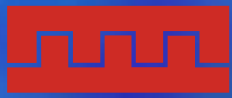


Abeans Application Framework and Widgets Running with EPICS

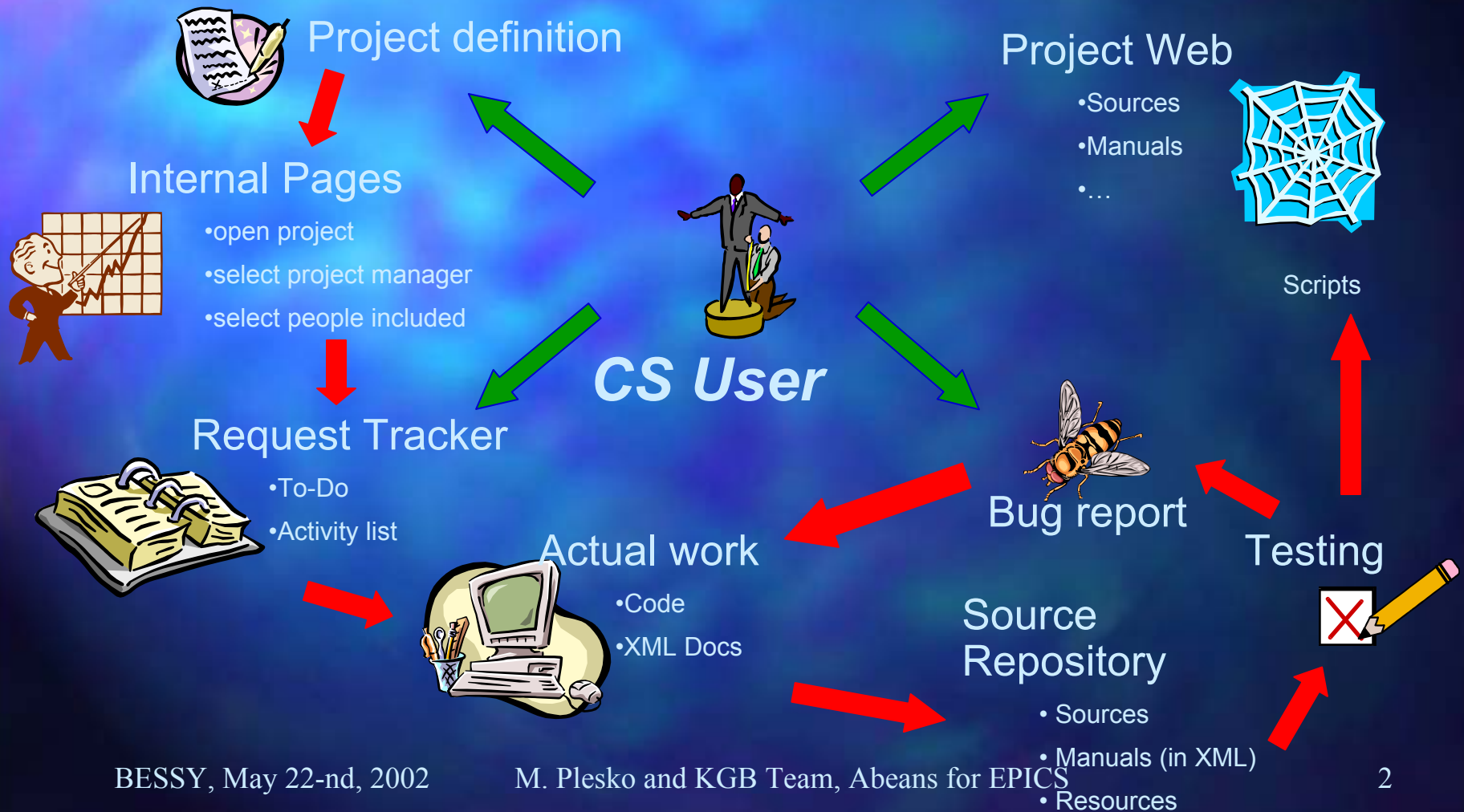
M.Plesko on behalf of **KGB Team**
J. Stefan Institute and Cosylab Ltd.
in collaboration with **SNS** and **DESY**

