



Content
Creational Design Patterns

1 **Singleton**

www.elqoo.com

PAGE 5



Content
Creational Design Patterns

1 **Singleton**

2 **Builder**

www.elqoo.com

PAGE 6



Content
Creational Design Patterns

PAGE 7

- 1 Singleton**
- 2 Builder**
- 3 Abstract Factory**

www.elqoo.com



Content
Creational Design Patterns

PAGE 8

- 1 Singleton**
- 2 Builder**
- 3 Abstract Factory**
- 4 Factory Method**

www.elqoo.com



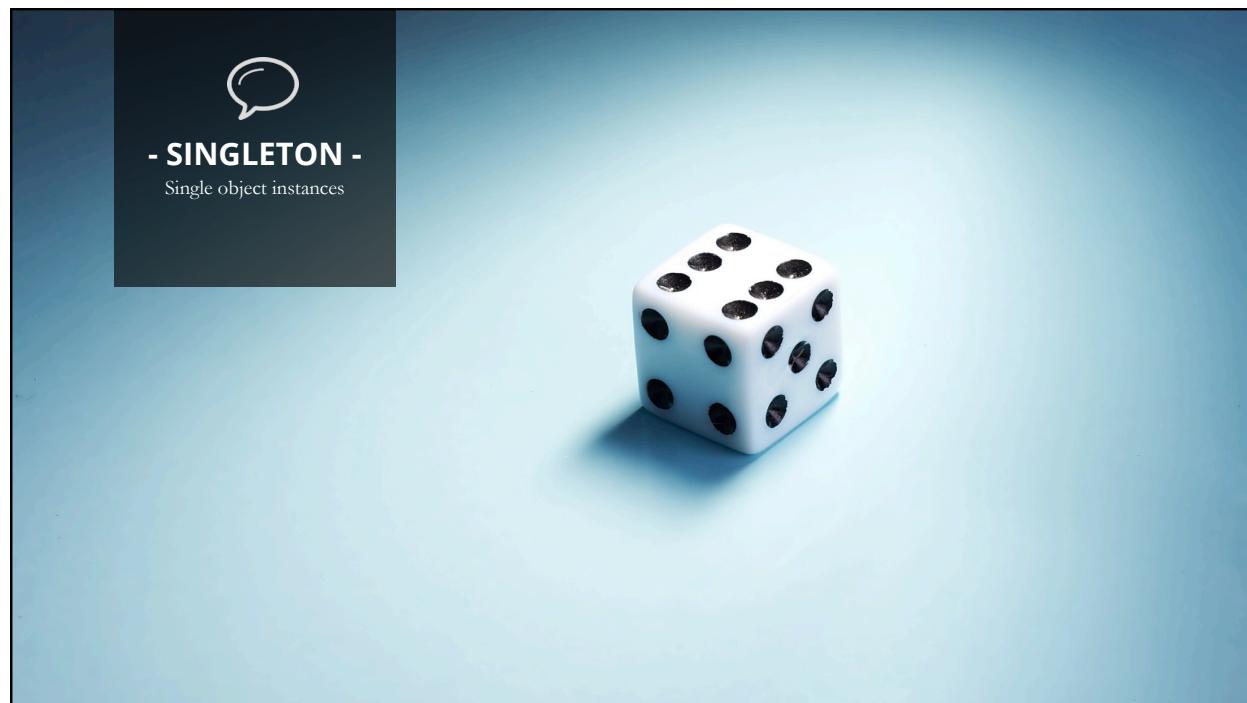
Content

Creational Design Patterns

PAGE 9

- 1 Singleton
- 2 Builder
- 3 Abstract Factory
- 4 Factory Method
- 5 Prototype

www.elqoo.com



.....

