

*Introduction to FlexGantt*

---

# Topic: Components

Dirk Lemmermann  
Software & Consulting  
Zurich, Switzerland

---

---

# Content

---

- ❖ Gantt Chart / Dual Gantt Chart
- ❖ Layer Container
- ❖ Tree Table (sorting, filtering)
- ❖ Row Headers (Tree Table / Layer Container)
- ❖ Timeline, Dateline, Eventline
- ❖ Utility & Navigation Control Panel
- ❖ Selectors

# Gantt Charts

---

# AbstractGanttChart

---

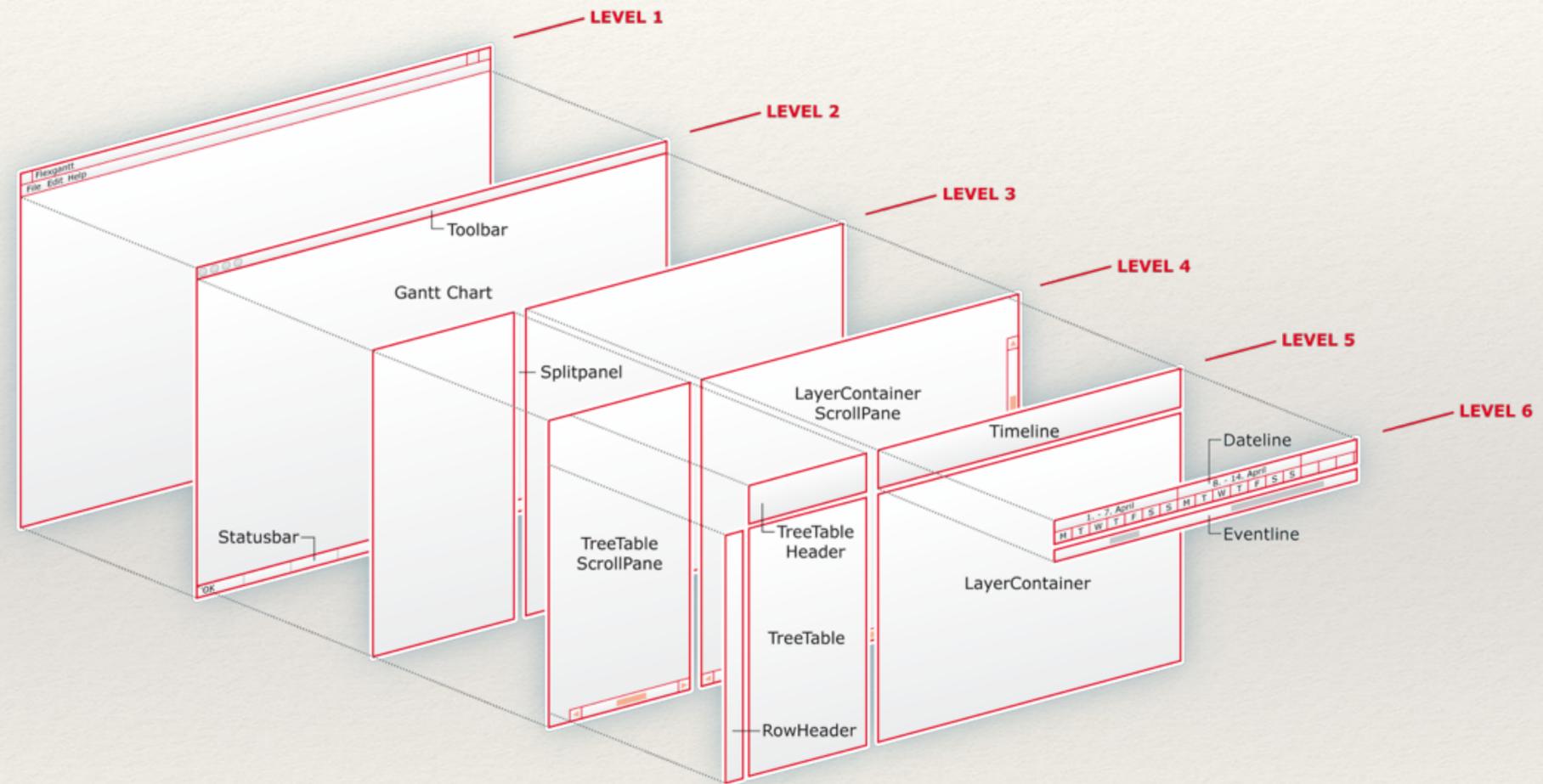
- ❖ Carries almost all functionality.
  - ❖ Creates sub-components via pluggable component factory
  - ❖ Controls visibility of all layers (e.g. `setPopupVisible()`).
  - ❖ Utility methods to control the horizon / overall time span.
  - ❖ Column model (for tree table).
  - ❖ Paging model.
  - ❖ Command stack (`commandExecute / Undo / Redo()`).
  - ❖ Command interceptors.
  - ❖ Messages (`addMessage(IMessage)`).

