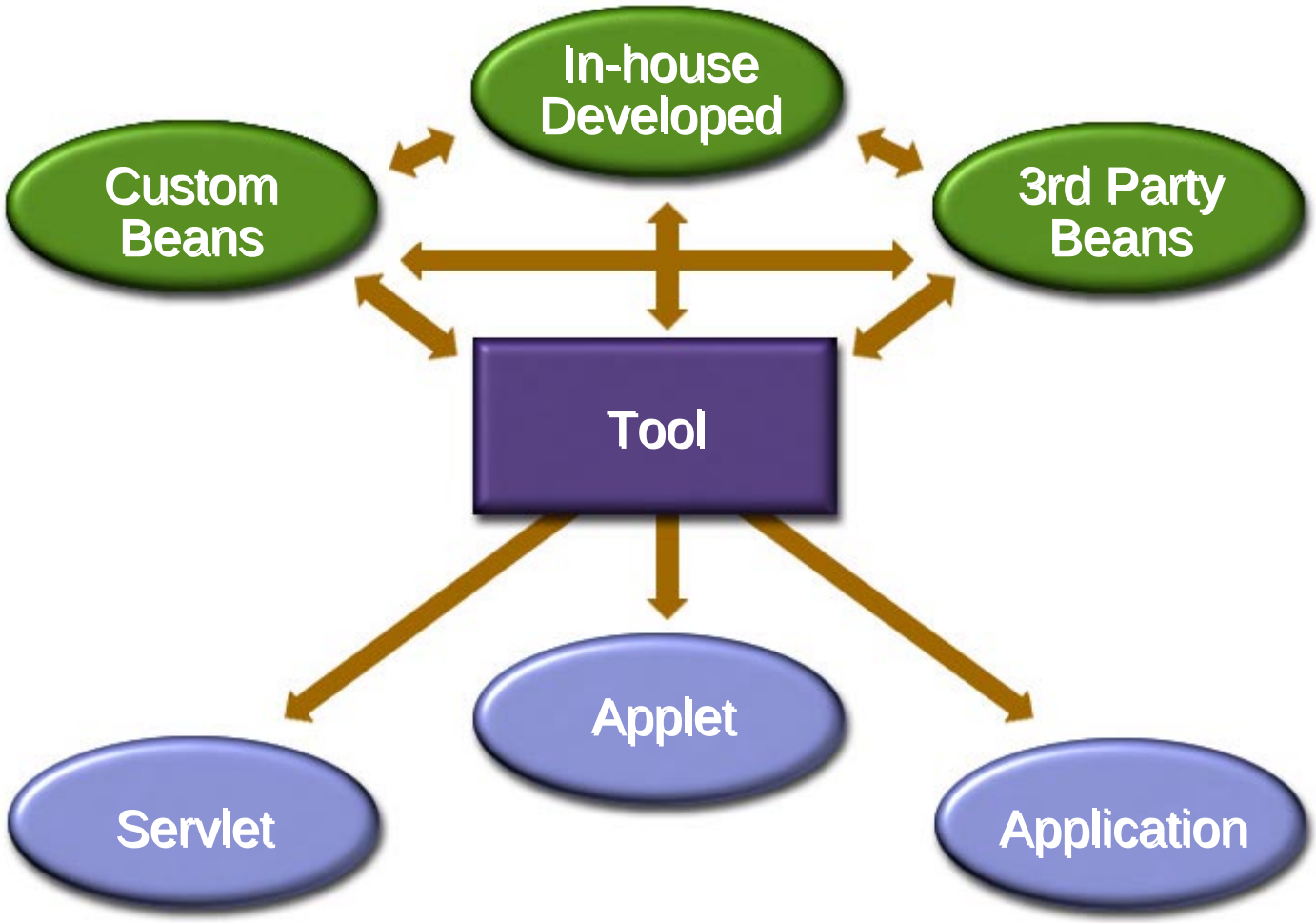
- JavaBeans™ component architecture overview
- BeanContext API in detail
- Q&A



