



Content (1)
Creational design patterns

1 Façade

www.elqoo.com

PAGE 5



Content (1)
Creational design patterns

1 Façade

2 Adapter

www.elqoo.com

PAGE 6



Content (1)

Creational design patterns

PAGE 7

- 1 Façade
- 2 Adapter
- 3 Decorator

www.elqoo.com



Content (1)

Creational design patterns

PAGE 8

- 1 Façade
- 2 Adapter
- 3 Decorator
- 4 Bridge

www.elqoo.com



Content (1)

Creational design patterns

PAGE 9

- 1 Façade
- 2 Adapter
- 3 Decorator
- 4 Bridge
- 5 Composite

www.elqoo.com



Content (2)

Creational design patterns

PAGE 10

- 6 Proxy

www.elqoo.com

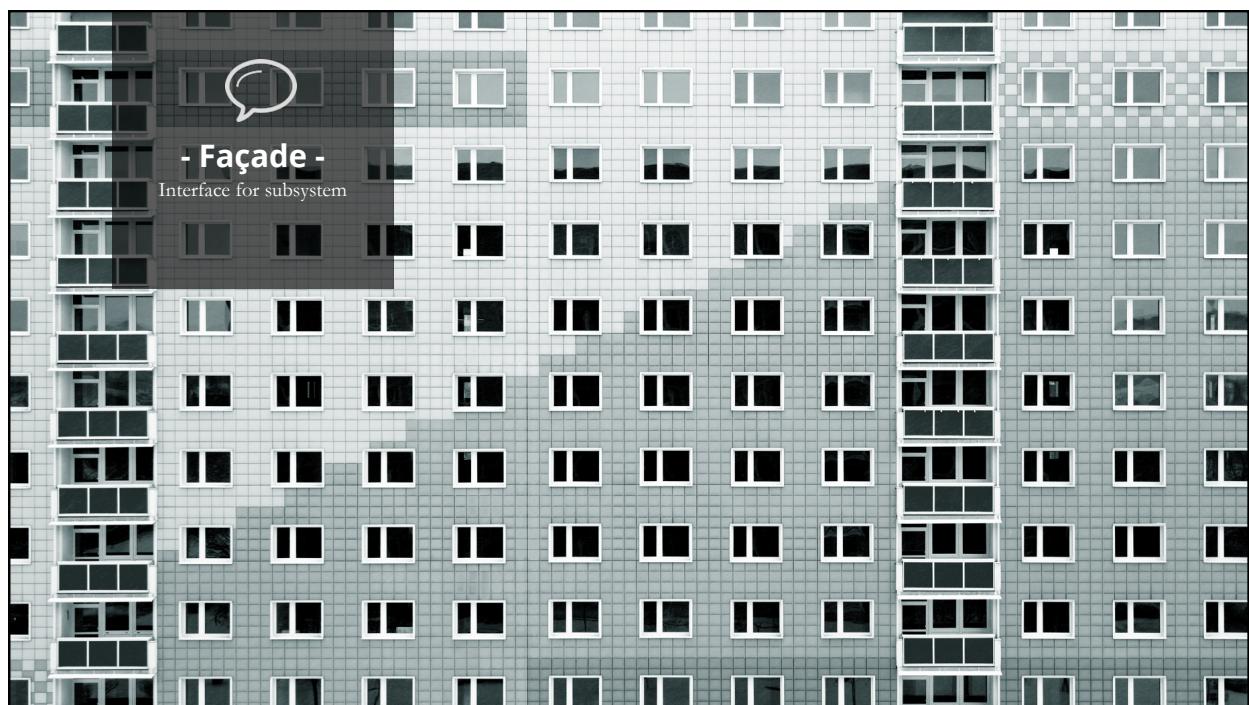
PAGE 11

Content (2)

Creational design patterns

- 6 Proxy
- 7 Flyweight

www.elqoo.com



.....

Problem Statement

PAGE 13


SD
Brad

- Hello Brad
- Hi Suzy
- We need an extra mobile client for our financial system.
- Okido


PM
Suzy

Elqoo

www.elqoo.com

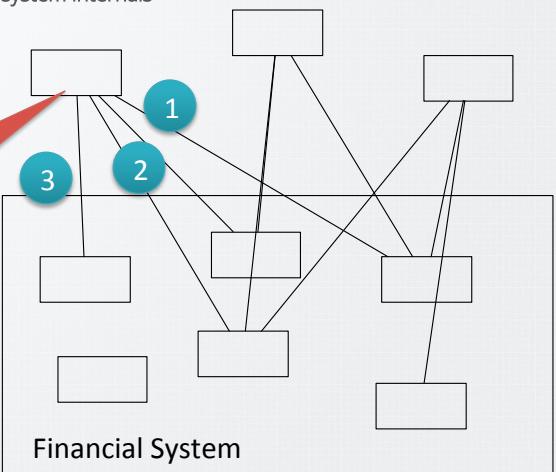
.....

Problem Statement Overview

Clients must be aware of subsystem internals

PAGE 14

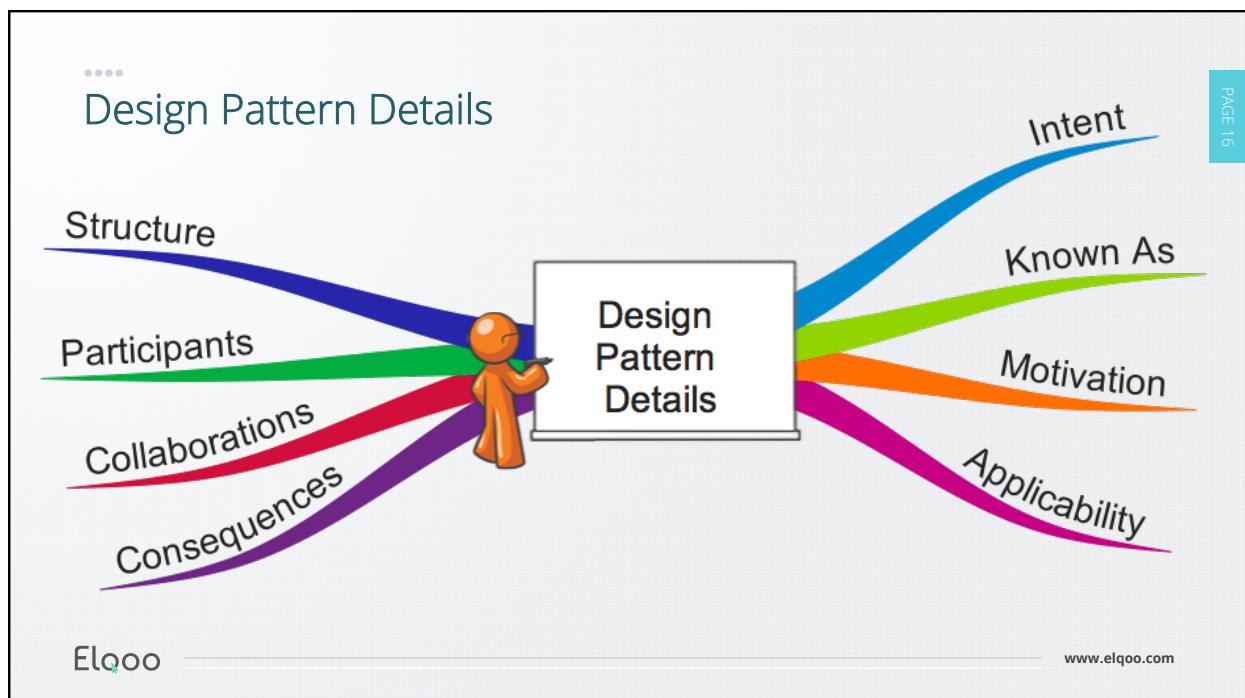
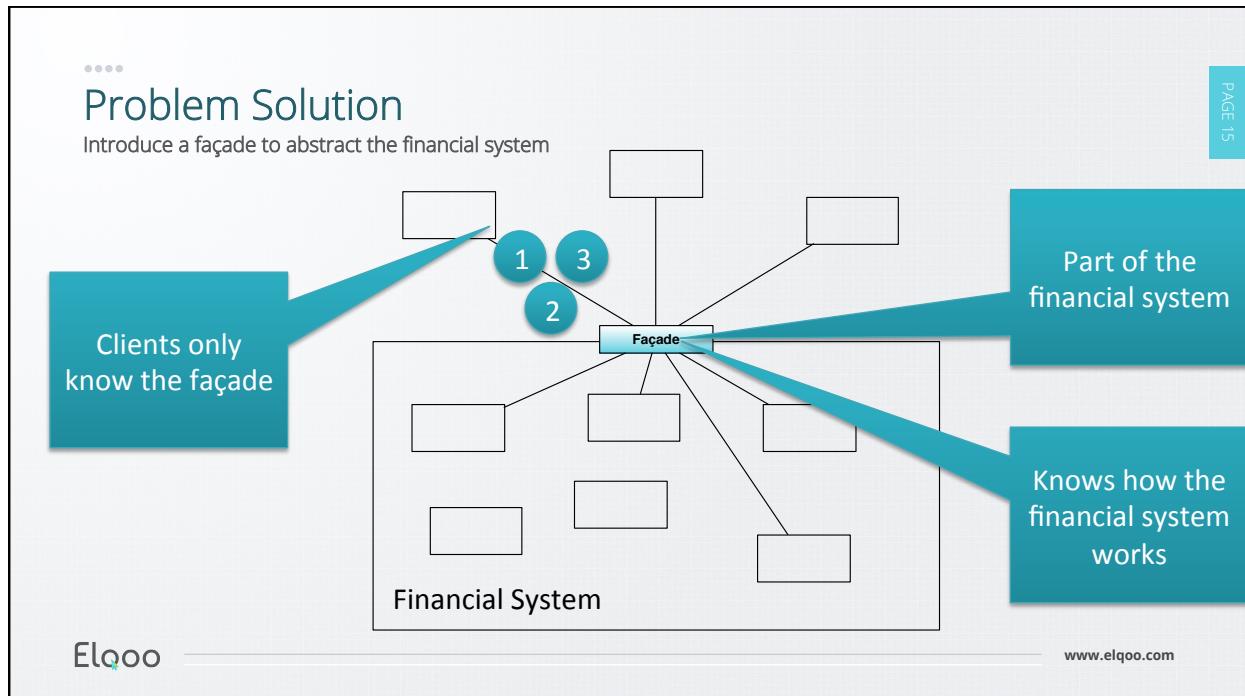
Clients need to know the subsystem



The diagram illustrates a complex subsystem structure. It features several rectangular boxes representing components or modules. Three specific nodes are highlighted with teal circles and numbered 1, 2, and 3. Node 1 is at the top left, node 2 is below it, and node 3 is further down and to the left. Lines connect these numbered nodes to other parts of the subsystem, showing their interconnectedness. A red arrow points from a red box containing the text "Clients need to know the subsystem" towards node 3.

Elqoo

www.elqoo.com



Façade Pattern

Intent and known as

PAGE 17



Intent

Provide a **unified interface** to a set of interfaces in a subsystem. Façade defines a **higher-level interface** that makes the **subsystem easier to use**.

