

# Replace and extend the Eclipse Framework

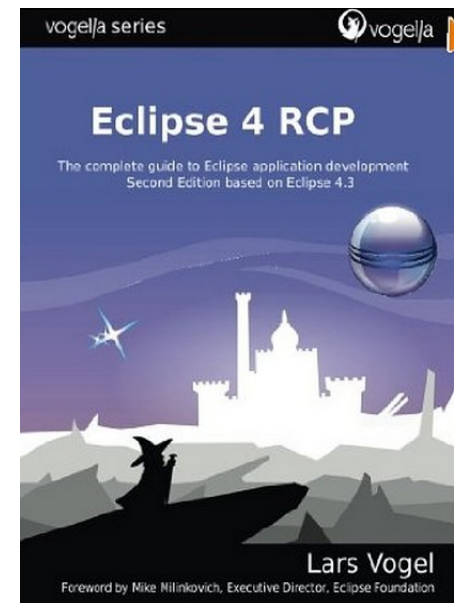


# Lars Vogel

[Lars Vogel](#) Eclipse platform and e4 committer

Founder of vogella GmbH, Trainer and Consultant for Eclipse and Android.

Author of the Eclipse 4 RCP book



# Framework design rule #1

As simple as possible

DnD is  
provided by a  
model add-on

DnDAddon is  
registered via a  
model  
processor at  
runtime

DnDAddon get  
triggered via  
events

DnDAddon  
modifies the  
application  
model























# Framework design rule #2

Extensible and flexible

What was  
Number  
#1 again?





- ▼  Application
  - ▶ Addons
  - ▼ Binding Contexts
    - ▼  Binding Context - In Dialog and Windows
      -  Binding Context - In Windows
      -  Binding Context - In Dialogs
  - ▼ BindingTables
    - ▼  BindingTable - In Dialog and Windows
      -  KeyBinding - M1+Q
      -  KeyBinding - M1+O
      -  KeyBinding - M1+S
      -  KeyBinding - M1+A
  - ▼ Handlers
    -  Handler - quitCommand
    -  Handler - openCommand
    -  Handler - saveCommand
    -  Handler - aboutCommand
  - ▼ Commands
    -  Command - quitCommand
    -  Command - openCommand
    -  Command - saveCommand
    -  Command - aboutCommand
    - Command Categories
  - ▼ Windows and Dialogs
    - ▼  Trimmed Window - test
      - ▼  Main Menu
        - ▼  Menu - File
          - ▼  HandledMenuItem - Open
            - Parameters
          - ▼  HandledMenuItem - Save

# Presentation is flexible

- ▼ Application
  - ▶ Addons
  - ▼ Binding Contexts
    - ▼ Binding Context - In Dialog and Windows
      - Binding Context - In Windows
      - Binding Context - In Dialogs
  - ▼ BindingTables
    - ▼ BindingTable - In Dialog and Windows
      - KeyBinding - M1+Q
      - KeyBinding - M1+O
      - KeyBinding - M1+S
      - KeyBinding - M1+A
  - ▼ Handlers
    - Handler - quitCommand
    - Handler - openCommand
    - Handler - saveCommand
    - Handler - aboutCommand
  - ▼ Commands
    - Command - quitCommand
    - Command - openCommand
    - Command - saveCommand
    - Command - aboutCommand
    - Command Categories
  - ▼ Windows and Dialogs
    - ▼ Trimmed Window - test
      - ▼ Main Menu
        - ▼ Menu - File
          - ▼ HandledMenuItem - Open
            - Parameters
          - HandledMenuItem - Save

# What about the structure?

Eclipse Modeling  
Framework (EMF) is the  
underlying basis of the  
application model



EMF extension +  
Custom renderer

# Demo

Solution App in Github repo  
`com.vogella.e4.appmodel.app`

We have this also as  
exercise but  
it will be a loooooong  
exercise  
if you do not  
know EMF



Example for  
model  
extensions:

WAZAABI talk  
on Thursday  
11:00 – 11:35

+Context Modifications

All the flexibility at your  
fingertips

+Model Add-ons

+ContextFunctions



A waiter in a white tuxedo shirt, black bow tie, and black vest is holding a silver tray. The waiter's face is not visible. The background is a plain, light-colored wall.