© 1999 Sun Microsystems GmbH

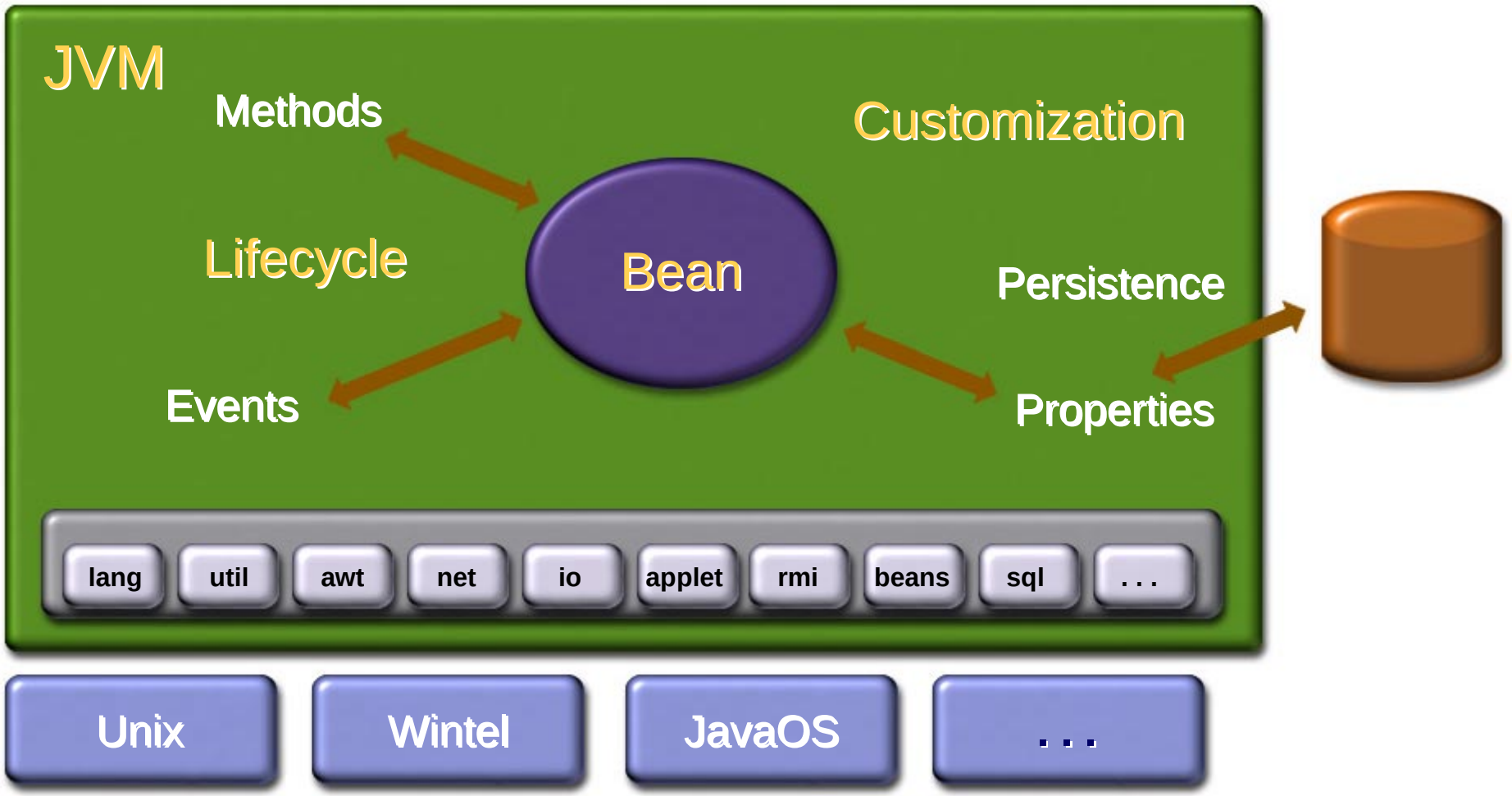
Sun microsystems

What the JavaBeans™ Technology Enables



© 1999 Sun Microsystems GmbH

What Is a Bean



© 1999 Sun Microsystems GmbH

Milestones

- Delivered in the Java™ 2 Platform:
  - Extensible Runtime Containment and Services Protocol
  - Drag & Drop
- JavaBeans Activation Framework 1.0.1
- InfoBus 1.2
- Beans Development Kit (BDK) 1.1

© 1999 Sun Microsystems GmbH

JAVA DAYS ´99



Extensible Runtime Containment and Services Protocol

Sun microsystems

Overview



- Support for “container” type applications
  - Frameworks
  - Browsers
  - Word processors
- Support for providing services to components
  - Error logging
  - Document printing interface
  - Spelling checker