cosylab 
CONTROL SYSTEM LABORATORY

EPICS Workshop,
Berlin, May 2002



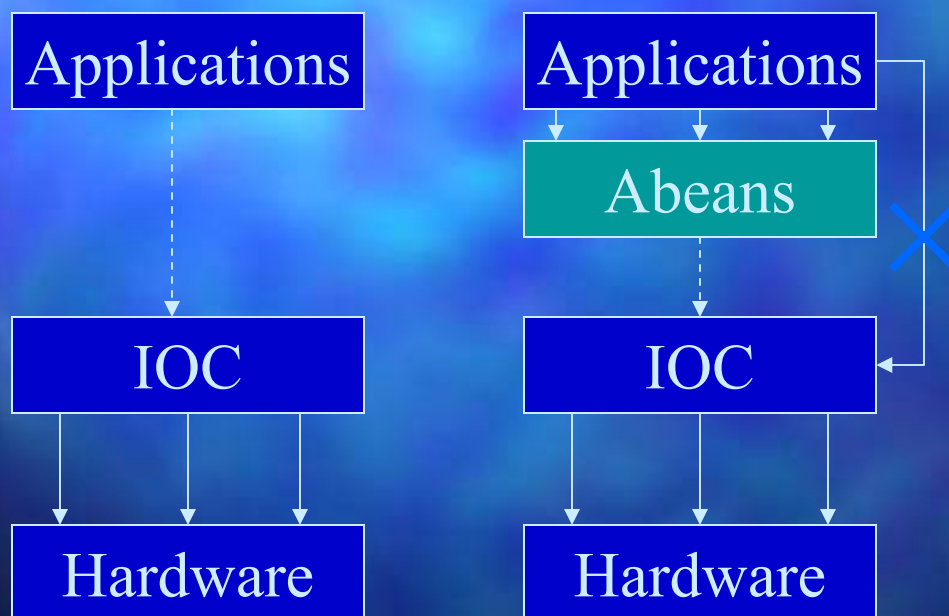
eManagement Cycle



Abeans (or xal at SNS)