# Gantt Chart

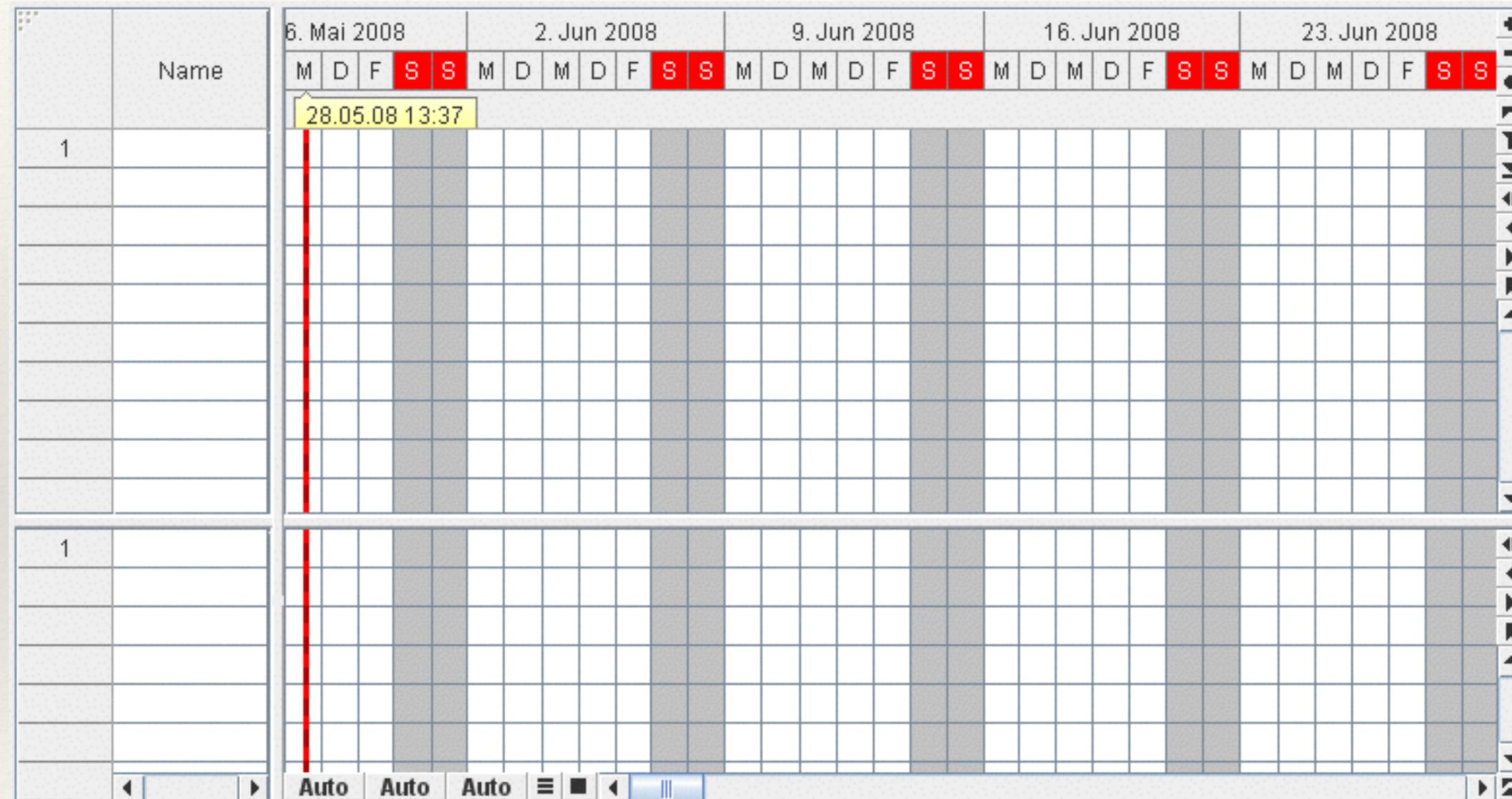


# Gantt Chart

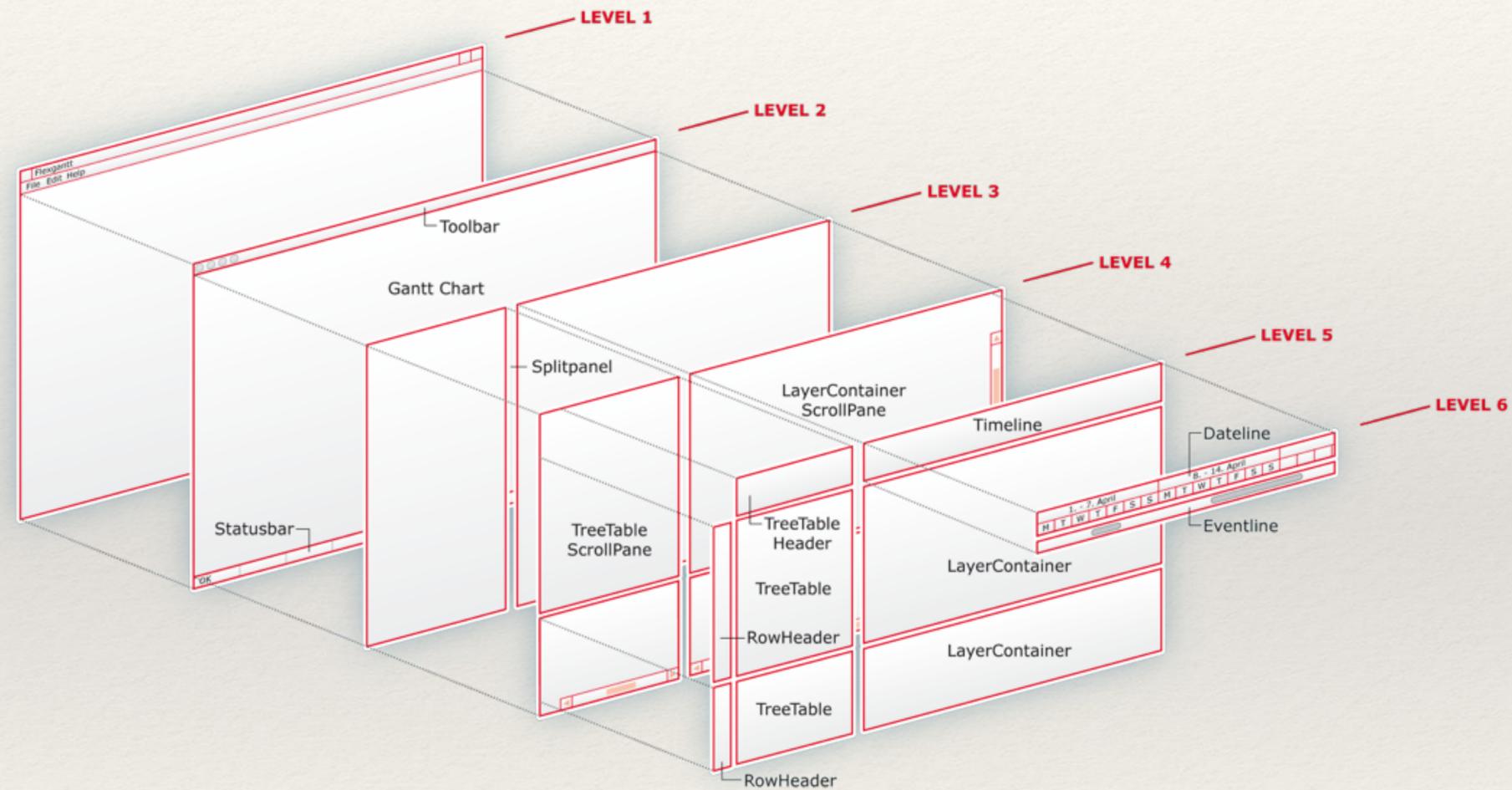
## COMPONENTS



# DualGanttChart



# Dual Gantt Chart



---

# Subclassing / Customizing Components

---

- ❖ Implement your own component factory and pass it to the Gantt chart constructor.
- ❖ Subclass DefaultComponentFactory.
- ❖ Override appropriate component factory method.
- ❖ Call `super.createXYZ()` if you just want to perform some settings or .....
- ❖ ..... create your own subclass of the XYZ component and return it.