Elqoo

www.elqoo.com

Apply Façade Pattern

Applicability

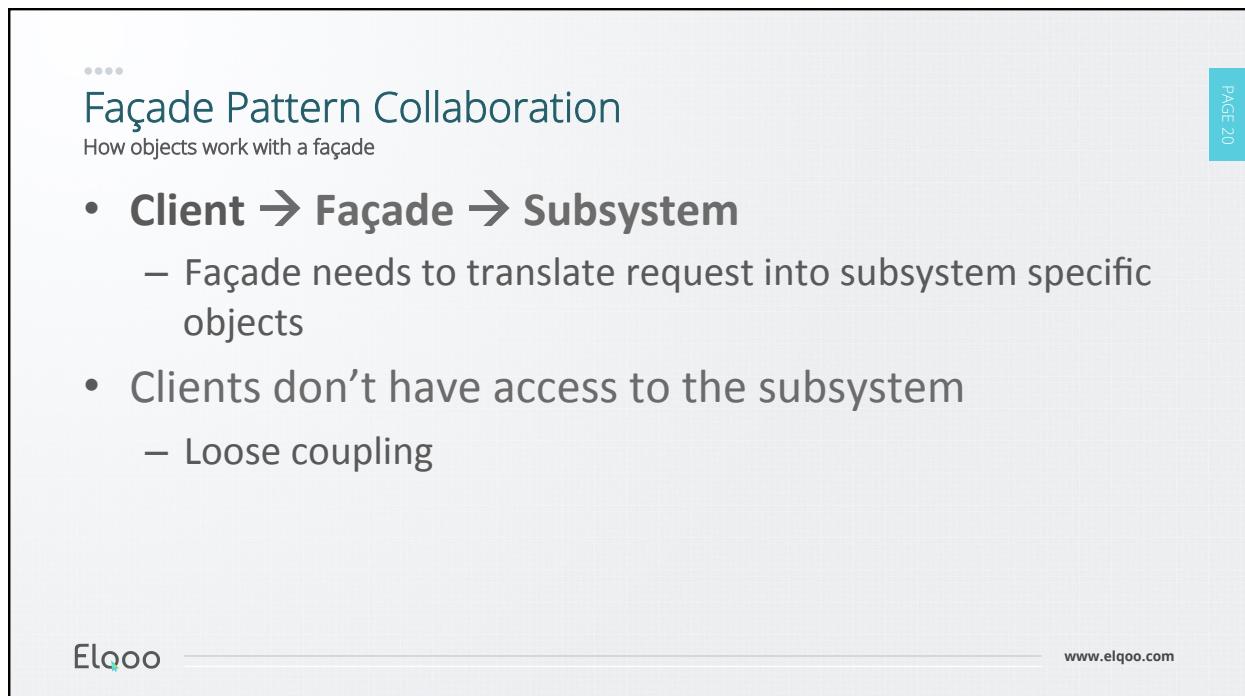
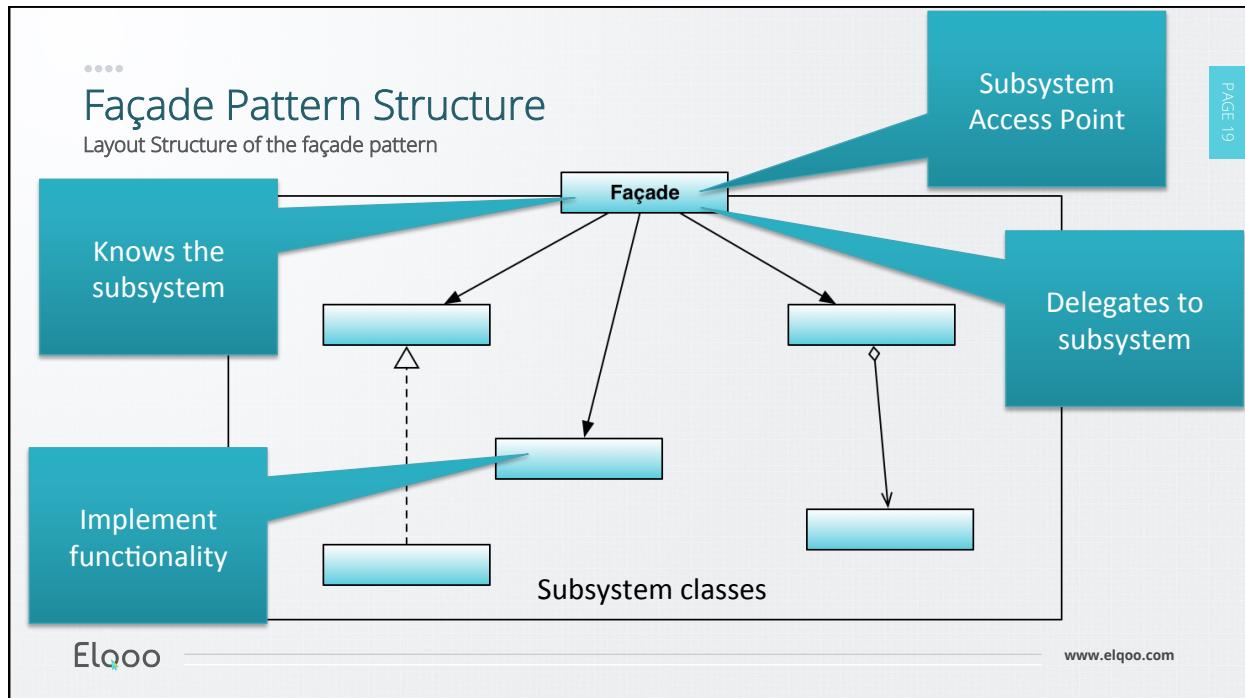
PAGE 18

- **Use**

- **Decouple** clients from subsystems
- Provide **simple interface**
- Subsystem layering (business, data and client services)

Elqoo

www.elqoo.com



..... Façade Pattern Consequences

Benefits of the façade pattern

- **Benefits**

- Subsystem **easier** to use
 - Client don't require specific knowledge
- Loose coupling
- Subsystem **can still be used directly** (if necessary)

- **Drawbacks**

- Façade introduces an extra programming layer

..... Conclusion

- **Façade pattern is great**

- Abstract complex system from client
- Provide extra loose coupling



.....

Problem Statement

PAGE 24

SD

Brad

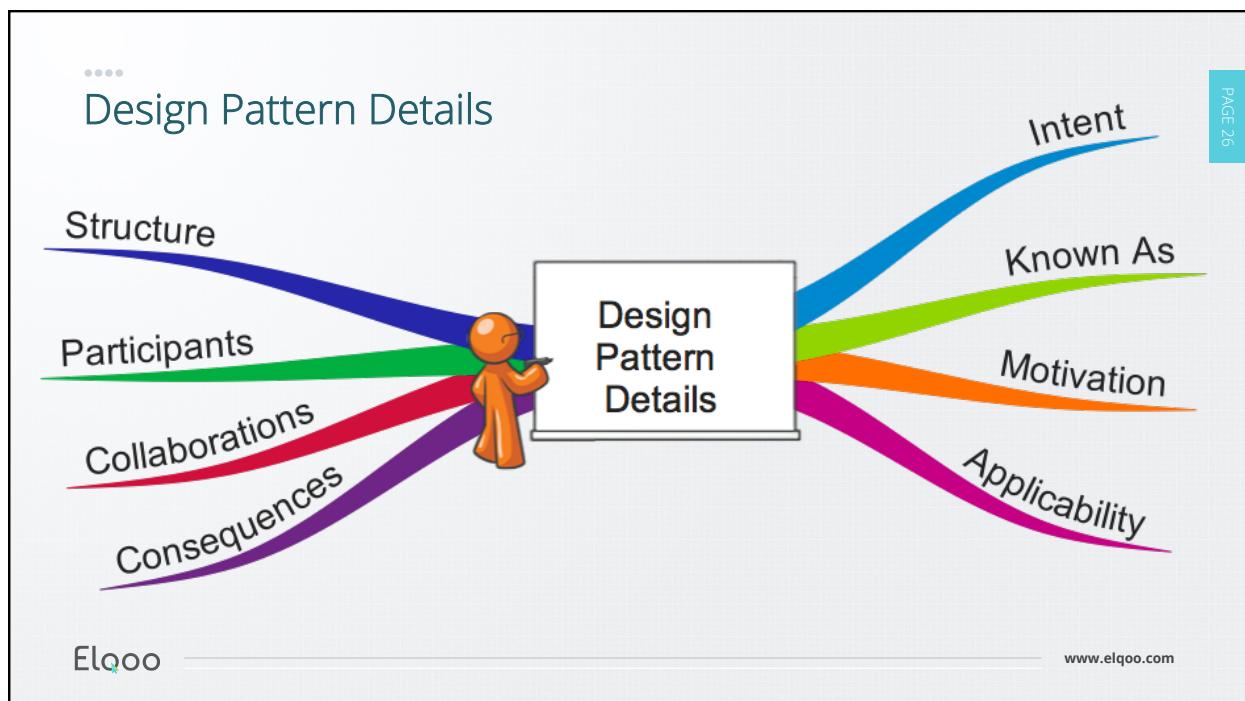
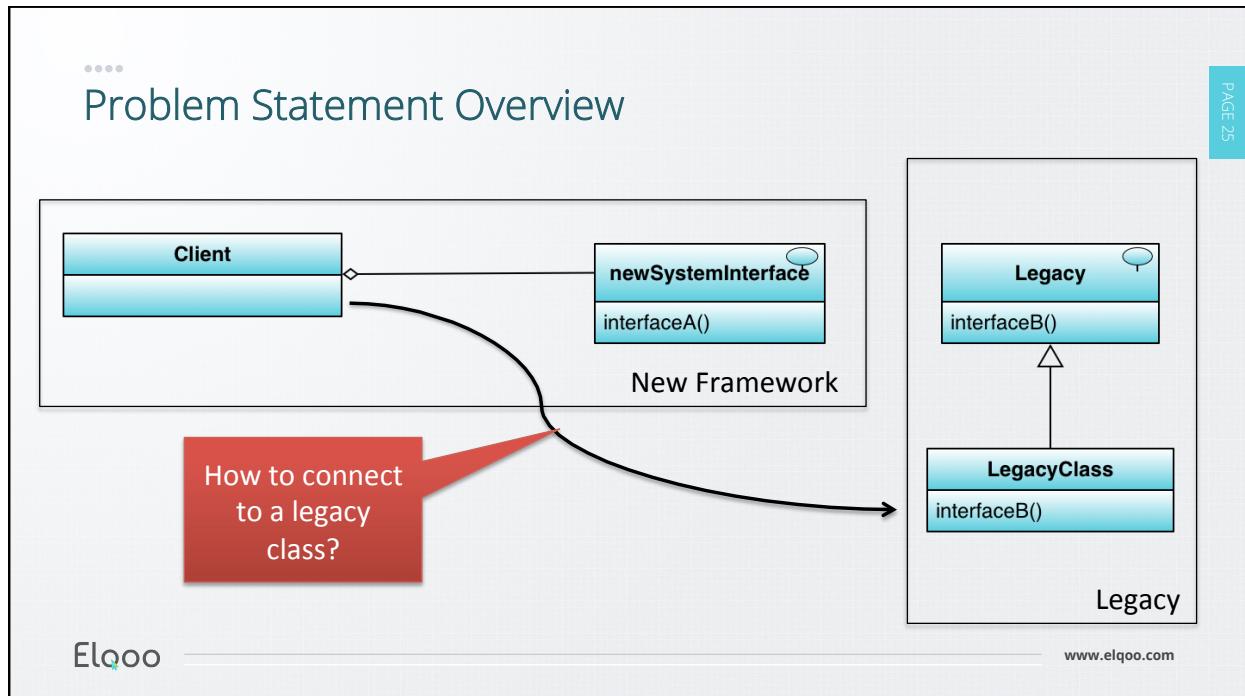
- Hello Brad
- Hi Suzy
- We want to connect to our old legacy systems, but there're not conform or standards. Can it be done?
- I'll have to investigate

PM

Suzy

Elqoo

www.elqoo.com



Adapter Pattern

Intent and known as

PAGE 27



Intent

Convert the interface of a class into another interface the clients expect. Adapter **lets classes work together** that couldn't otherwise because of incompatible interfaces.