**Lets mix  
some things**



# Propagating the selection

To-Dos

Load Data

Summary	Description
Application m	Flexible and extensible
DI	@Inject as programming mo
OSGi	Services
SWT	Widgets
JFace	Especially Viewers!
CSS Styling	Style your application
Eclipse service	Selection, model, Part



Details

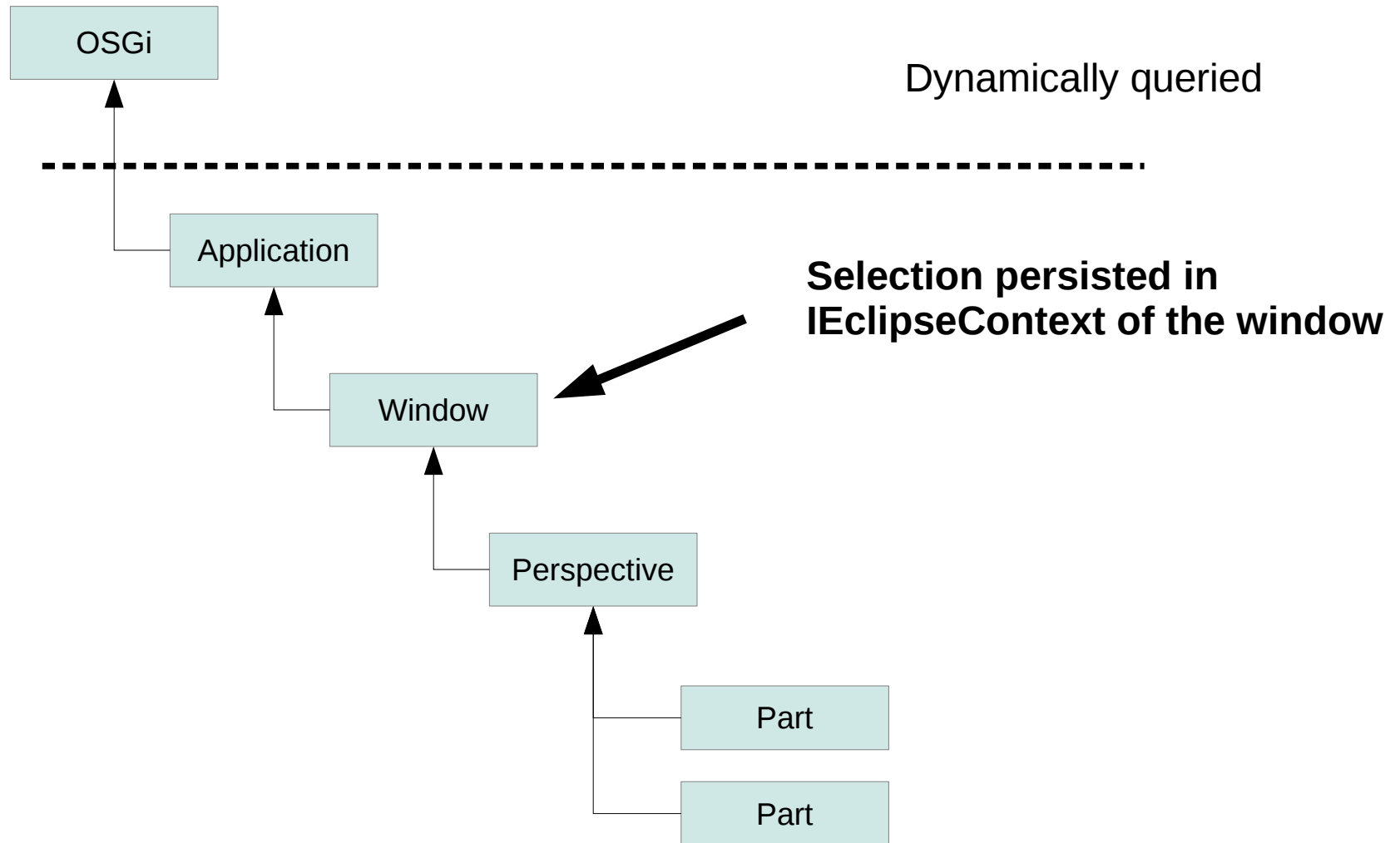
Summary

Description

Due Date

☐ Done

# Dependency injection



## Setting the selection in one Part:

```
@Inject
ESelectionService selectionService;

viewer.addSelectionChangedListener(new ISelectionChangedListener() {
    @Override
    public void selectionChanged(SelectionChangedEvent event) {
        IStructuredSelection selection = (IStructuredSelection) viewer.getSelection();
        service.setSelection(selection.getFirstElement());
    }
});
```