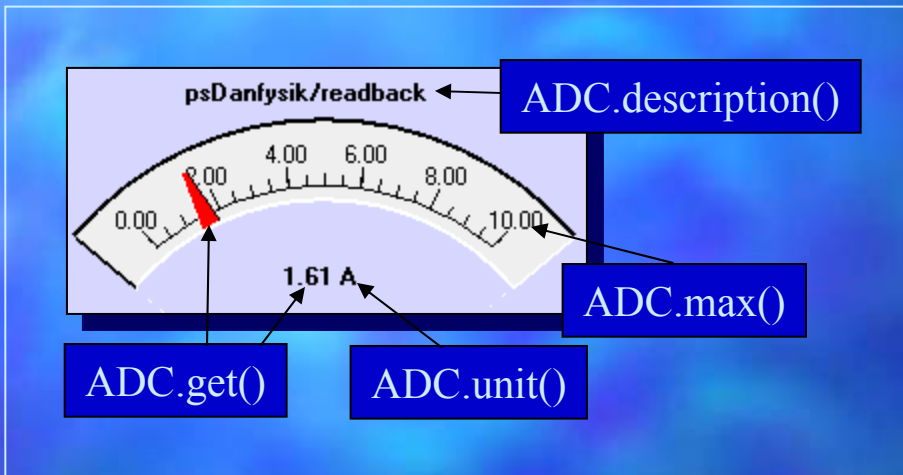
■ Application framework



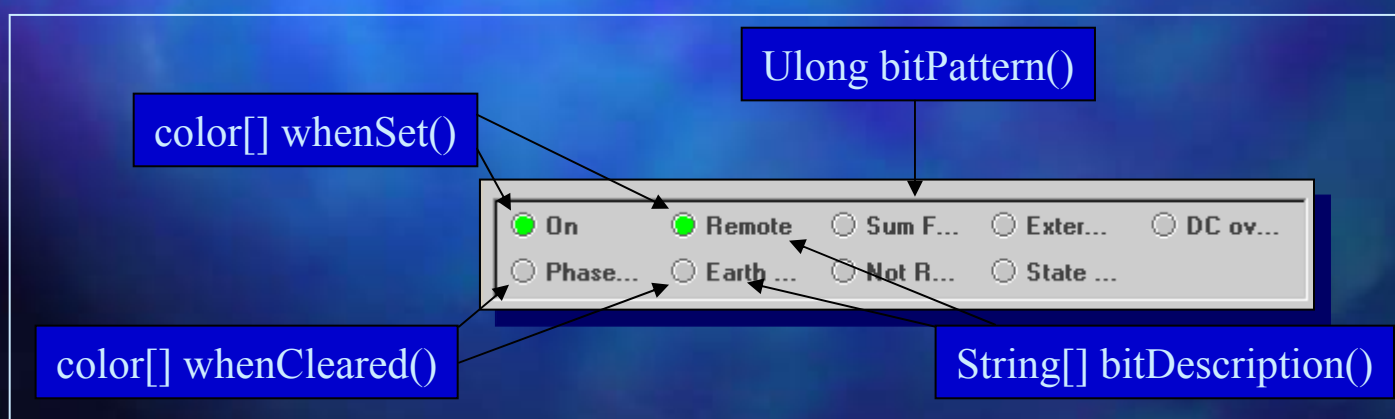
- Hide comm layer
 - Provide application services
 - Provide CS models (device, channel, etc.)
- Σ : RAD, easier maintenance



CosyBeans Widgets



Gauge



Status

Visual Composition

The screenshot displays the EPICSPanel 1.0 software interface. The main window is titled "EPICSPanel 1.0 in com.cosylab.epics" and features a menu bar (File, Bean, Edit, Tools, Workspace, Window, Help) and a toolbar with various icons. Below the toolbar is a tabbed interface with tabs for Source, Members, Hierarchy, Editions, Visual Composition (selected), and BeanInfo.

The central workspace is labeled "Composition Editor" and contains a "Swing" panel. This panel displays two visual components: a line graph and a gauge. The line graph shows four data series (red, pink, blue, yellow) plotted against time (01:00:00 to 01:00:02) with a y-axis ranging from 8 to 26. The gauge is a semi-circular scale from 0 to 100, currently showing 0.00. Below the gauge is a control panel with a text field containing "0.00" and a slider below it.

Two beans are visible in the composition editor: "DoubleChannelBean1" and "DoubleChannelBean2". Blue lines connect these beans to the data series in the line graph and the gauge's control panel.

At the bottom of the window, the status bar indicates "Nothing selected".

Device Table

PowerSupply Table

System

Select All Deselect All PowerSupply * Select >>

Trend Profile Command Snapshot

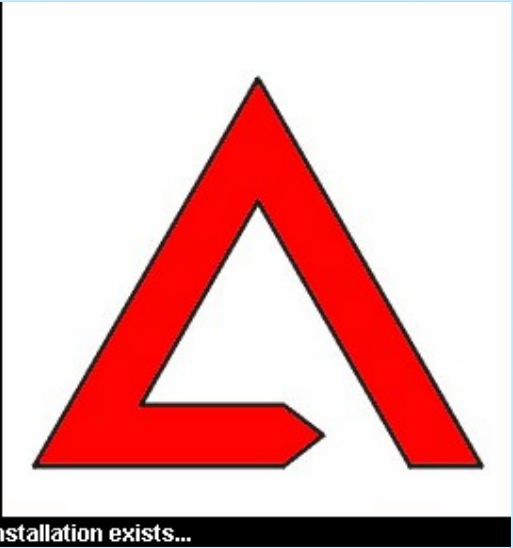
Last Opened Snapshot File: <none>

s...	position	Device	current	readback	s
<input type="checkbox"/>	1.0	PBEND_M.02	0.0000	0.2212	
<input type="checkbox"/>	0.0	PBEND_M.01	0.0000 [in: 2.5000]*	0.2212	
<input type="checkbox"/>	1.0	PBEND_I.01	0.0000		
<input type="checkbox"/>	2.0	PBEND_E.02	0.0000		
<input type="checkbox"/>	1.0	PBEND_E.01	0.0000		
<input type="checkbox"/>	1.0	PBEND_B.01	0.0000		