Problem Statement

PAGE 11


SD
Brad

- Hello Brad
- Hi Suzy
- We would like to manage preferences in different parts of our application
- I'll get on it


PM
Suzy

Elqoo

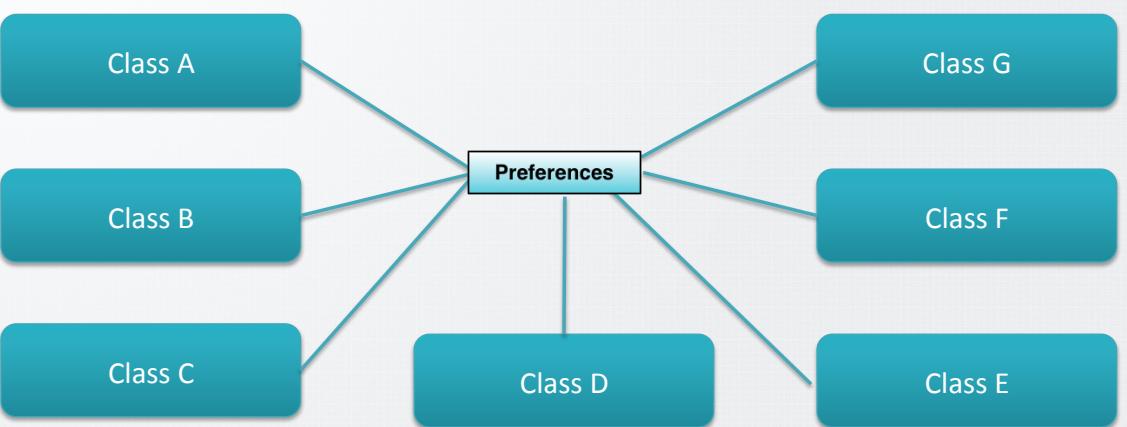
www.elqoo.com

.....

Problem Statement Overview

Preferences need to be accessible from all over the application

PAGE 12



```

graph TD
    Preferences[Preferences] --- ClassA[Class A]
    Preferences --- ClassB[Class B]
    Preferences --- ClassC[Class C]
    Preferences --- ClassD[Class D]
    Preferences --- ClassE[Class E]
    Preferences --- ClassF[Class F]
    Preferences --- ClassG[Class G]
  
```

Elqoo

www.elqoo.com

..... Problem Statement Detail

Detailed explanation of the problem statement

- **Issues**

- Multiple classes require the same object instance
- There can **only be one** object for the entire application
- It must be guaranteed that there is only one object