© 1999 Sun Microsystems GmbH

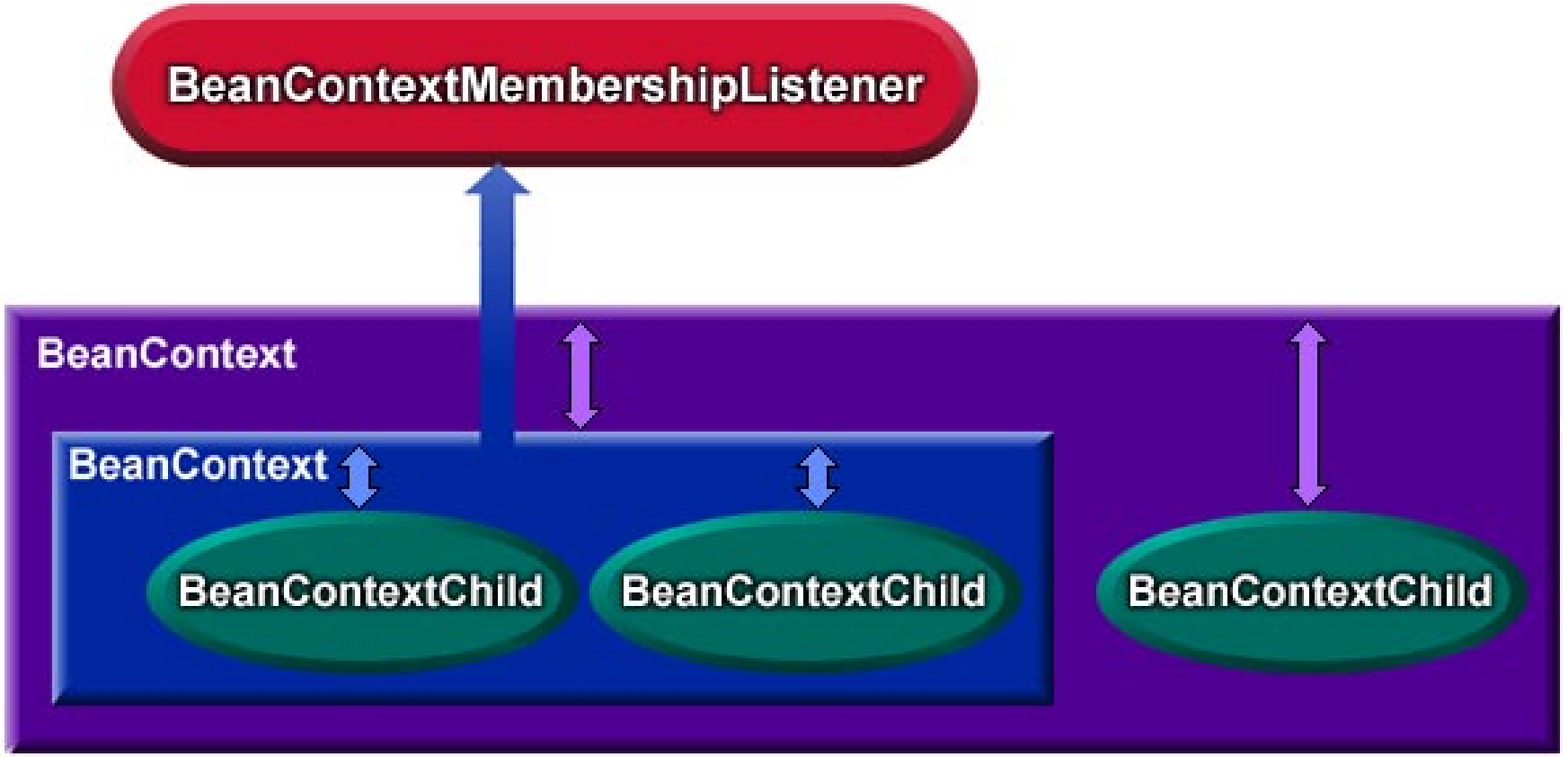
Sun microsystems

Goals

- Standard logical containment hierarchy for components
- Standard mechanism for components to obtain services from their environment
- Improve existing support for propagating execution environment to components
  - DesignMode
  - Visibility
  - Support for Beans that are also Applets