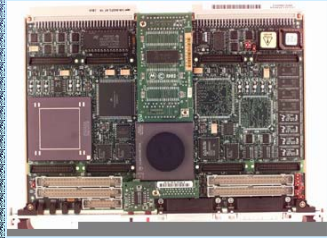
01:17)
Found 141 devices matching search criteria. [SB] (2002-05-19 01:17)
Connection completed. [PBEND_M.02] (2002-05-19 01:17)
Connection completed. [PBEND_M.01] (2002-05-19 01:17)
Connection completed. [PBEND_I.01] (2002-05-19 01:17)
Connection completed. [PBEND_E.02] (2002-05-19 01:17)
Connection completed. [PBEND_E.01] (2002-05-19 01:17)
Connection completed. [PBEND_B.01] (2002-05-19 01:17)

Launcher

Setting up logging. OK
Checking environment. OK
Checking permissions. OK
Checking other instances. OK
Initializing models. OK
Instantiating...
Launcher: Determining if local installation exists...



Abeans + CosyBeans



EPICS Plug (JCA)

Abeans

Double Adapter

Double Adapter Setter

Pattern Adapter

The screenshot shows a control window for 'PBEND_I.01.Current'. At the top is a semi-circular gauge with a scale from 0 to 100 and a green needle pointing to 50. Below the gauge is a control panel with a 'sync' button, navigation arrows, a numerical display showing '123.456', a unit selector 'SI', and a '+' button. A slider below the display is labeled 'Double Displayer, Setter' and has a red marker at approximately 25. At the bottom is a status panel with three columns of indicator lights and labels: 'On', 'Remote', 'Not Ready' (all green); 'Sum Failure', 'Phase Failure', 'Earth Leakage' (all red); and 'DC Overcurrent', 'External Interlock', 'State Inconsistent' (all green).

CosyBeans Features



- Optimized for Control Systems, not just GUIs
 - A result of long thinking, designing, comparing, testing, use at other laboratories, reconsiderations, eg.
 - initialize from CS at run-time
 - control system connection status+timeout handling
 - colour coding of states and alarms
 - Consistent behaviour of widgets in all panels
 - no subtle differences to learn
 - user can fully configure at run-time
 - resize-able, adjust shape to fully use available space
 - intuitive/safe modification of CS parameters (clever slider,...)
 - trend everywhere: history, save as CSV, histogram, zoom, ...
 - No duplication of programming efforts