---

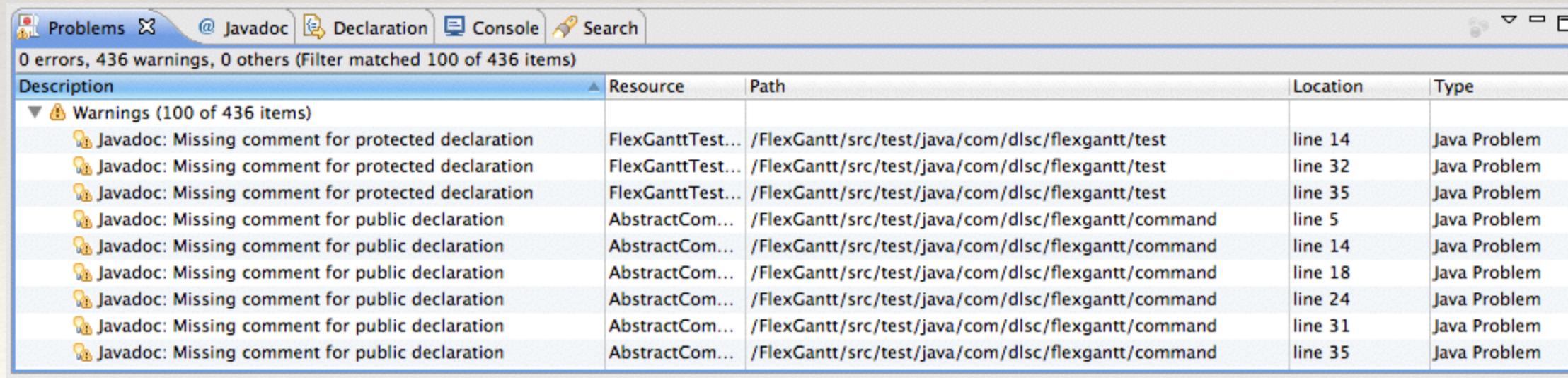
# ComponentFactory

---

```
public interface IComponentFactory {  
    Timeline createTimeline(AbstractGanttChart gc);  
    Dateline createDateline(Timeline timeline);  
    Eventline createEventline(Timeline timeline, Dateline dateline);  
    TreeTable createTreeTable(AbstractGanttChart gc, ITreeTableModel model);  
    TreeTableRowHeader createTreeTableRowHeader(TreeTable table);  
    LayerContainer createLayerContainer(AbstractGanttChart gc, TreeTable table, IGanttChartModel model);  
    LayerContainerRowHeader createLayerContainerRowHeader(LayerContainer lc);  
    ...  
}
```

# Messages

- ❖ Messages can be attached to the Gantt chart.
- ❖ Similar to warnings / errors shown in Eclipse view.
- ❖ `AbstractGanttChart.addMessage(IMessage)`: `TreePathMessage`, `TimelineObjectPathMessage`, `RelationshipMessage`
- ❖ Double-click takes the user to the problematic area.



The screenshot shows the Eclipse IDE's Problems view. The title bar includes tabs for Problems, Javadoc, Declaration, Console, and Search. The main area displays a summary: "0 errors, 436 warnings, 0 others (Filter matched 100 of 436 items)". Below this is a table with columns for Description, Resource, Path, Location, and Type. The table lists several warnings related to missing Javadoc comments for protected and public declarations in various files.

Description	Resource	Path	Location	Type
Warnings (100 of 436 items)				
Javadoc: Missing comment for protected declaration	FlexGanttTest...	/FlexGantt/src/test/java/com/dlsc/flexgantt/test	line 14	Java Problem
Javadoc: Missing comment for protected declaration	FlexGanttTest...	/FlexGantt/src/test/java/com/dlsc/flexgantt/test	line 32	Java Problem
Javadoc: Missing comment for protected declaration	FlexGanttTest...	/FlexGantt/src/test/java/com/dlsc/flexgantt/test	line 35	Java Problem
Javadoc: Missing comment for public declaration	AbstractCom...	/FlexGantt/src/test/java/com/dlsc/flexgantt/command	line 5	Java Problem
Javadoc: Missing comment for public declaration	AbstractCom...	/FlexGantt/src/test/java/com/dlsc/flexgantt/command	line 14	Java Problem
Javadoc: Missing comment for public declaration	AbstractCom...	/FlexGantt/src/test/java/com/dlsc/flexgantt/command	line 18	Java Problem
Javadoc: Missing comment for public declaration	AbstractCom...	/FlexGantt/src/test/java/com/dlsc/flexgantt/command	line 24	Java Problem
Javadoc: Missing comment for public declaration	AbstractCom...	/FlexGantt/src/test/java/com/dlsc/flexgantt/command	line 31	Java Problem
Javadoc: Missing comment for public declaration	AbstractCom...	/FlexGantt/src/test/java/com/dlsc/flexgantt/command	line 35	Java Problem

# Messages

The screenshot displays a software interface for managing tasks and messages. At the top, a 'Navigation' toolbar contains various icons for navigation and editing. Below it, a Gantt chart shows a timeline from February 10, 2014, to March 10, 2014. The chart is divided into weekly segments, each with a day-of-the-week indicator (S, M, D, M, D, F, S, S). Blue bars represent task durations across the timeline. A 'Messages' dialog box is open in the foreground, listing several messages with their hierarchical paths. The second message, 'Node 0 / Sub Node 1 / tlo #7', is highlighted in blue. To the right of the Gantt chart, an 'Overview / Radar' component shows a grid with a red vertical bar and several colored dots (red, blue, green, yellow). Below this, a smaller 'Messages' list is visible, mirroring the one in the dialog box. At the bottom left, a tree view shows a hierarchy of nodes: 'Sub Node 1', 'Sub Node 2', 'Node 7', and 'Sub Node 0'. The bottom status bar shows the file name 'Untitled', the system clock '11.01.14 16:03 - 17.01.14 16:03', and system information '08.12.13 16:03' and '44,0 MB / 81,1 MB'.