Known As

Wrapper

Elqoo

www.elqoo.com

Apply Adapter Pattern

Applicability

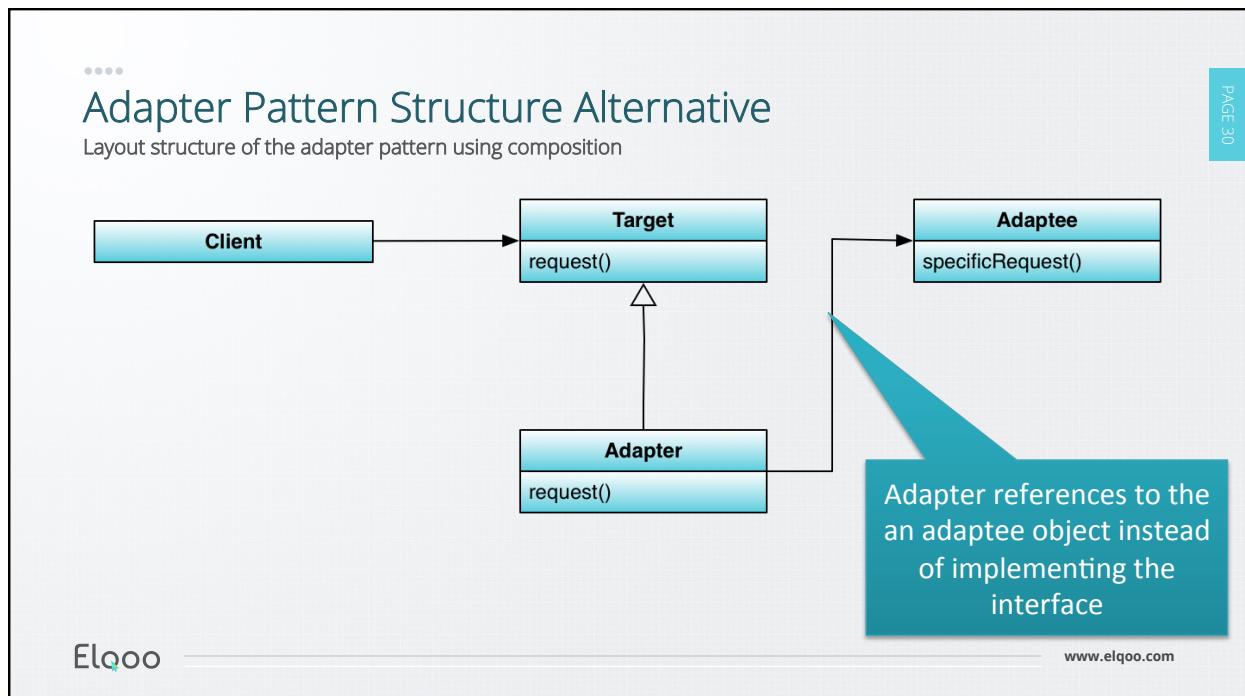
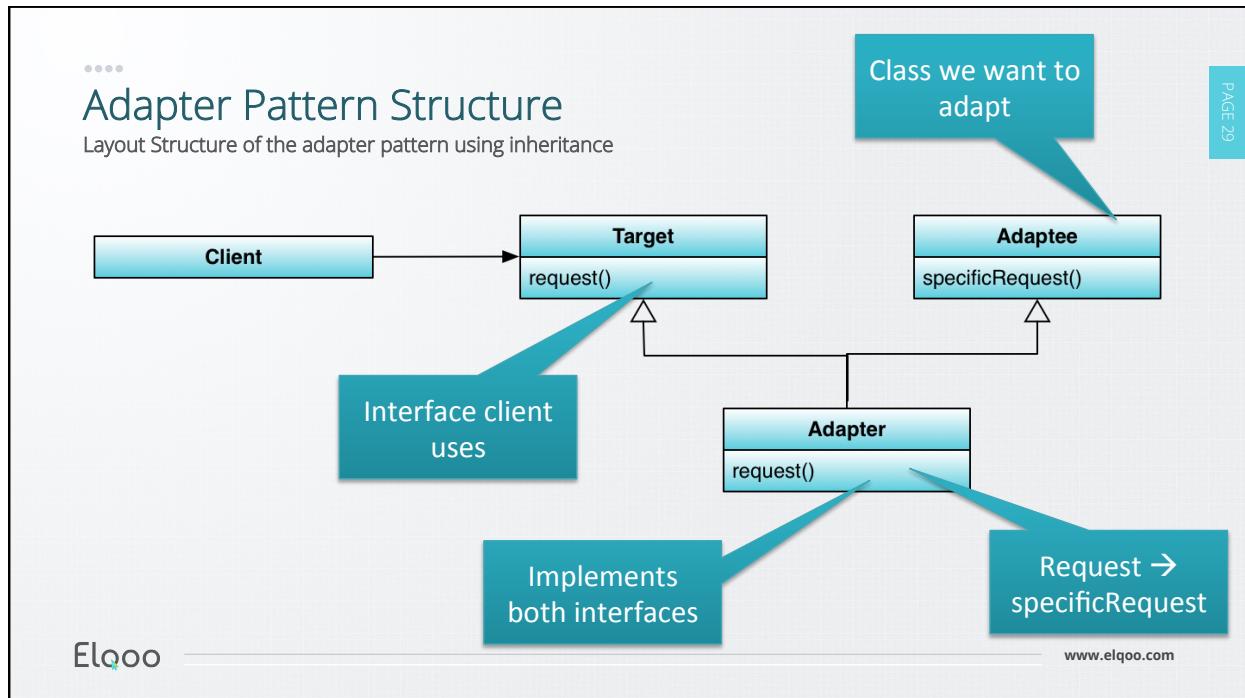
PAGE 28

- **Use**

- **Re-use** an existing class
- **Combine** unrelated **classes** with an incompatible interface

Elqoo

www.elqoo.com



Adapter Pattern Collaboration

Collaboration between objects

- Adapter **forwards request** to adaptee
 - This can be adapter itself (**inheritance**)
 - This can be a **reference** of the adaptee

Adapter Pattern Consequences (1)

Considerations for the adapter pattern

- **Benefits**
 - Adapter can **override** adaptee **behaviour** → it is a subclass
 - One adapter → Many adaptees
- **Drawbacks**
 - Adapter doesn't work for
 - Class with many subclasses → can't extend them all
 - Harder to override adaptee behaviour (do we want this?)

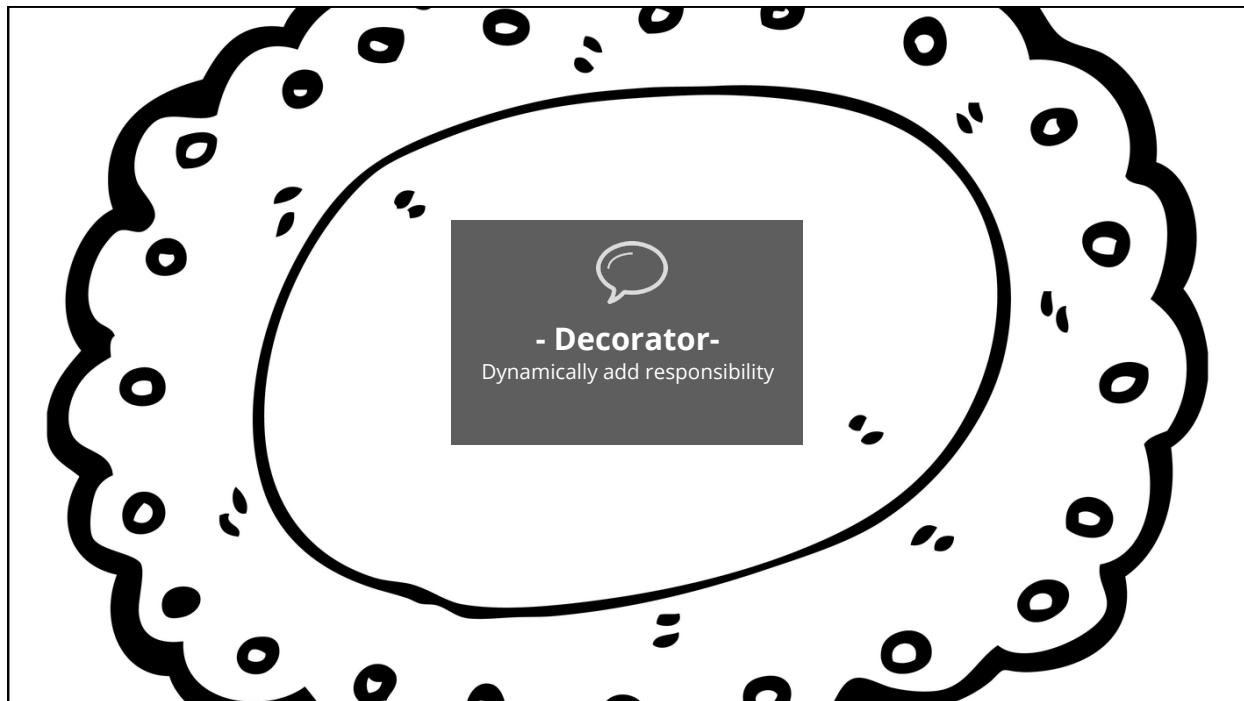
.... Adapter Pattern Consequences (2)

Considerations for the adapter pattern

- How much adapting does the adapter do?
 - **Complex adaptations** will be hard to write.

.... Conclusion

- **Adapter pattern is great**
 - Connect to legacy system
 - Adapt one interface to another



.....

Problem Statement

PAGE 36

SD

Brad

- Hello Brad
- Hi Suzy
- We would like to have a framework that supports a visual window with a scrollbar and an icon.
- This is easy

PM

Suzy

Elqoo

www.elqoo.com

.....

Problem Statement Overview

PAGE 37

```

classDiagram
    class Window
    class IconWindow
    class ScrollBarWindow
    class ScrollBarIconWindow
    Window <|-- IconWindow
    Window <|-- ScrollBarWindow
    Window <|-- ScrollBarIconWindow
    
```

For each functionality a separate class

Can we avoid creating this class?

Framework

Elqoo

www.elqoo.com

.....