Abeans Features

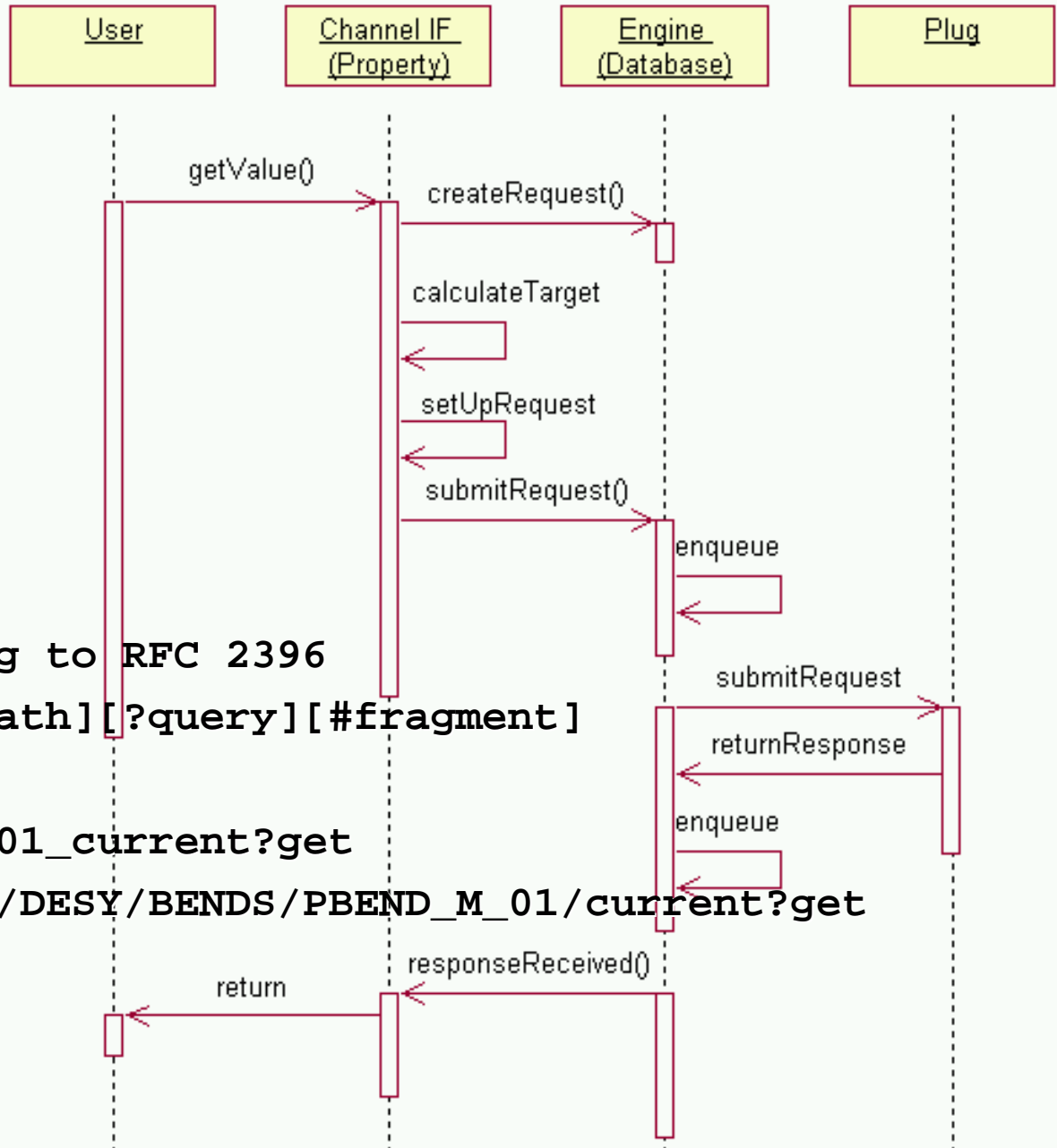
- Hide details of Control System - just one of many:
 - launch application/applet, remote install
 - Logging, alarm, and communication error reporting
 - Resource initialization and destruction, provide defaults
 - find, choose and connect to selected device
 - supports different protocol via plugs (e.g. CORBA, EPICS,...)
- Same core code is used everywhere
 - The code is tested very well
- Visual composition and normal programming together
 - choose device type and command through menu!
 - almost all errors detected by compiler not during operation
 - Abeans discover vis/man mode => change behaviour



Abeans and EPICS ?

- MEDM is faster for “expert” panels, TCL is simpler for prototyping, but:
- **Abeans application development is scalable!**
 - profit from standard Java features
 - GUI layout, file I/O, XML, WebStart, resources, logging,...
 - Abeans fight Java deficiencies
 - optimize graphic performance and reduce memory usage
- **Use Abeans for complex applications...**
 - client-side algorithms (machine physics)
 - display of multiple data (tables, tree, bird-view)
- **...and when long-term maintenance matters**

ca_get()