---

## Consume active selection in another Part:

```
@Inject
public void setTodo(
    @Optional @Named(IServiceConstants.ACTIVE_SELECTION) Todo todo) {
    if (todo != null) {
        // Remember the todo as field
        this.todo = todo;
        // update the user interface
        updateUserInterface(todo);
    }
}
```

## Setting the selection in one Part:

```
@Inject
ESelectionService selectionService;

viewer.addSelectionChangedListener(new ISelectionChangedListener() {
    @Override
    public void selectionChanged(SelectionChangedEvent event) {
        IStructuredSelection selection = (IStructuredSelection) viewer.getSelection();
        service.setSelection(selection.getFirstElement());
    }
});
```

---


## Consume active selection in another Part:

```
@Inject
public void setTodo(
    @Optional @Named(IServiceConstants.ACTIVE_SELECTION) Todo todo) {
    if (todo != null) {
        // Remember the todo as field
        this.todo = todo;
        // update the user interface
        updateUserInterface(todo);
    }
}
```

**Selection is based  
on the  
MWindow  
context**







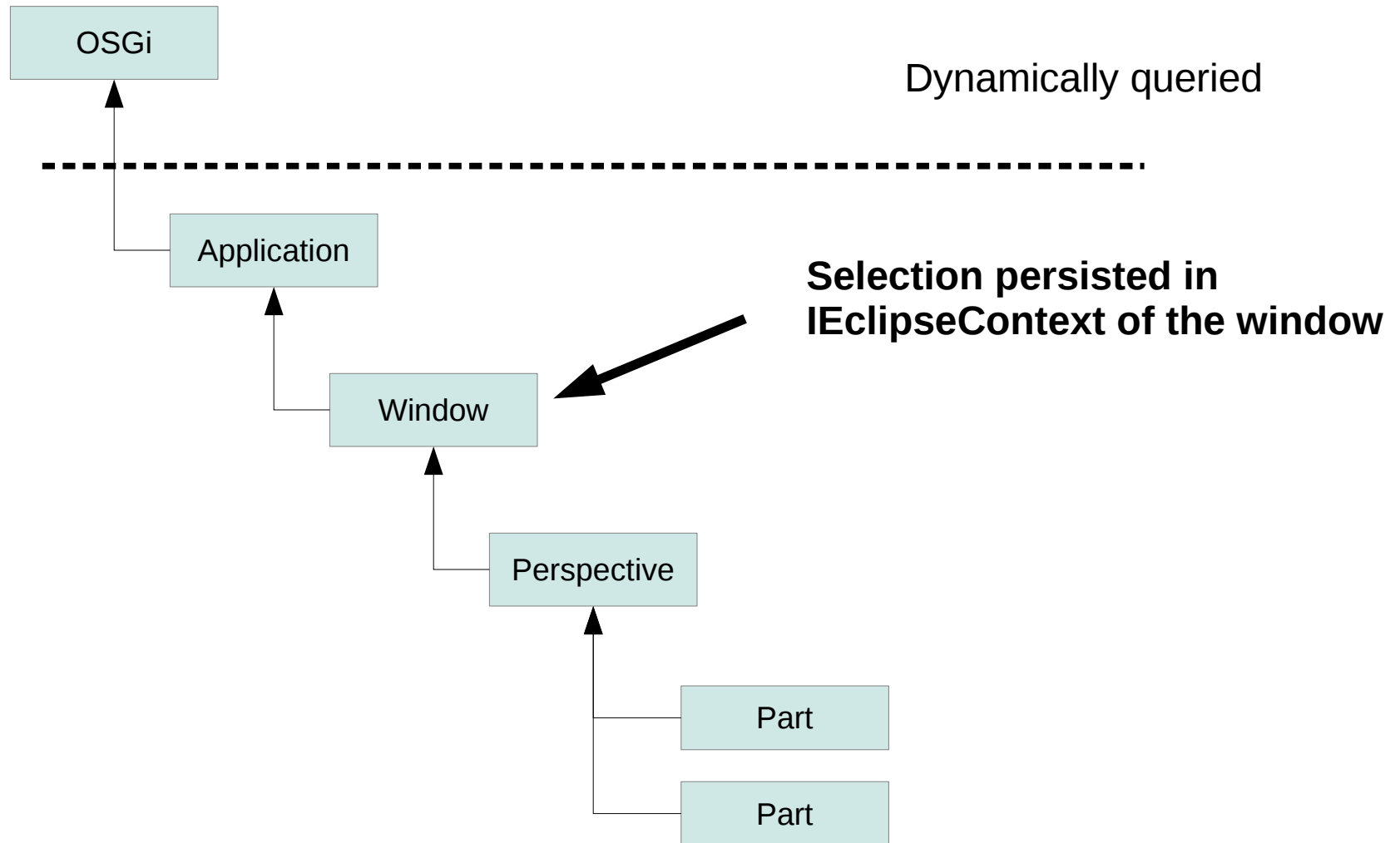
**I deleted  
the  
ESelectionService**



A waiter in a formal tuxedo, consisting of a white shirt, a black bow tie, and a black vest, is holding a large, round, silver tray. The waiter's hands are visible, with the right hand supporting the tray from underneath and the left hand resting on the edge. The background is a plain, light-colored wall. The word "Context" is overlaid in white text on the right side of the image.