Problem Solution (1)

PAGE 38

```

classDiagram
    class Window
    class WindowDecorator
    class WindowScrollBarDecorator
    class WindowIconDecorator
    Window <|-- WindowDecorator
    WindowDecorator <|-- WindowScrollBarDecorator
    WindowDecorator <|-- WindowIconDecorator
    
```

Subclass decorator for added behavior

Decorator window → Reference window

Add icon

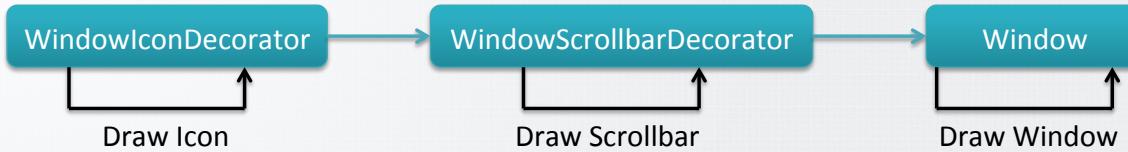
Elqoo

www.elqoo.com

Problem Solution (2)

PAGE 39

1. We need a window with an **icon** and a **scrollbar**



2. We need a window with **just** a **scrollbar**



Elqoo

www.elqoo.com

Design Pattern Details

PAGE 40

Structure

Participants

Collaborations

Consequences

Design
Pattern
Details

Intent

Known As

Motivation

Applicability

Elqoo

www.elqoo.com

.... Decorator Pattern

Intent and known as

PAGE 41



Intent

Attach additional responsibilities to an object dynamically. Decorators provide a **flexible alternative to subclassing** for extending functionality

Known As

Wrapper

Elqoo

www.elqoo.com

.... Apply Decorator Pattern

Applicability

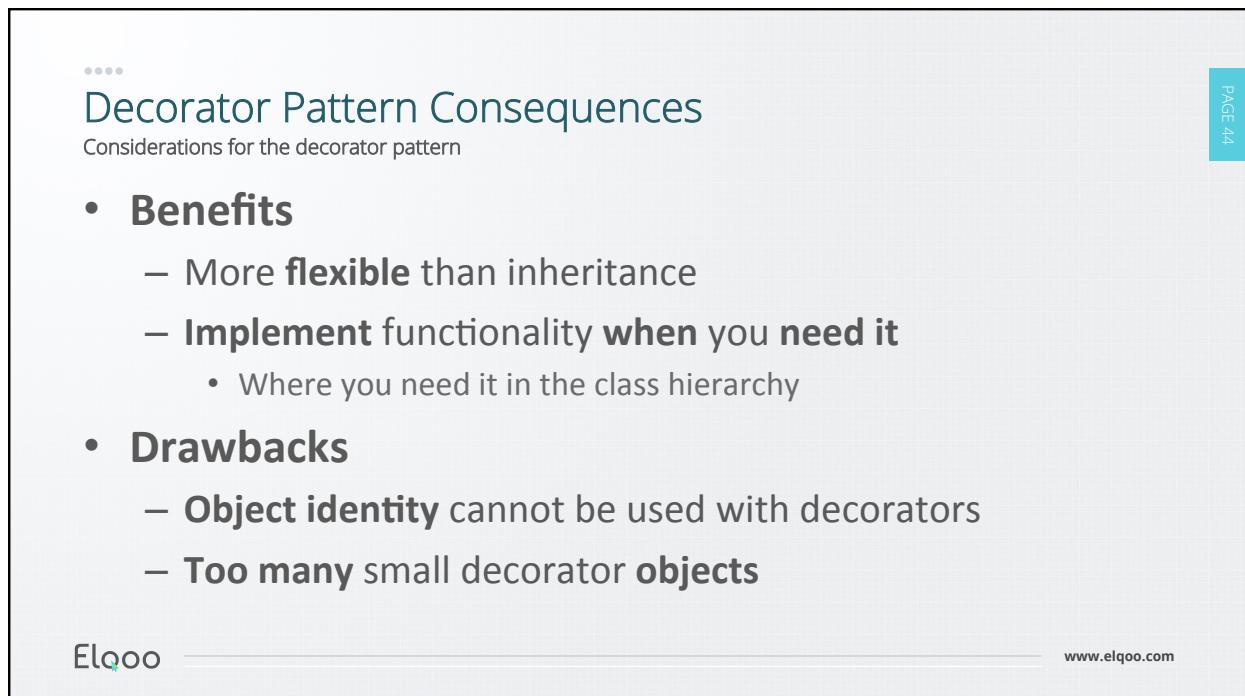
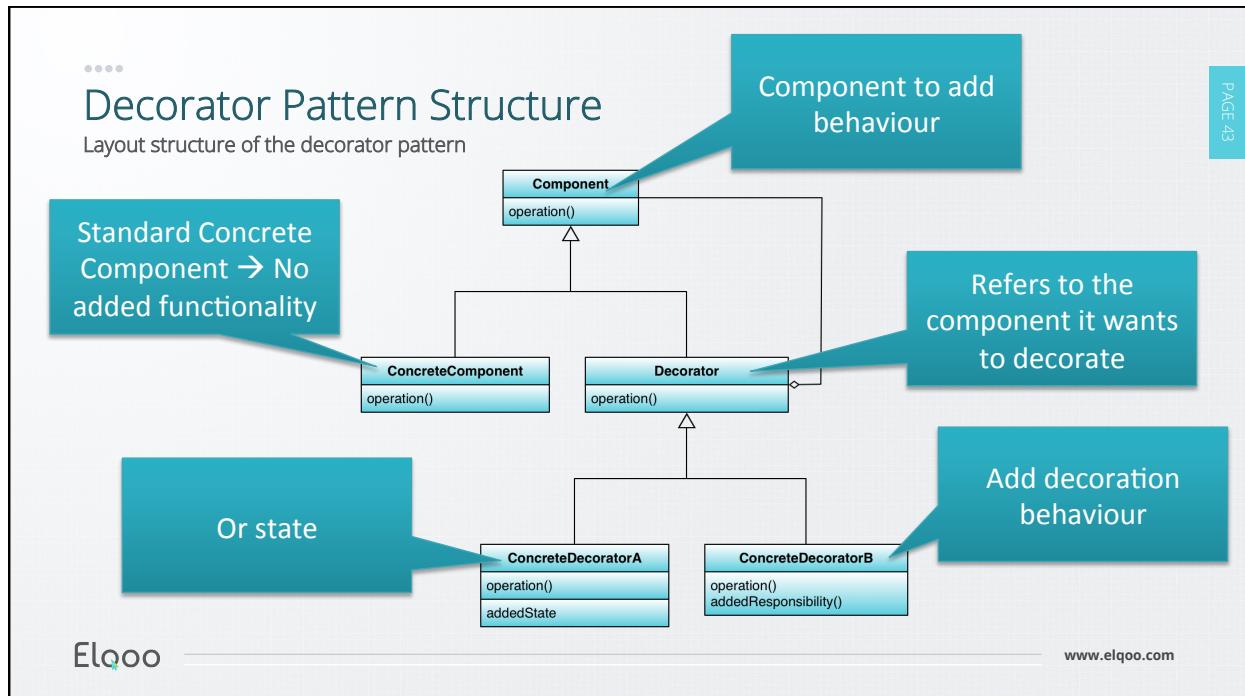
PAGE 42

- **Use**

- Add **functionality** to objects, without affecting other objects
- Functionalities can be taken away in the future
- **Extension by subclassing is difficult**

Elqoo

www.elqoo.com



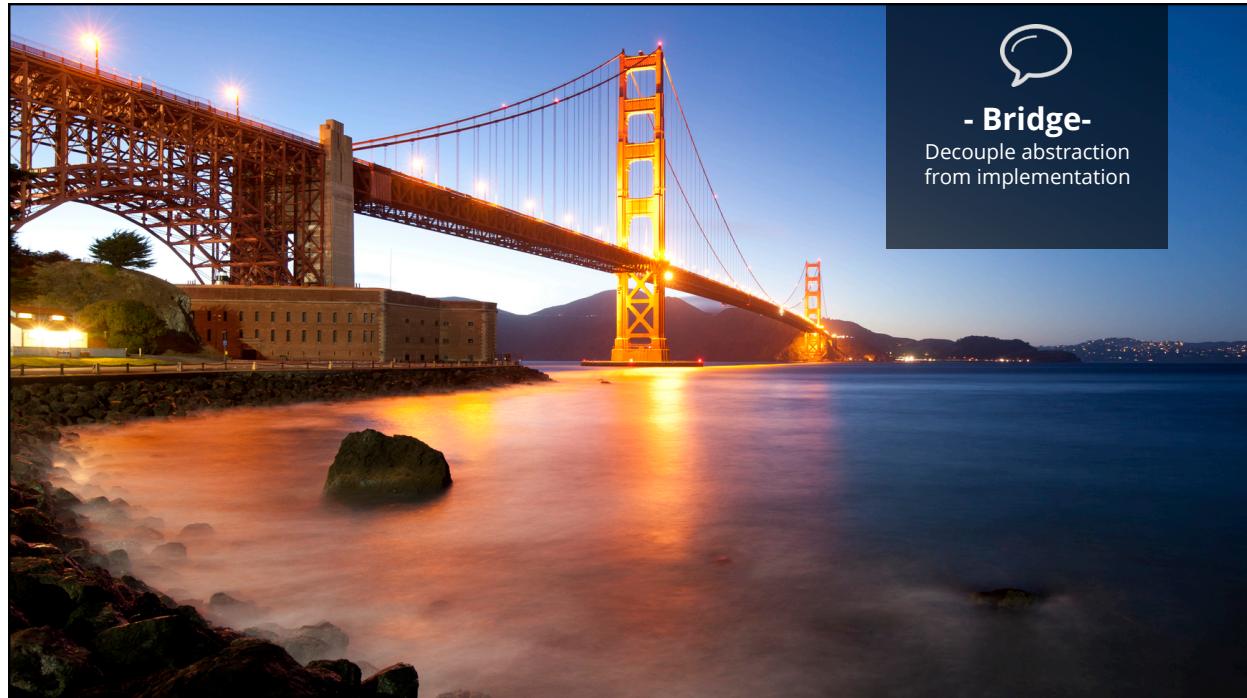
Conclusion

PAGE 45

- **Decorator pattern is great**
 - Add behavior to objects
 - Higher flexibility than extension

Elqoo

www.elqoo.com



.....

Problem Statement

PAGE 47


SD
Brad

- Hello Brad
- Hi Suzy
- We want to create a multiplatform window toolkit
- Sounds a challenge


PM
Suzy

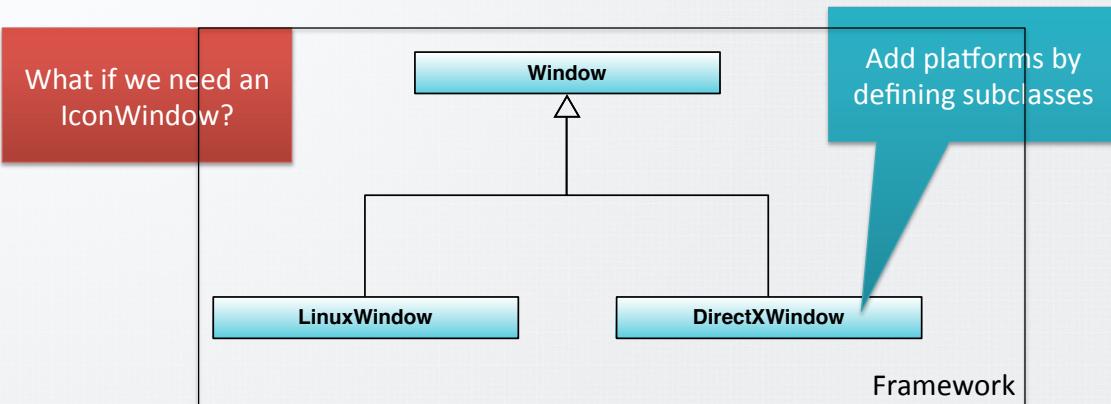
Elqoo

www.elqoo.com

.....