© 1999 Sun Microsystems GmbH

Containment Model



© 1999 Sun Microsystems GmbH

BeanContext

- Logical hierarchy based on Collection API
- Resource support via getResource() and getResourceAsStream()
- Distinguish between design mode and runtime via DesignMode interface
- Support for visibility via visibility interface

© 1999 Sun Microsystems GmbH

BeanContextProxy



- If extending java.awt.Component
  - Delegate to BeanContextProxy
  - Allows one BeanContext to support multiple containers
- If implementing BeanContext interface
  - Delegate to BeanContextContainerProxy

© 1999 Sun Microsystems GmbH

Sun microsystems

BeanContextChild



- Simple Beans function normally
- Interface is necessary if Bean will request services
- Exposes component property for nesting BeanContext
- Nesting/unnesting operation is vetoable

© 1999 Sun Microsystems GmbH

Sun microsystems

Changes in Containment

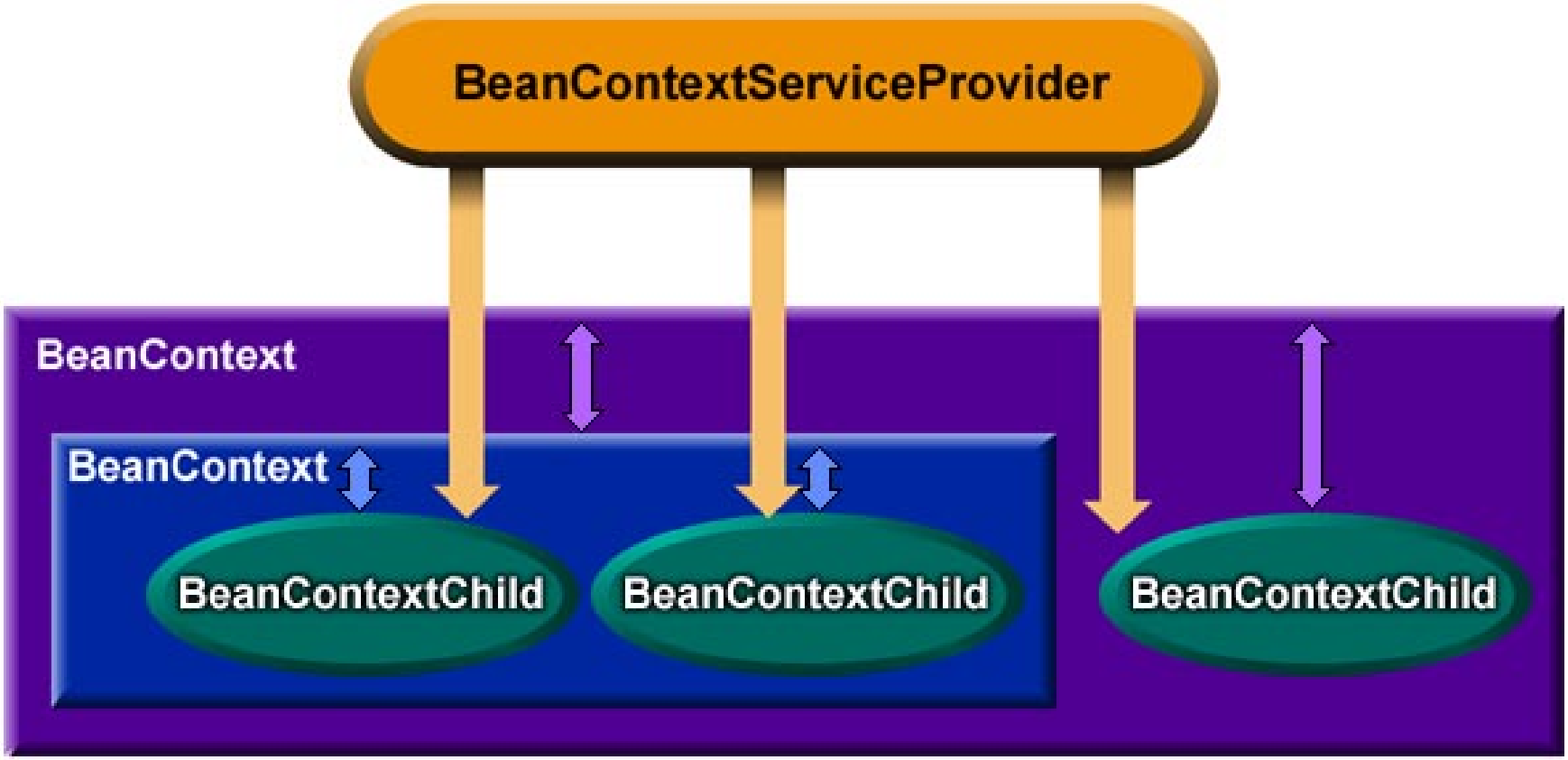


- **`BeanContextMembershipListener`**
  - **`childrenAdded()`**
  - **`childrenRemoved()`**
- **`BeanContextMembershipEvent`**
  - Encapsulates child/children added or removed
  - Event propagation through hierarchy via **`get/setPropagatedFrom()`**

© 1999 Sun Microsystems GmbH

Sun microsystems

Services Model



© 1999 Sun Microsystems GmbH

BeanContextServices



- Register services with this BeanContext
  - addService()
  - revokeService()
- Mechanism for child components to obtain or release service implementations
  - hasService()
  - getService()
  - releaseService()

© 1999 Sun Microsystems GmbH

Sun microsystems

BeanContextServiceProvider



- BeanContext forwards requests to associated BeanContextServiceProvider
- Factory for requested service
- Supports requests from multiple sources

© 1999 Sun Microsystems GmbH

Sun microsystems

Change in Services Events



`BeanContextServiceAvailableEvent`

`BeanContextServiceRevokedEvent`

- Services can be “forcibly revoked”
- Propagation through hierarchy

© 1999 Sun Microsystems GmbH

Sun microsystems

Change in Service Listeners



- **`BeanContextServiceRevokedListener`**
  - Components automatically registered when service reference is successfully obtained
- **`BeanContextServicesListener`**
  - Listener for both service revoked and service available events

© 1999 Sun Microsystems GmbH

Sun microsystems

Getting Started



- Compliant, extensible support implementations