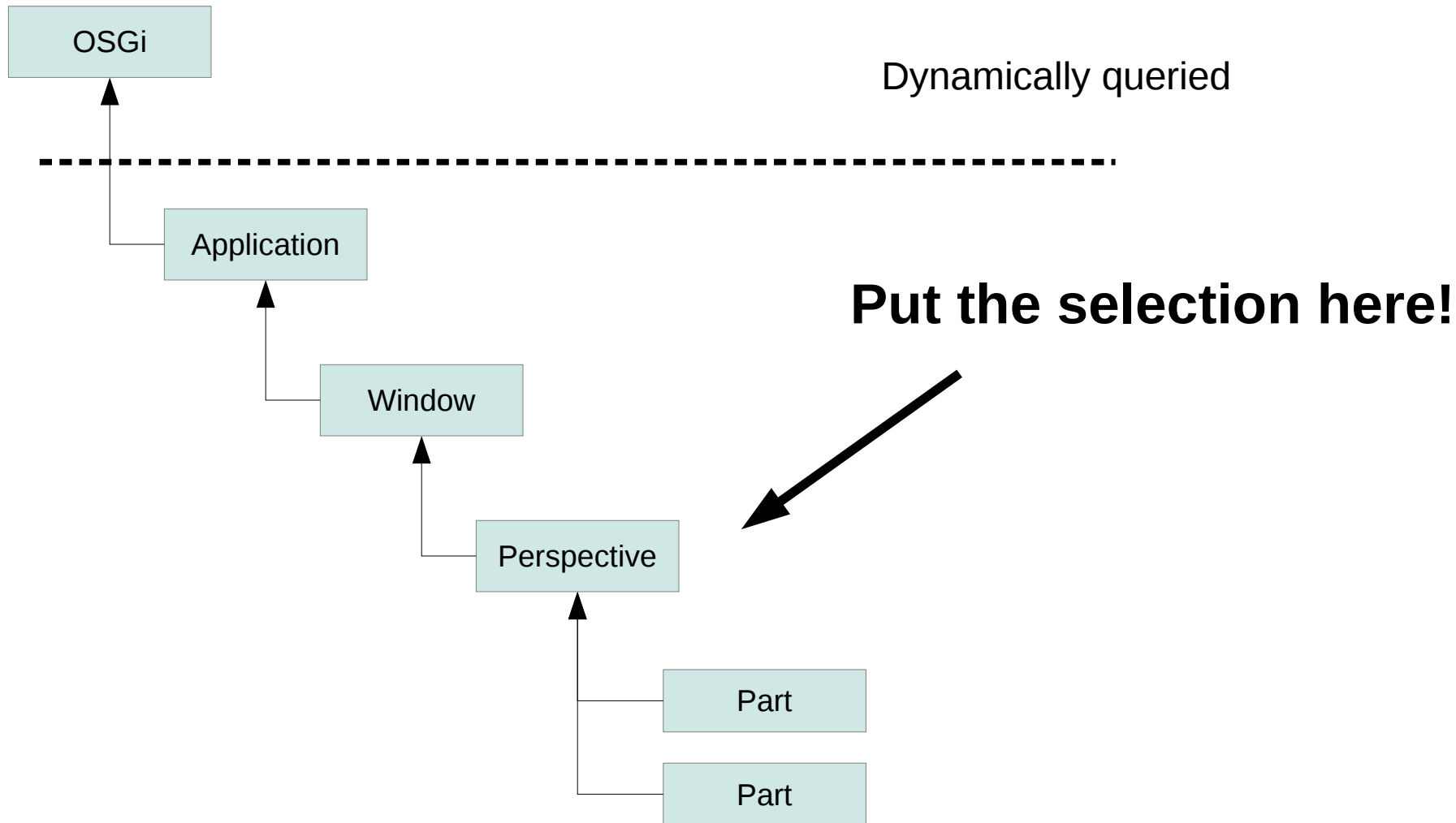
Context

# Current solution





# Desired solution



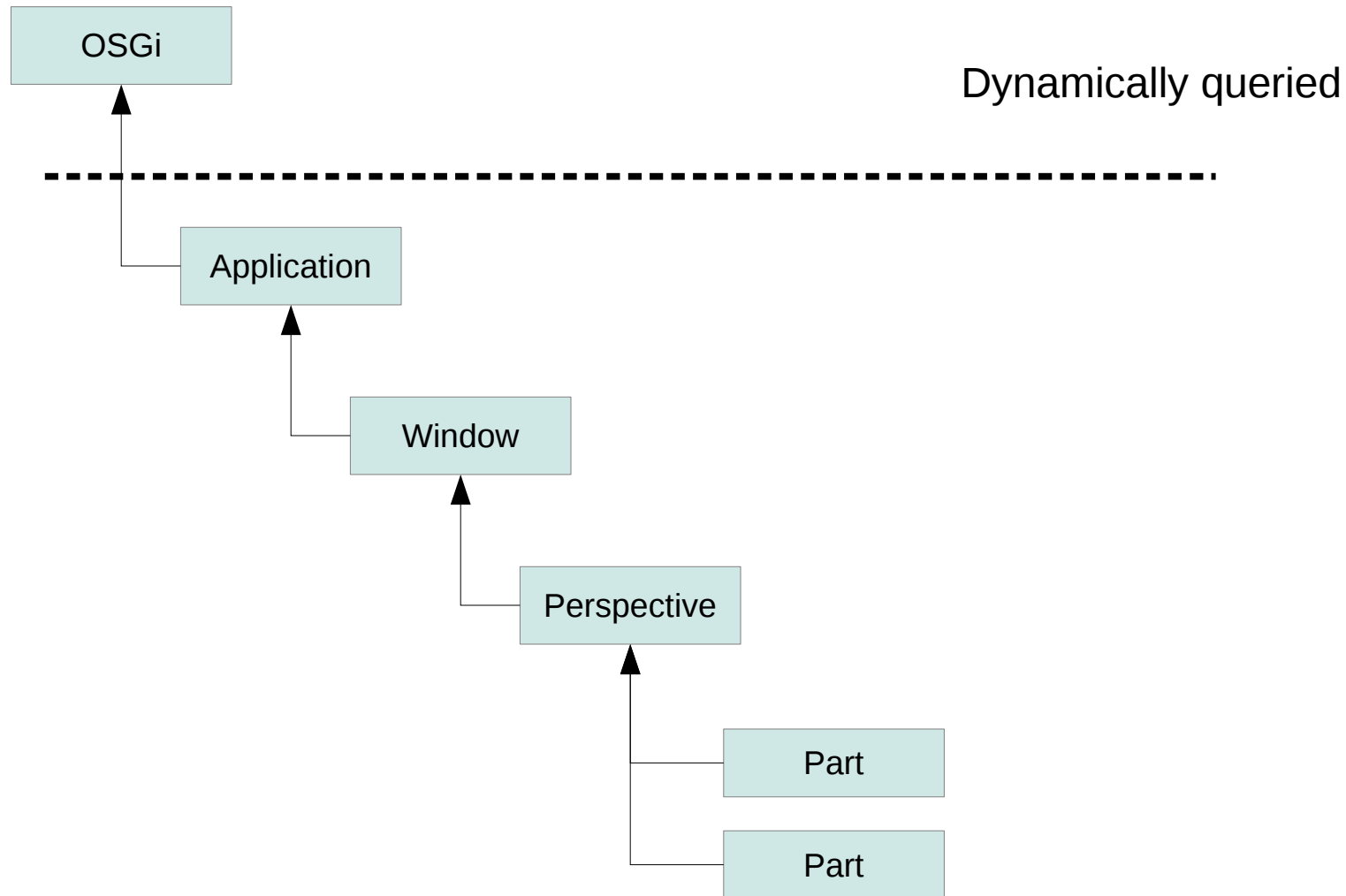
# Implementation

We write a service which  
can modify a given  
context

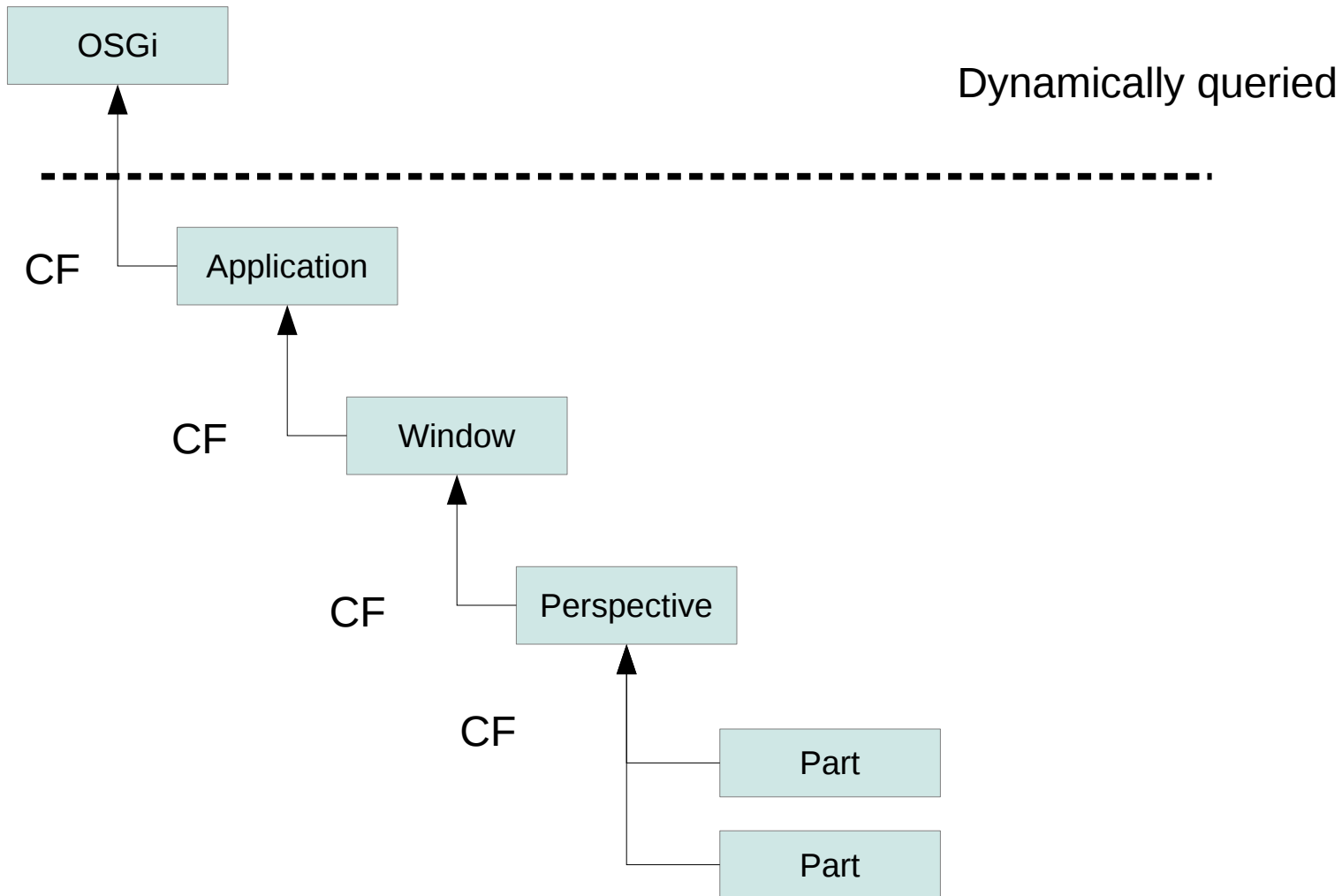


Joining the  
Eclipse  
framework  
with context  
functions

# Searching the context



# Context Functions



# Dependency injection with Context Functions



CF registers for a type and has access to the requesting context

# Model Add-ons



Use a model add-on to register  
the context function for the new  
service

Demo application

# Exercise

Questions?

# Evaluate This Session

- 1 Sign-in: [www.eclipsecon.org](http://www.eclipsecon.org)
- 2 Select session from schedule
- 3 Evaluate: 