Problem Statement Overview

PAGE 48



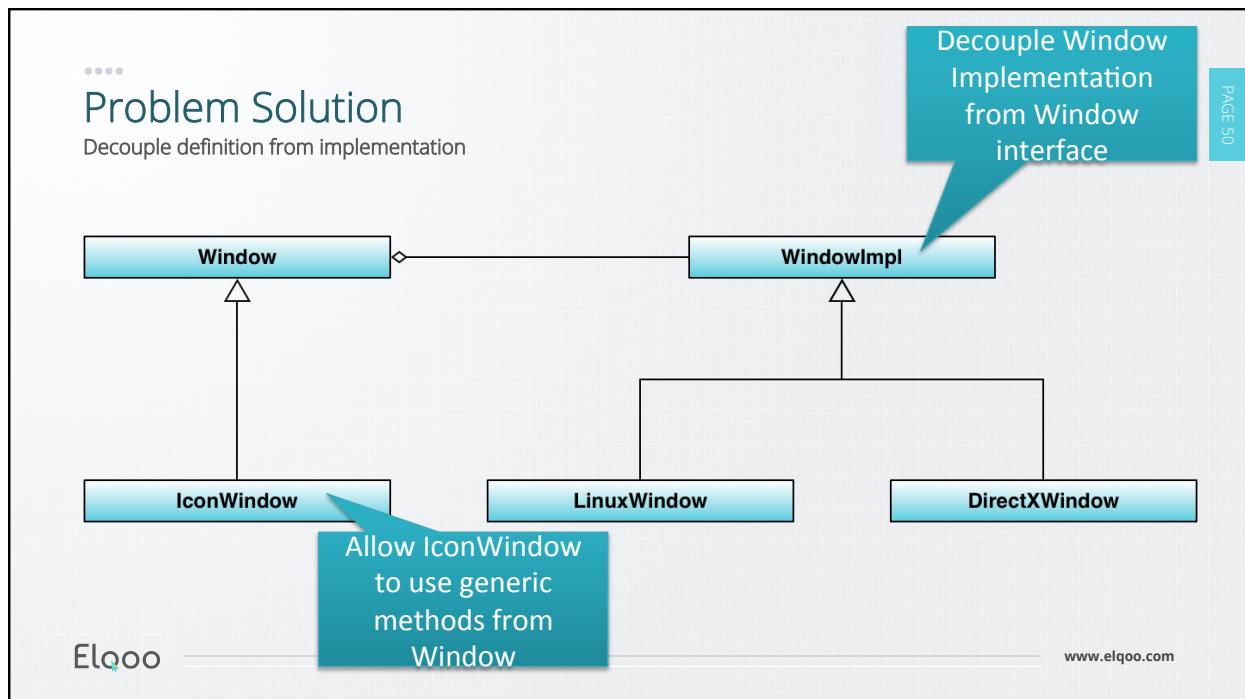
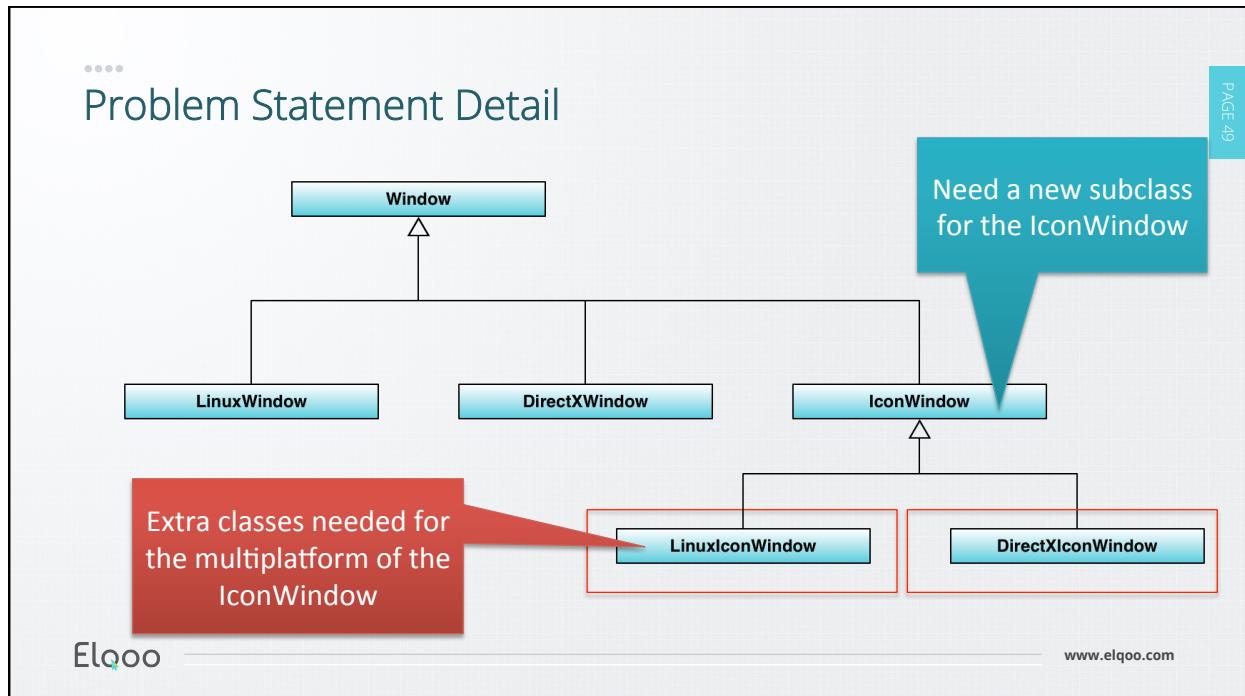
What if we need an IconWindow?

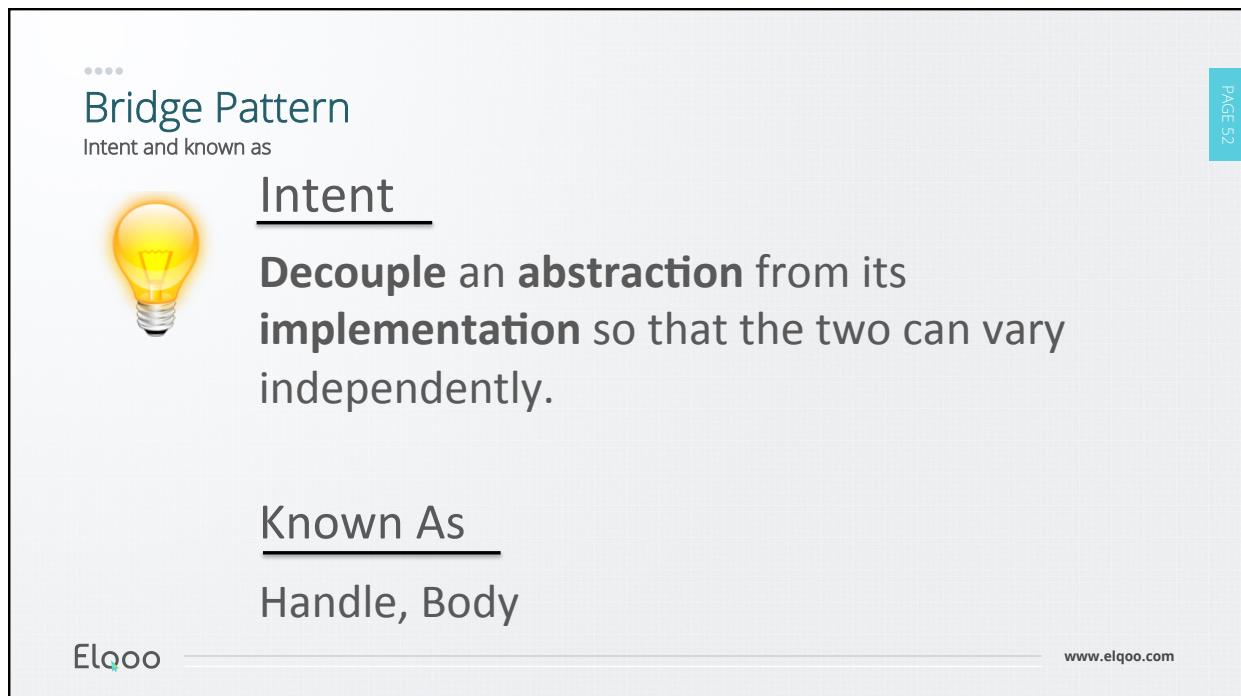
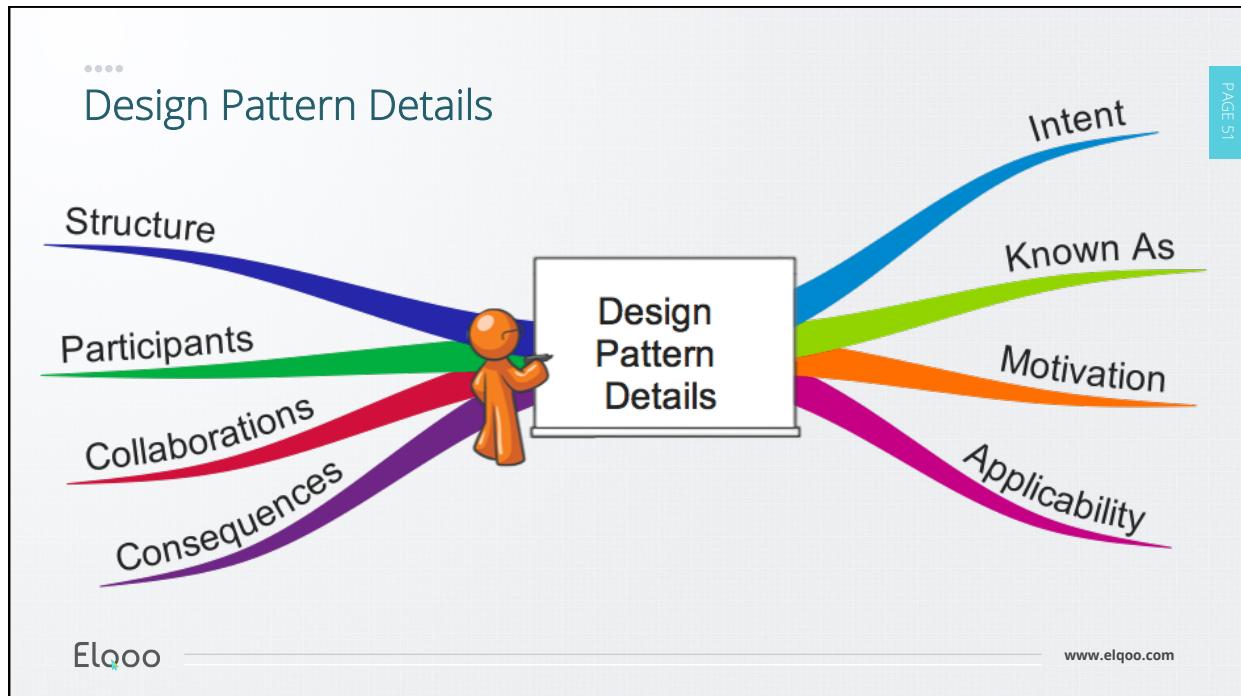
Add platforms by defining subclasses