Navigation

Name	10. Feb 2014	17. Feb 2014	24. Feb 2014	3. Mrz 2014	10. Mrz
	S M D M D F S S	M D M D F S S	M D M D F S S	M D M D F S S	M D M D

Messages

Mitteilung

- Node 0 / Sub Node 0 / tlo #1
- Node 0 / Sub Node 1 / tlo #7**
- Node 0 / Sub Node 1 / tlo #8
- Node 0 / Sub Node 2 / tlo #5
- Node 0 / Sub Node 2 / tlo #20
- Node 0 / Sub Node 2 / tlo #25

Overview / Radar

Messages

Mitteilung

- Node 0 / Sub Node 0 / tlo #1
- Node 0 / Sub Node 1 / tlo #7**
- Node 0 / Sub Node 1 / tlo #8
- Node 0 / Sub Node 2 / tlo #5
- Node 0 / Sub Node 2 / tlo #20
- Node 0 / Sub Node 2 / tlo #25

NavigationDemo.java  
NavigationDemoTimelineObjectRenderer.java  
NavigationGanttChartModel.java

27 Sub Node 1  
28 Sub Node 2  
29 Node 7  
30 Sub Node 0

Auto Auto

Untitled 11.01.14 16:03 - 17.01.14 16:03 08.12.13 16:03 44,0 MB / 81,1 MB

# Layer Container

---

# LayerContainer

---

- ❖ Responsible for displaying the various layer types:
  - ❖ System Layer, Timeline Object Layers, Custom Layers.
  - ❖ Layer functions: show / hide / to front / to back.
  - ❖ Stores the timeline object layer selection models.
  - ❖ Navigation functions: show all, earliest, latest TLOs.