  - **`BeanContextSupport`**
  - **`BeanContextChildSupport`**
  - **`BeanContextServicesSupport`**
- BeanBox sample code

© 1999 Sun Microsystems GmbH

Sun microsystems

BeanBox




```
public class BeanBox extends Panel implements BeanContextProxy
{
    // Instantiate BeanContext proxy
    BeanContextServicesSupport services =
                new BeanContextServicesSupport();
    // …
    // BeanContextProxy interface method for exposing proxy
    public BeanContext getBeanContext()
    {
        return services;
    }
    // …
}
```


© 1999 Sun Microsystems GmbH

Sun microsystems

BeanBox (Cont.)




```
// Register the Method Tracing service with the BeanBox
services.addService( MethodTracer.class,
        // Service provider for Method Tracer service
        new BeanContextServiceProvider() {
            public Object getService(BeanContextServices bcs,
                                     Object requestor,
                                     Class serviceClass,
                                     Object serviceSelector) {
                // …
            }
        });
// …
```


© 1999 Sun Microsystems GmbH

Sun microsystems

Juggler




```
// Juggler Bean
public class Juggler extends Applet implements Runnable,
   BeanContextProxy, BeanContextServicesListener,
   PropertyChangeListener, DesignMode {
    // …

    // Instantiate BeanContextChild proxy
    private BeanContextChildSupport child =
       new BeanContextChildSupport() {
        // …
```


© 1999 Sun Microsystems GmbH

Sun microsystems

Juggler (Cont.)




```
// Invoked by BeanBox's BeanContextProxy when it adds new Juggler instance.
protected void initializeBeanContextResources() {
    //…
   // If service is available, get a reference to it, else listen for it.
   if (context.hasService( MethodTracer.class)) {
        mtService = (MethodTracer)context.getService( … );
   } else {
        context.addBeanContextServicesListener( … );
   }
   context.addPropertyChangeListener( "designMode", Juggler.this);
   // …
}
```


© 1999 Sun Microsystems GmbH

Sun microsystems

Juggler (Cont.)




```
// Disable method tracing for runtime. Enable for design mode if debugging.
public void setDesignTime( boolean mode) {
   // …
   }

// Listener method for “designMode” property change event.
public void propertyChange( PropertyChangeEvent evt) {
   // …
    setDesignTime( mode );
   }
```