Framework

Elqoo

www.elqoo.com





Apply Bridge Pattern

Applicability

PAGE 53

- **Use**

- **Avoid binding** between interface and implementation
- Possible subclasses for abstraction and implementation
- Must be possible to **change implementation at runtime** without affecting clients

Elqoo

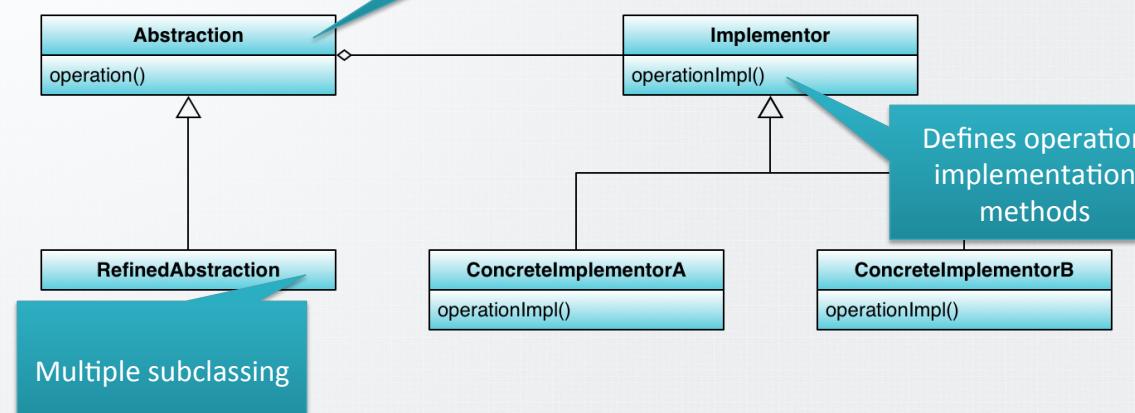
www.elqoo.com

Bridge Pattern Structure

Layout structure of the bridge pattern

PAGE 54

- Define abstraction interface
- Maintain reference to implementer



Elqoo

www.elqoo.com

Bridge Pattern Consequences

Considerations for the bridge pattern

- **Benefits**

- Decouple interface and implementation
- Improve **extensibility**
- Hide implementation details

Conclusion

- **Bridge pattern is great**

- Decouple interface from implementation
- Reduce nr of subclasses



.....

Problem Statement

PAGE 58



SD

Brad

- Hello Brad
- Hi Suzy
- We want to be able to group objects in our graphical user interface
- No problem



PM

Suzy

Elqoo

www.elqoo.com

.....

Problem Statement Overview

User interface data structure

```

graph TD
    Node1[Node] --- Node2[Node]
    Node1 --- Node3[Node]
    Node1 --- Node4[Node]
    Node2 --- Leaf1[Leaf]
    Node2 --- Leaf2[Leaf]
    Node3 --- Leaf3[Leaf]
    Node4 --- Leaf4[Leaf]
  
```

PAGE 59

Elqoo

www.elqoo.com

.....

Problem Statement Overview (2)

Different treatment for Rectangle , Line and Group

Group must link to each Rectangle and Line

Client

Leaf

Rectangle

Line

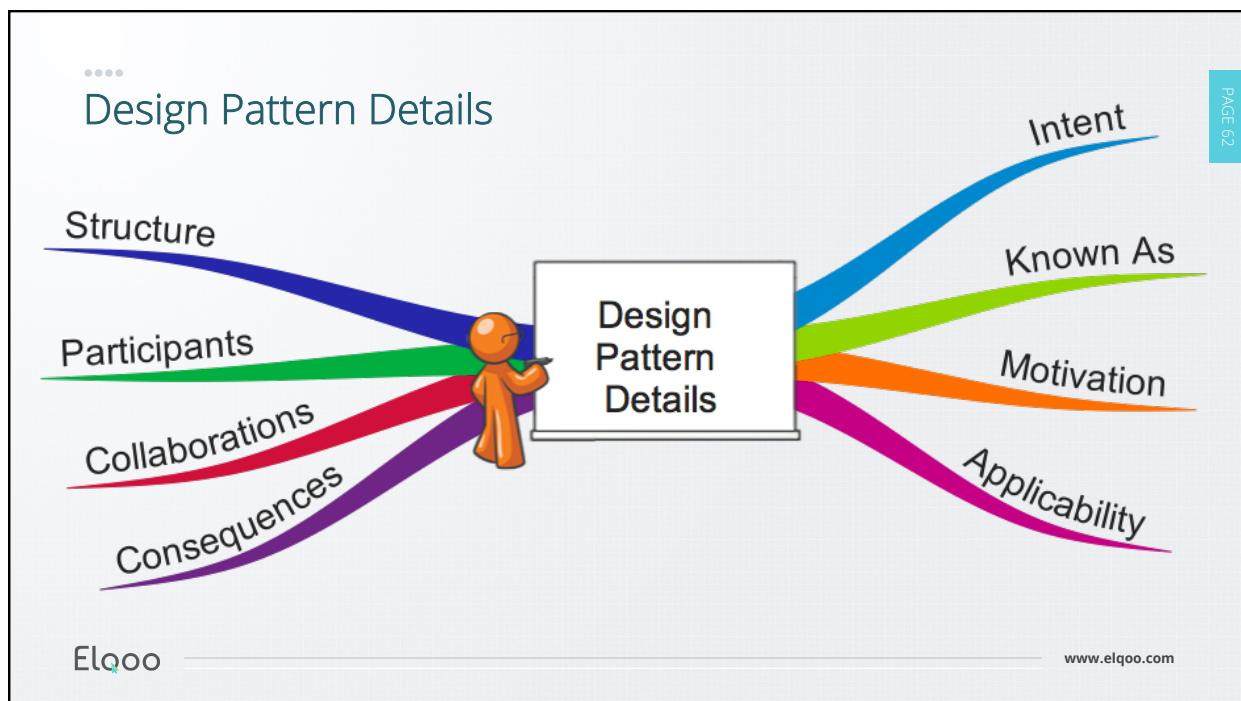
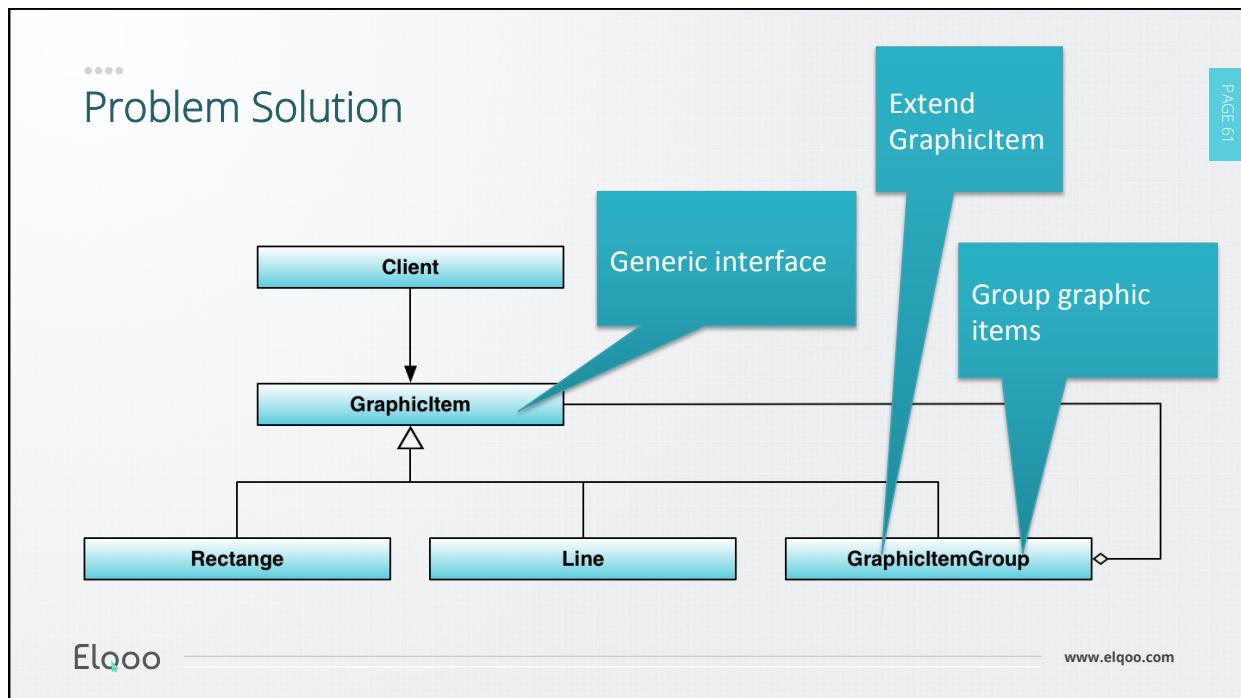
GraphicItemGroup

Node

Elqoo

www.elqoo.com

PAGE 60



.... Composite Pattern

Intent and known as

PAGE 63



Intent

Compose **objects** into **tree structures** to represent part-whole hierarchies. Composite lets clients **treat individual objects** and compositions of objects **uniformly**