use URI mapping according to RFC 2396

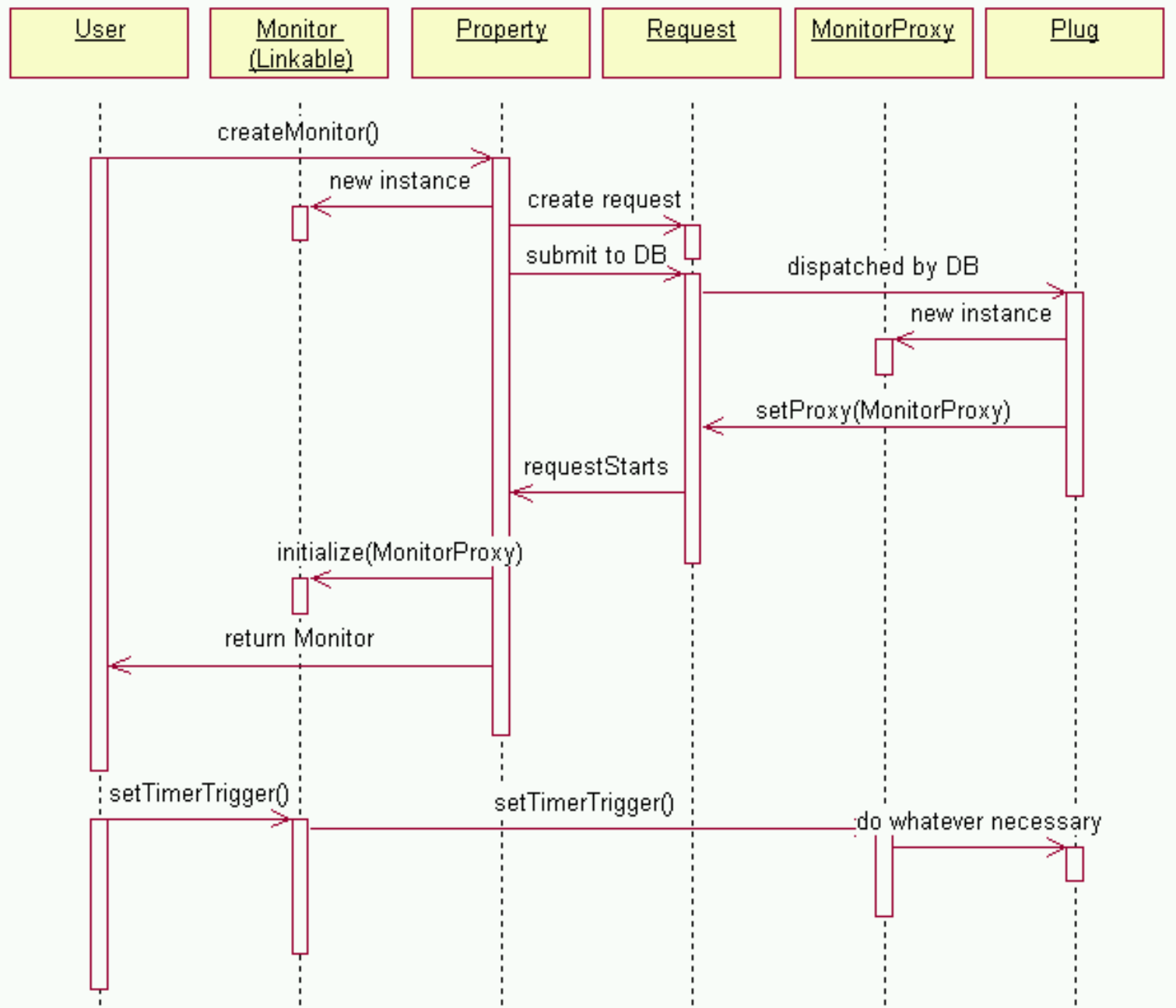
[scheme:][//authority][path][?query][#fragment]

e.g.

abeans-EPICS:///PBEND_M_01_current?get

abeans-TINE://ns.desy.de/DESY/BENDS/PBEND_M_01/current?get

Monitor





Visual Programming

- Panels in RAD (no hand-written code)
 - implement default behavior
 - notifications to user when communication errors occur
 - timeouts
 - exceptions
 - interpret error numbers
 - callback, monitor and alarm queued&dispatched as event
 - default lifecycle management
 - initialization when GUI becomes visible
 - connection when all necessary parameters have been set
 - destruction when application closes
 - ServiceBean provides access to services in “visual programming” mode

Serious Programming



- Manual programming with Abean device Beans
 - device manual = API (Bean properties & methods)
 - compile-time error checking through strong typing
 - sync and async program flow (through lock object)
 - group Abeans into families for same behaviour
 - fine-grained control: ~10 event types, ~15 system properties
- Abeans discover vis/man mode => change behaviour
- Aggregate data for display of large number of similar devices (i.e. tables)