---

# LayerContainer

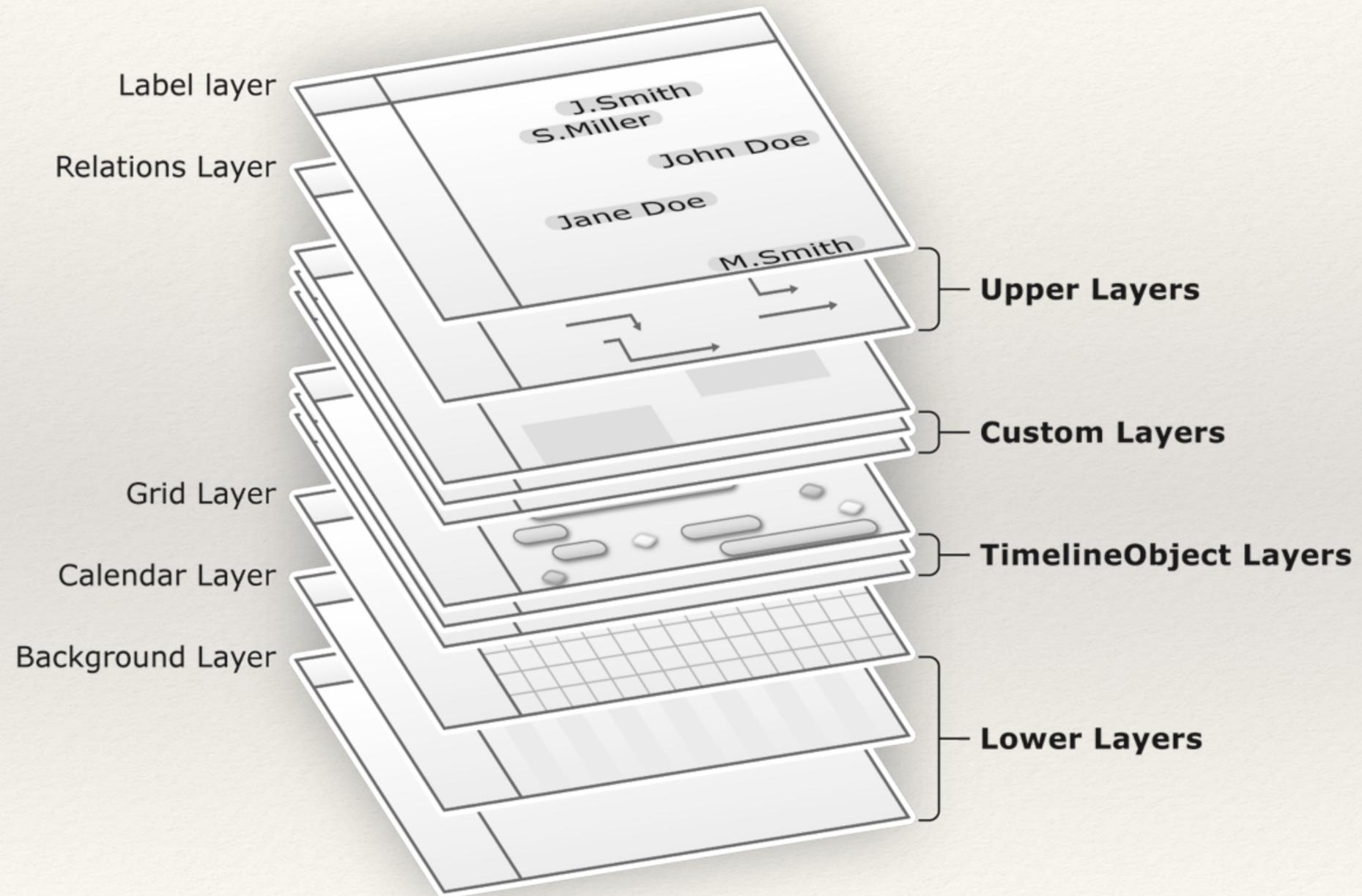
---

- ❖ Determines the order of the system layers.

```
protected List<Class<? extends AbstractSystemLayer>>  
    getTypesOfLowerSystemLayers();
```

```
protected List<Class<? extends AbstractSystemLayer>>  
    getTypesOfUpperSystemLayers();
```

# Layer Stack



# Tree Table

# Tree Table

- ❖ The left-hand side of each Gantt chart is a tree table.
- ❖ The tree table is the „master“ controlling the layer container.
- ❖ Not Swing table or Swing tree.
- ❖ Uses Swing concepts: pluggable renderers and editors.
- ❖ Invokes policies to control its behavior.
- ❖ Uses a central column model.

1		 Node 0
2		 Sub Node 0
3		 Sub Node 1
4		 Sub Node 2
5		 Node 1
6		 Sub Node 0
7		 Sub Node 1
8		 Sub Node 2

---

# Tree Table: Filtering

---

```
/**
 * This interface can be used to filter the rows / nodes shown in the
 * tree table component.
 *
 * @see TreeTable#setNodeFilter(INodeFilter)
 */
public interface INodeFilter {

    /**
     * Determines whether the given node will be included in the tree table
     * view or not.
     */
    boolean includeNode(Object node);
}
```

---

# Tree Table: Sorting 1

---

DefaultMutableTreeNode:

```
public void sort(int[] modelIndices, boolean[] sortDirections) {
    this.sortModelIndices = modelIndices;
    this.sortDirections = sortDirections;
    if (children != null) {
        Collections.sort(children, this);
        for (Object obj : children) {
            ITreeNode child = (ITreeNode) obj;
            child.sort(modelIndices, sortDirections);
        }
    }
}
```

---

# Tree Table: Sorting 2

---

DefaultMutableTreeNode (implements Comparable):

```
public int compare(Object o1, Object o2) {
    ITreeNode node1 = (ITreeNode) o1;
    ITreeNode node2 = (ITreeNode) o2;
    for (int i = 0; i < sortModelIndices.length; i++) {
        int modelIndex = sortModelIndices[i];
        boolean ascending = sortDirections[i];
        Object value1 = null;
        Object value2 = null;
        if (modelIndex == KeyColumn.MODEL_INDEX) {
            value1 = node1.getKey();
            value2 = node2.getKey();
        } else {
            value1 = node1.getColumnValue(modelIndex);
            value2 = node2.getColumnValue(modelIndex);
        }
        int result = compare(modelIndex, value1, value2, ascending);
        if (result != 0) {
            return result;
        }
    }
    return 0;
}
```

---

# Tree Table: Sorting 3

---

DefaultMutableTreeNode (implements Comparable):

```
protected int compare(int modelIndex, Object value1, Object value2,
    boolean ascending) {

    if (value1 == null) return -1;
    if (value2 == null) return +1;

    Comparator comparator = null;
    if (modelIndex == KeyColumn.MODEL_INDEX) {
        comparator = getKeyComparator();
    } else {
        comparator = getComparator(modelIndex);
    }
    int result = 0;
    if (comparator != null) {
        result = comparator.compare(value1, value2);
    } else if (value1 instanceof Comparable) {
        result = ((Comparable) value1).compareTo(value2);
    } else {
        throw new IllegalArgumentException("unable to sort model index");
    }
    if (!ascending) {
        if (result != 0) {
            result = -result;
        }
    }
    return result;
}
```

# Row Headers

---

# Tree Table Row Header

---

- ❖ A component used for displaying row numbers and expand / collapse icons on the left-hand side of the tree table.
- ❖ Can be customized to display any kind of information.
- ❖ Uses an `ITreeTableRowHeaderRenderer` instance to draw the cells.