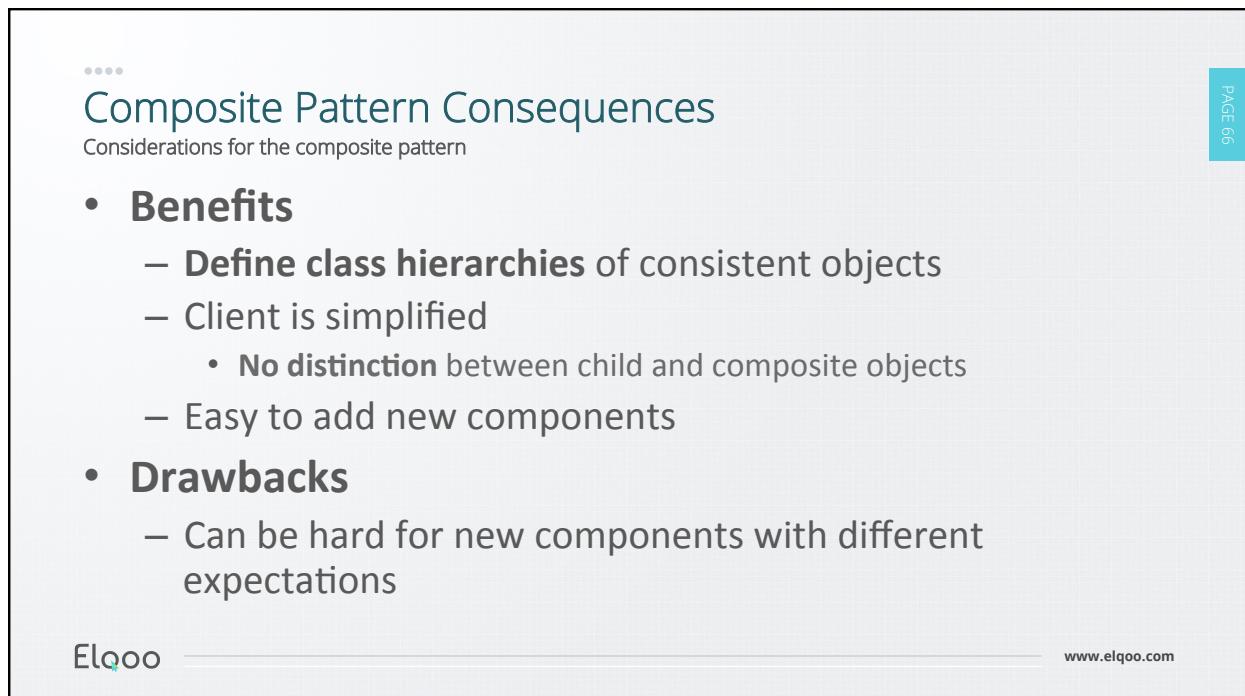
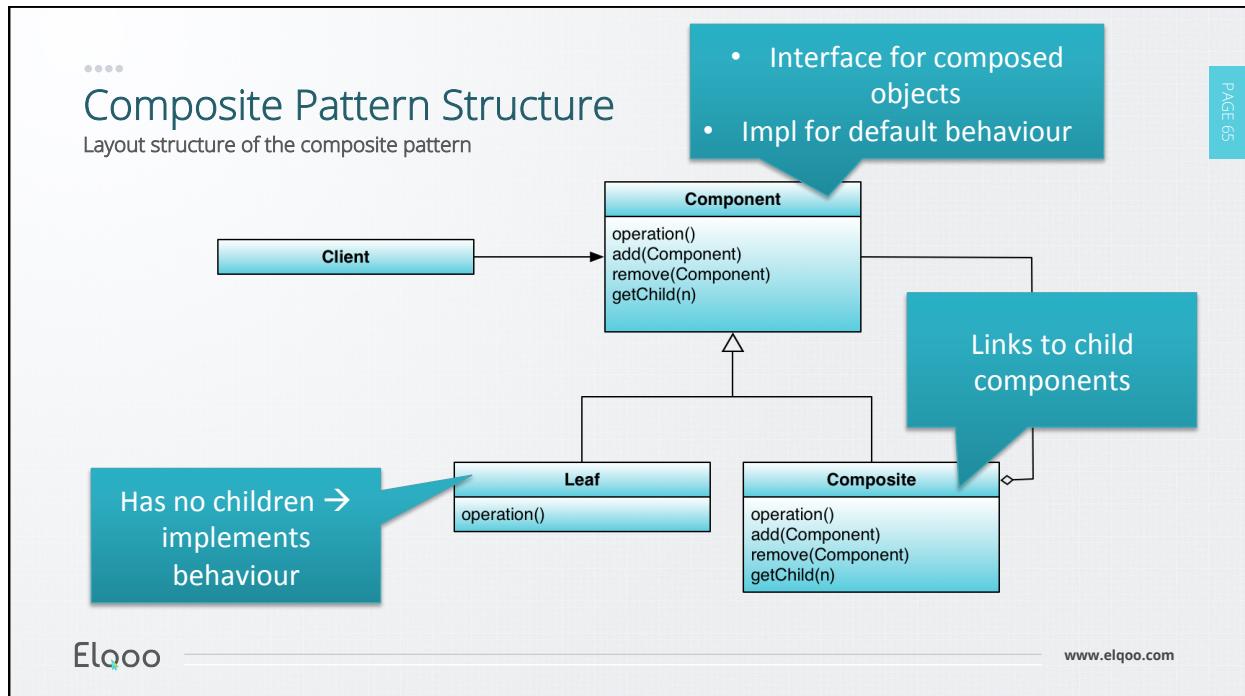
.... Apply Composite Pattern

Applicability

PAGE 64

- **Use**

- Ignore differences between **compositions** and **individual items**
- Represent part-whole **hierarchies of objects**



Conclusion

PAGE 67

- **Composite pattern is great**
 - Represent part-whole hierarchies of objects
 - Abstraction from client

Elqoo

www.elqoo.com



- Proxy-
Placeholder for objects



.....

Problem Statement

PAGE 69



SD

Brad

- Hello Brad
- Hi Suzy
- We want to add on-demand loading to our graphics editor
- It might be a lot of work to do, but I'll look into it.



PM

Suzy

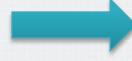
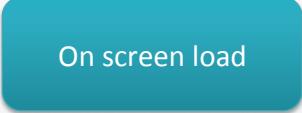
Elqoo

www.elqoo.com

.....

Problem Statement Overview

PAGE 70



On screen load → Loading →

Elqoo

www.elqoo.com

.....

Problem Statement Detail

PAGE 71

Takes too much time to load all images out of the box

The diagram illustrates a problem statement. At the top, a red box contains the text "Takes too much time to load all images out of the box". Below this, two blue rectangular boxes represent components: "Application" on the left and "Image" on the right. A horizontal arrow points from the Application box to the Image box, labeled "images" above it. Red downward-pointing arrows originate from the bottom of each component box and point towards a central area between them. The left arrow is associated with the text "Application requires images" in a red box below it. The right arrow is associated with the text "Pre-loads out of the box" in a red box below it.

Application

Image

Application requires images

Pre-loads out of the box

Elqoo

www.elqoo.com

.....

Problem Solution

PAGE 72

Loads image when drawing is requested

The diagram illustrates a solution using a proxy pattern. It features a large teal rounded rectangle labeled "Proxy" at the bottom. Inside this rectangle is a smaller teal rounded rectangle labeled "Image". A speech bubble originates from the right side of the "Proxy" box and contains the text "Loads image when drawing is requested". Red downward-pointing arrows from the previous diagram now point to the "Proxy" box, indicating its role in managing image loading.

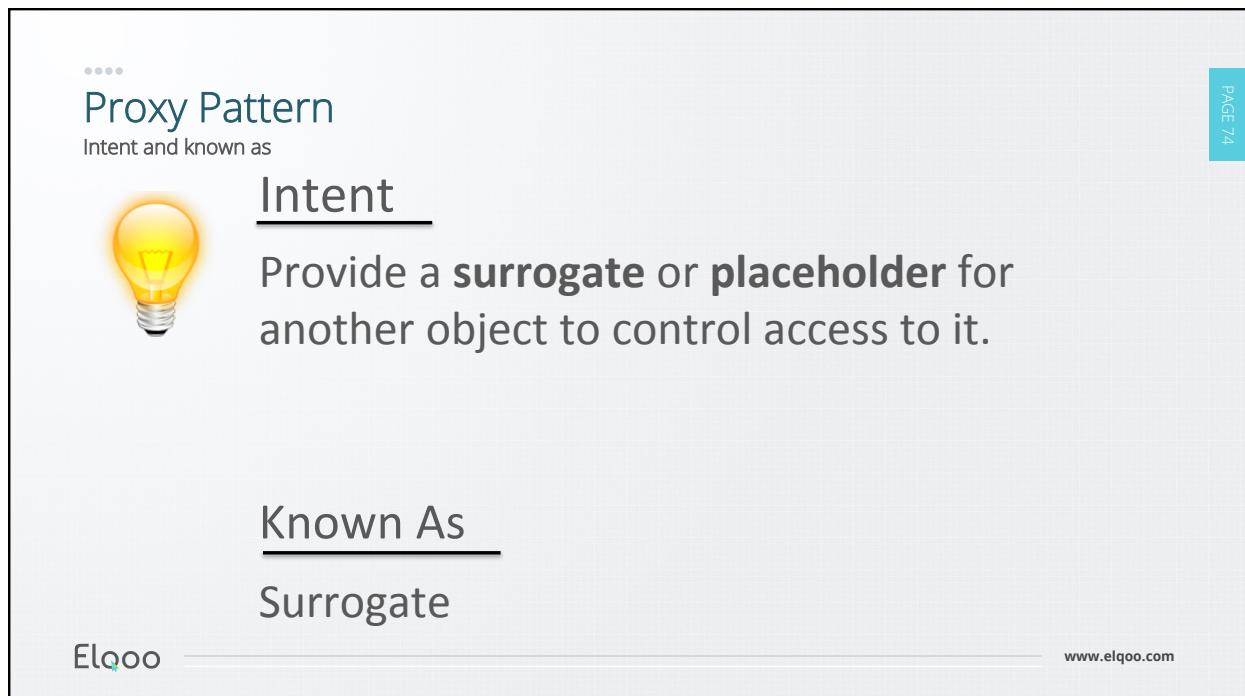
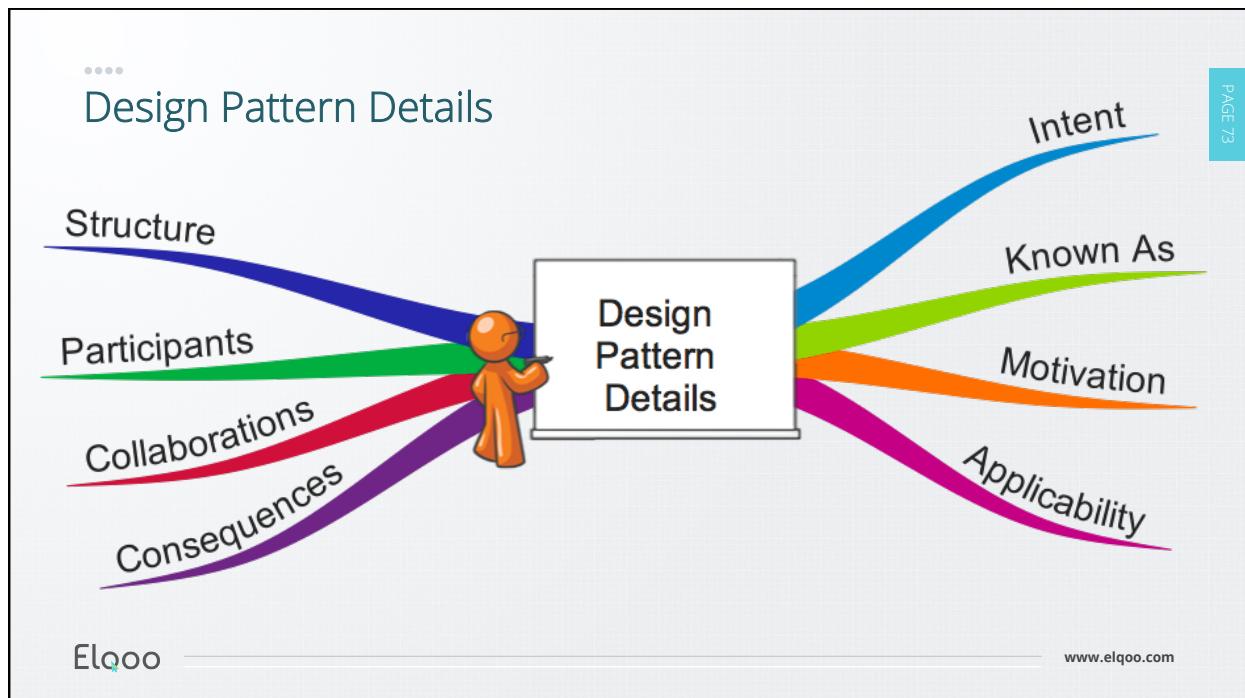
Proxy