© 1999 Sun Microsystems GmbH

Sun microsystems

JAVA DAYS ´99

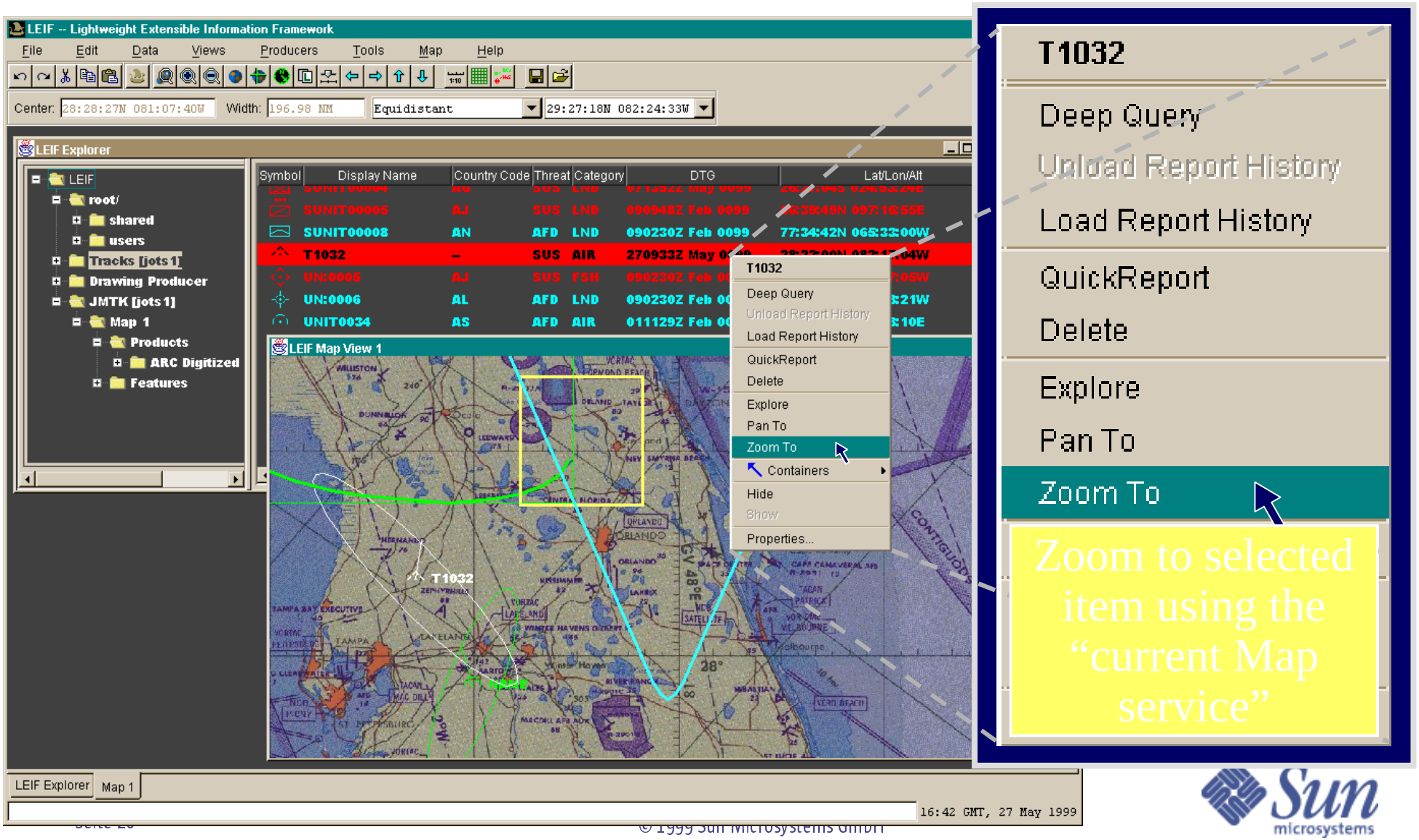


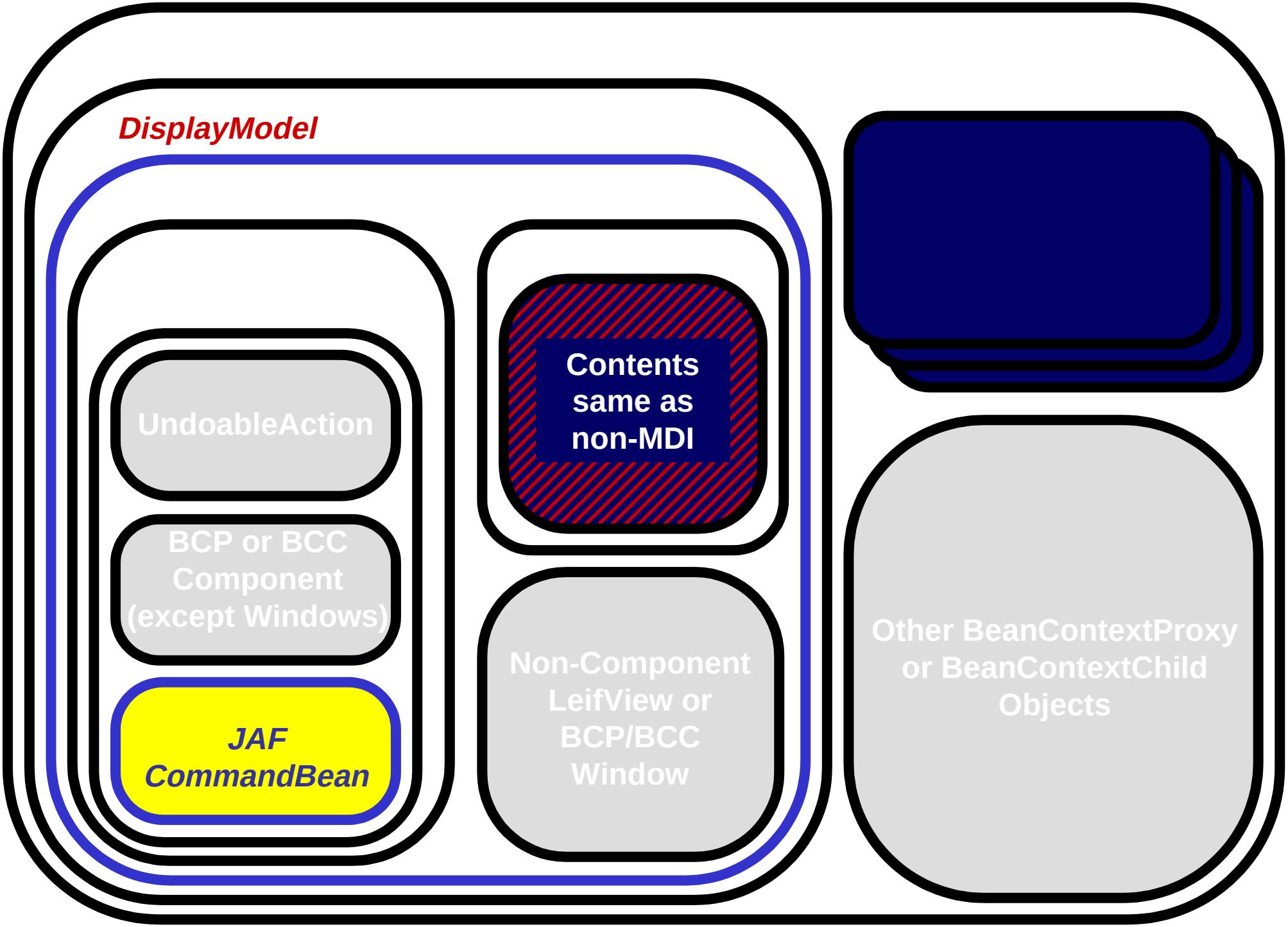
DTAI‘s LEIF, a real world example

http://www.dtai.com/leif
info@dtai.com

Sun microsystems

Dynamic MenuItems Need to Know What's Around




DisplayModel
UndoableAction
BCP or BCC Component (except Windows)
JAF CommandBean
Contents same as non-MDI
Non-Component LeifView or BCP/BCC Window
Other BeanContextProxy or BeanContextChild Objects

JAVA DAYS ´99



Summary

Sun microsystems

JavaBeans™: *The* Component Architecture for the Java™ Platform



- The Beans web site - http://java.sun.com/beans
- Tools listing - http://java.sun.com/beans/tools.html
- 3rd Party Beans catalog - http://java.sun.com/beans/directory
- Feedback - java-beans@eng.sun.com
- News group - comp.java.lang.beans

© 1999 Sun Microsystems GmbH

Sun microsystems