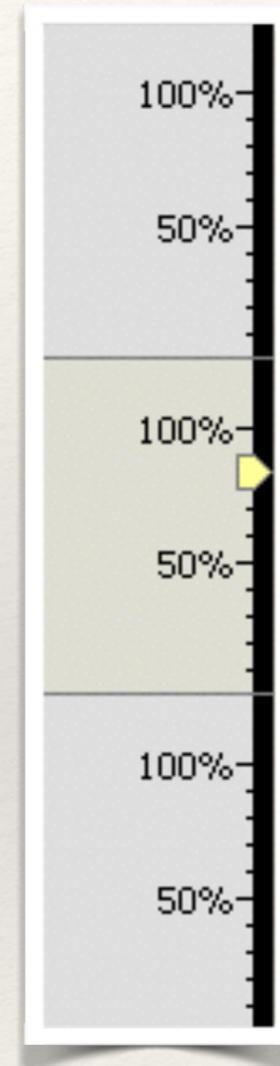
0	-
1	-
2	+
22	-
23	-
24	-
25	
26	
27	
28	-

---

# LayerContainerRowHeader

---

- ❖ Shown on the left edge of the layer container.
- ❖ The row header can be used to visualize row specific information that is not supposed to move while the user scrolls horizontally.
- ❖ Uses `ILayerContainerRowHeaderRenderer` to draw the cells.



# NavigationControlPanel

---

# Navigation Control Panel

---

- ❖ Created by the component factory as the UPPER\_RIGHT\_CORNER of the layer container.
- ❖ Various control features (zoom in / out, etc...).

```
public NavigationControl getNavigationControl(NavigationControlType type);
```

---

# Navigation Controls

---



Set time span / horizon



Zoom Out



Zoom In



Go to „now“



Show all objects



Go to latest objects



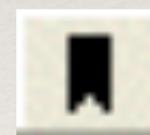
Go to earliest objects



Select a granularity



Go to specific time point



Bookmarks

# UtilityControlPanel

---

# UtilityControlPanel

---

- ❖ Created by the component factory as the LOWER\_LEFT\_CORNER of the layer container.
- ❖ Adjusts its controls by itself.
- ❖ Shows paging controls if a „paging model“ is set.
- ❖ Shows grid controls based on number of layer containers shown by the Gantt chart.



Timeline, Dateline, Eventline

---

# Timeline

---

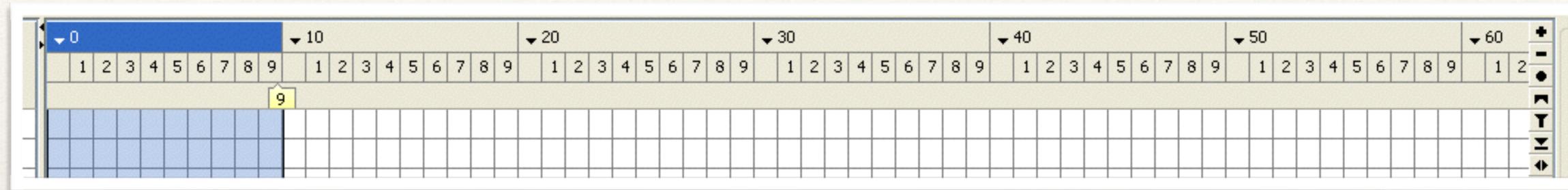
- ❖ Parent container for Dateline and Eventline.
- ❖ Dateline displays the time (Mon, Tue, Wed, .....
- ❖ Eventline displays „global“ activities and milestones..

# Dateline



- ❖ Displays a major and a minor time granularity.
- ❖ Uses an instance of IDatelineModel to ...
  - ❖ Compute grid line locations
  - ❖ Format date strings
  - ❖ Retrieve spreadsheet cell width
- ❖ Appearance customizable via pluggable dateline renderers.

# SimpleDateline



- ❖ „Simply“ counts units (1, 10, 100, 1000, 10000, ...)
- ❖ Used for planning, not scheduling.
- ❖ Uses an instance of SimpleGranularityDatelineModel

# Eventline



- ❖ Displays „markers“: time at cursor location, time span during drag.
- ❖ Displays „global“ activities and milestones that are relevant for all rows in the Gantt chart (e.g. holidays).
- ❖ Eventline objects can be created and edited directly inside the eventline.
- ❖ The time span of eventline objects can be visualized by the EventlineLayer in the layer container below.