Image

Loads image when drawing is requested

Elqoo

www.elqoo.com



Apply Proxy Pattern

Applicability

- **Use**

- Extra functionality is required
 - Transparency
 - More than just a reference

- **Example**

- Remote proxy (access network separated classes)
- Virtual Proxy (create expensive objects on demand)
- Protection proxy (access management)
- Added functionality e.g. count nbr of references

Elqoo

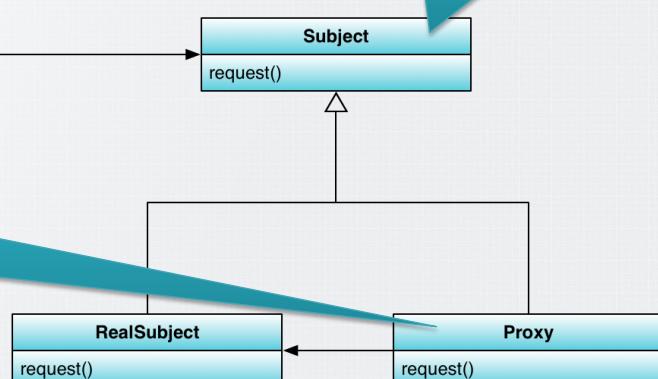
www.elqoo.com

Proxy Pattern Structure

Layout structure of the proxy pattern

- Implements subject interface
- Keeps reference to RealSubject
- Controls access

Common interface
for subjects



Elqoo

www.elqoo.com

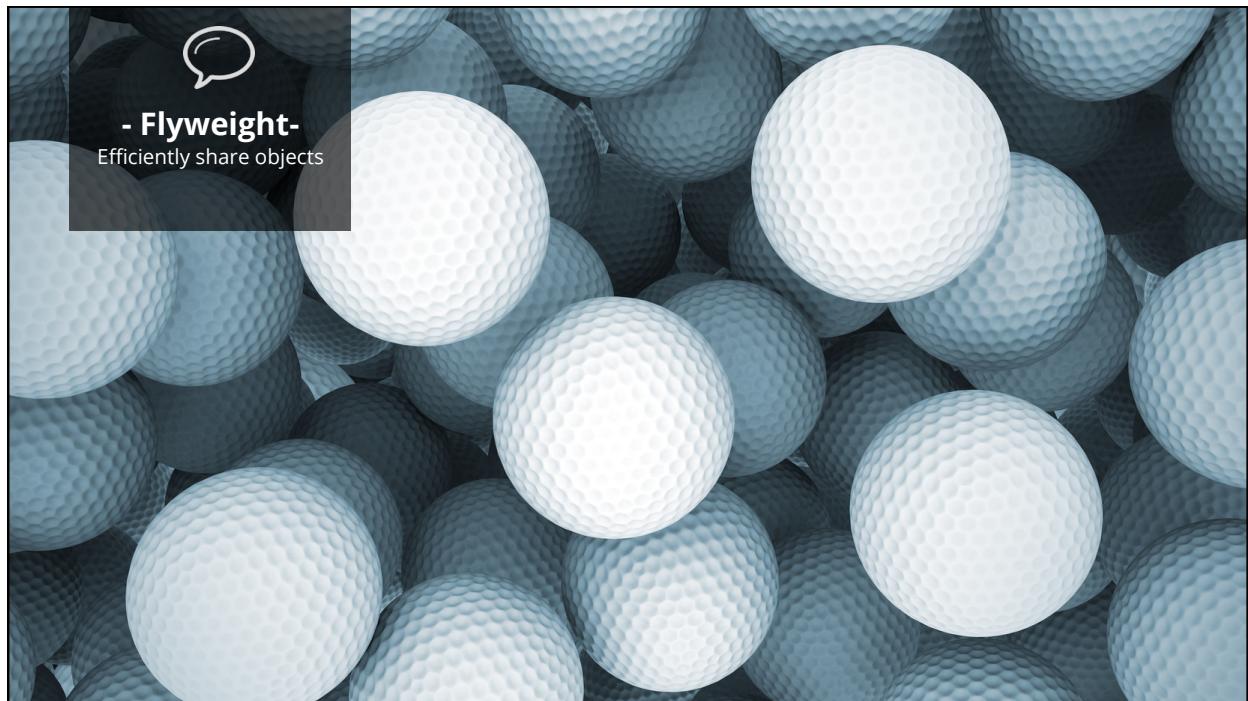
Proxy Pattern Consequences

Considerations for the proxy pattern

- Adds **indirection**
- Can be beneficial in many cases
 - Remote proxy
 - Virtual proxy
 - Smart references

Conclusion

- **Proxy pattern is great**
 - Transparently add functionality



.....

Problem Statement

PAGE 80



SD

Brad

- Hello Brad
- Hi Suzy
- Our text editor is running slow. Could you fix it.
- Sure



PM

Suzy

Elqoo

www.elqoo.com

..... Problem Statement Overview

PAGE 81

A B C D E F G

Each Letter = One object in memory

Serious Memory and performance overhead

Elqoo

www.elqoo.com

..... Design Pattern Details

PAGE 82



Structure

Participants

Collaborations

Consequences

Design
Pattern
Details

Intent

Known As

Motivation

Applicability

Elqoo

www.elqoo.com

Flyweight Pattern

Intent and known as

PAGE 83



Intent

Use Sharing to support large numbers of fine-grained objects efficiently.

Elqoo

www.elqoo.com

Apply Flyweight Pattern

Applicability

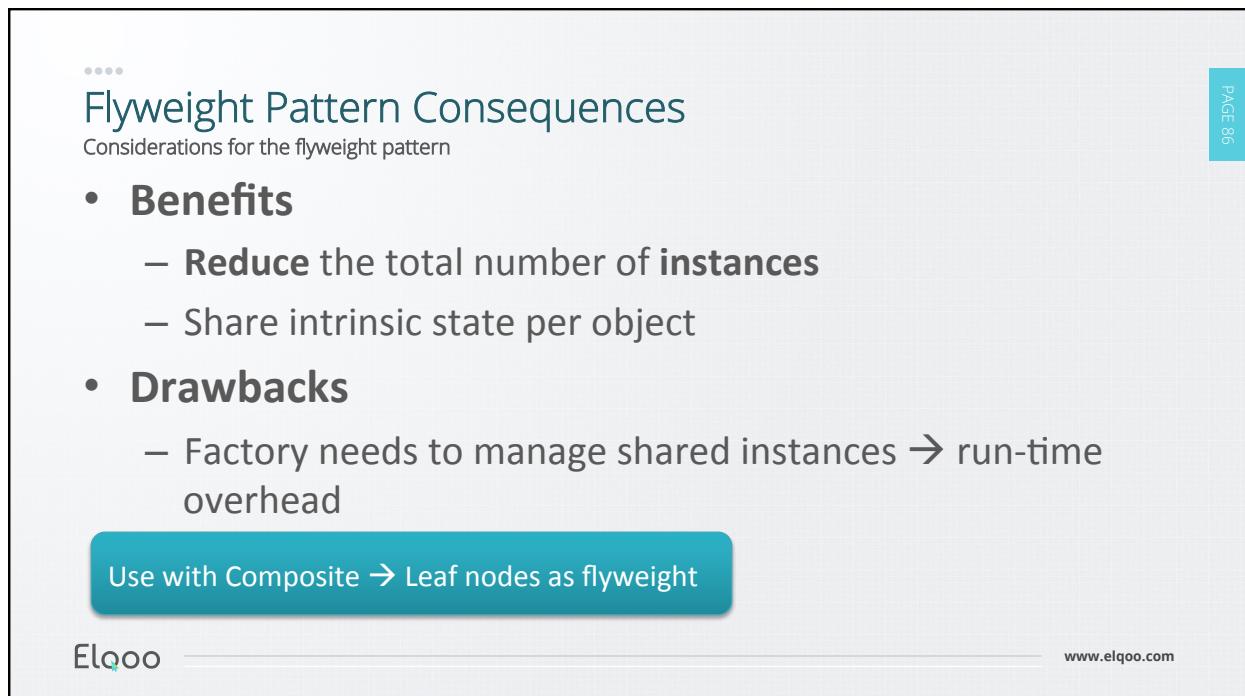
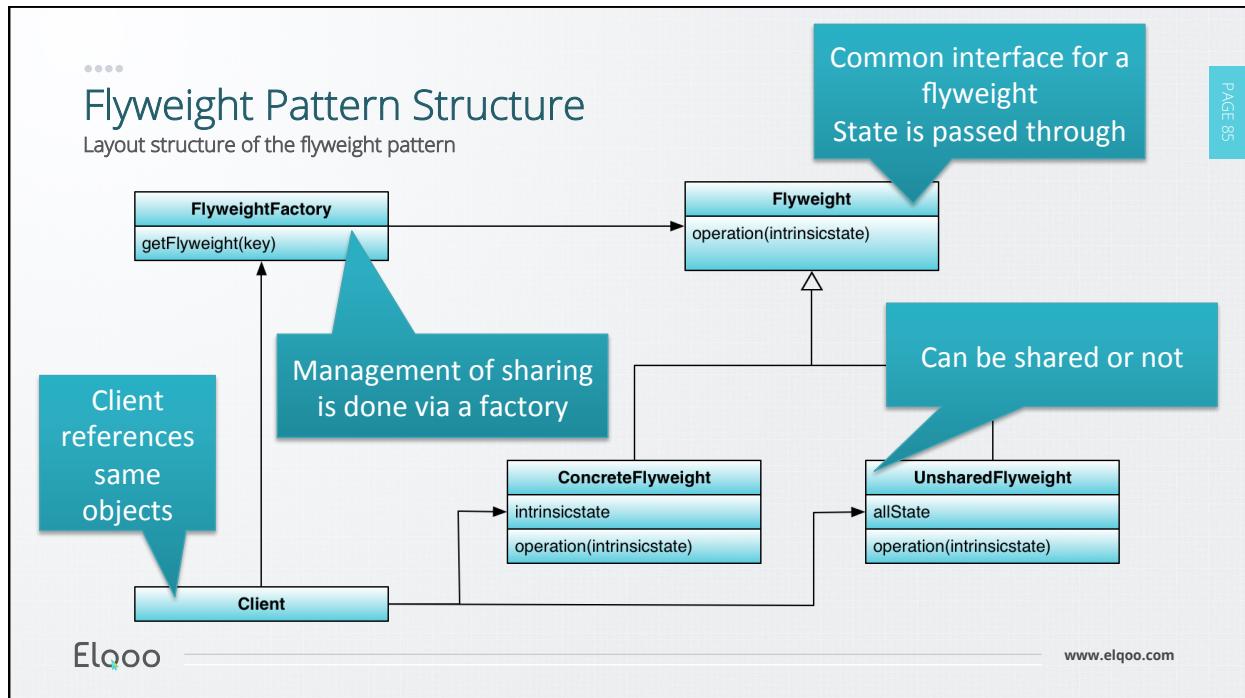
PAGE 84

- **Use**

- Large number of objects
- High storage costs
- Extrinsic state (**shareable**)
- Many objects → replaced by few objects
- Object identity isn't necessary

Elqoo

www.elqoo.com



Conclusion

- **Flyweight pattern is great**
 - Improve performance
 - Manage objects more efficiently