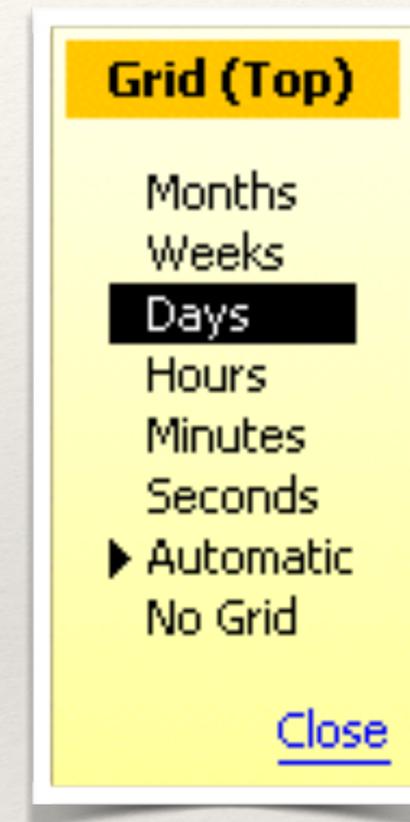
# Selectors

---

# Selectors

---

- ❖ Little slide in / slide out dialogs for quickly changing settings.
- ❖ Provided out-of-the-box, but can be customized.
- ❖ ISelectorFactory is used to produce the selectors.
- ❖ Override `AbstractGanttChart.getSelectorFactory()` to use custom factory.



# Selectors

- ❖ Bookmarks
- ❖ Granularity
- ❖ Grid
- ❖ Key Strokes
- ❖ Layers

**Bookmarks**

-  Conflicting Resources
-  Delayed Orders
-  Overallocations In Q1
-  Vacation Time

[Close](#)

**Grid (Top)**

- Months
- Weeks
- Days
- Hours
- Minutes
- Seconds
- ▶ Automatic
- No Grid

[Close](#)

**Granularity**

- Minutes
- Minutes (5)
- Minutes (10)
- Minutes (15)
- Minutes (30)
- Hours
- Hours (3)
- Hours (6)
- Days
- ▶ Days (Tiny)
- Weeks
- Months
- Months (Tiny)
- Months (3)
- Months (6)
- Years

[Close](#)

**Layers**

<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Layer 3 </li> <li><input checked="" type="checkbox"/> Layer 2 </li> <li><input checked="" type="checkbox"/> Layer 1 </li> </ul>	<ul style="list-style-type: none"> <li><input checked="" type="checkbox"/> Layer 3 </li> <li><input checked="" type="checkbox"/> Layer 2 </li> <li><input checked="" type="checkbox"/> Layer 1 </li> </ul>
---	---

[Close](#)

**Key Strokes**

**Timeline**

 Zoom in	<b>Strg+Plus</b>
 Zoom out	<b>Strg+Minus</b>
 Scroll Left	<b>Strg+Links</b>
 Scroll Left (Fast)	<b>Strg+Umschalt+Links</b>
 Scroll Right	<b>Strg+Rechts</b>
 Scroll Right (Fast)	<b>Strg+Umschalt+Rechts</b>
 Eventline	<b>Strg+E</b>

**Table**

Cell Above	<b>Oben, Umschalt+Eingabe</b>
Cell Below	<b>Unten, Eingabe</b>
Previous Cell	<b>Links, Umschalt+Tabulator</b>
Next Cell	<b>Rechts, Tabulator</b>
First Cell	<b>Pos 1</b>
Last Cell	<b>Ende</b>
Page Up	<b>Bild auf</b>
Page Down	<b>Bild ab</b>
Insert Row	<b>Einf, Strg+Alt+I</b>
 Right Shift Row	<b>Strg+Umschalt+I</b>
 Left Shift Row	<b>Strg+Umschalt+O</b>
Expand Row	<b>Strg+Eingabe</b>
 Delete Row	<b>Strg+Umschalt+Rücktaste, Umschalt+Entf</b>
Select Next Row	<b>Umschalt+Unten</b>
Select Previous Row	<b>Umschalt+Oben</b>
Switch Tables	<b>Strg+T</b>
Previous Table	<b>Strg+Umschalt+T</b>

**Gantt Chart**

Add Highlighting	<b>Strg+H</b>
Remove Highlighting	<b>Strg+Umschalt+H</b>
Clear Highlighting	<b>Alt+Umschalt+H</b>
 Delete Activity	<b>Strg+Rücktaste, Entf</b>
 Crosshair	<b>Strg+C</b>
 Grid Lines	<b>Strg+G</b>

**General**

 Undo	<b>Strg+Z</b>
 Redo	<b>Strg+Y</b>
 Split View	<b>Strg+S</b>

[Close](#)

# Selectors

- ❖ Overview
- ❖ Paging
- ❖ Show Time
- ❖ Columns
- ❖ Horizon