- **Examples**

- One FileSystem, Window Manager

..... Problem Solution

Reason or logic behind the singleton pattern

- **Solution 1**

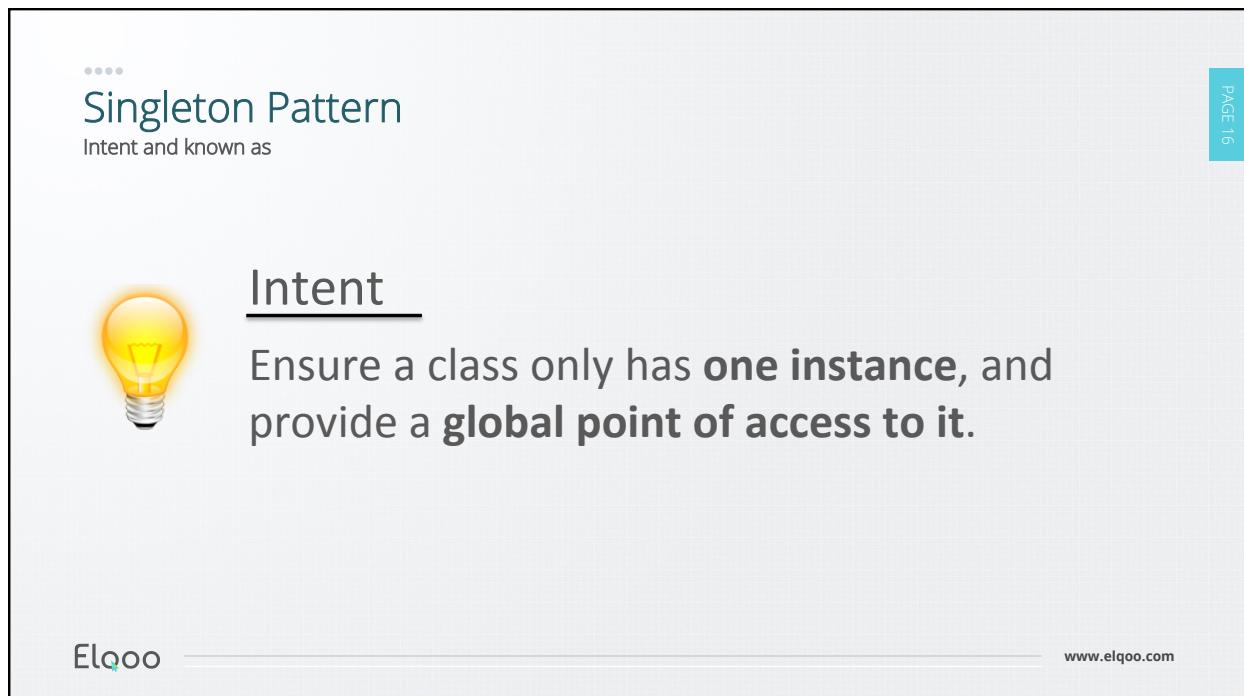
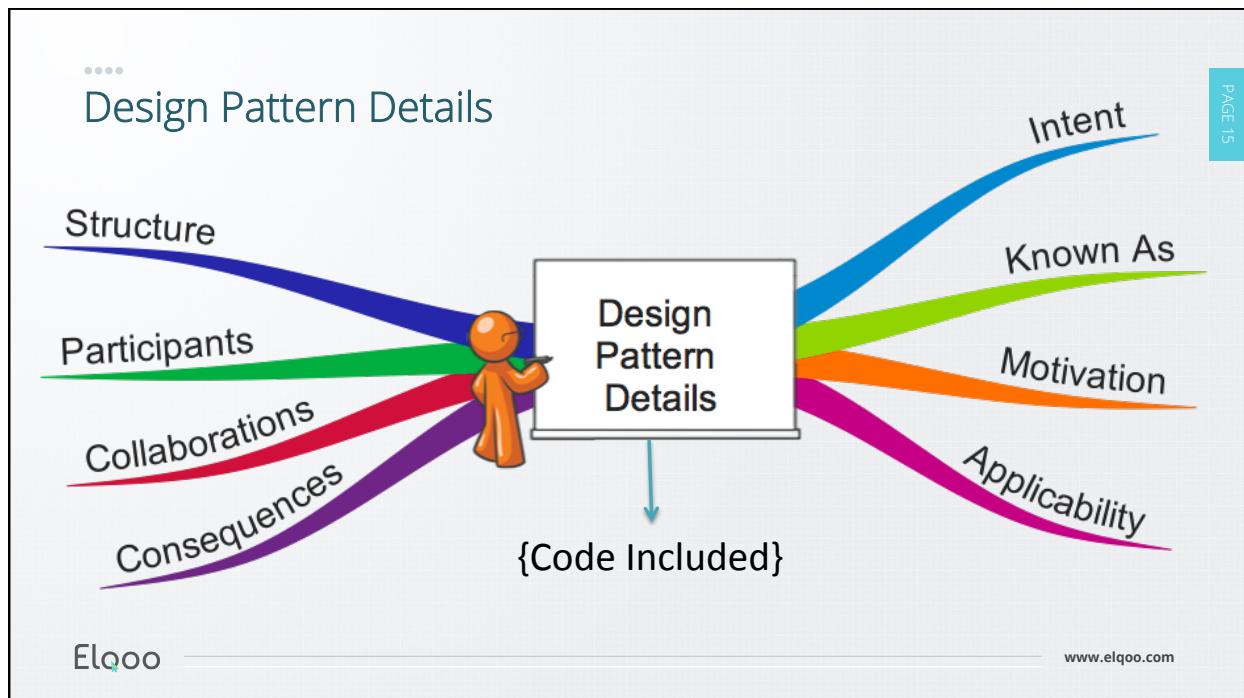
- Create a **global variable**
- Doesn't ensure multiple object creation

- **Solution 2**

- Let class manage its one instance



Singleton Pattern



Apply Singleton Pattern

Applicability

PAGE 17

- **When**

- Only **one instance** of class required
- Must be **one access point**
- Need to **manage object instances**

One instance



One access point

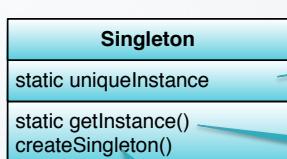
Elqoo

www.elqoo.com

Singleton Pattern Structure

Layout Structure of the singleton pattern

PAGE 18



Store Unique Instance

Only way to retrieve the instance

Create the singleton instance

Elqoo

www.elqoo.com

.... Singleton Pattern Consequences (1)

- **Benefits**

- **Controlled access** to one instance
- Reduce name space → Avoids global variables
- The ability to subclass the singleton class



.... Singleton Pattern Consequences (2)

- **Benefits**

- Configure the number of instances you need

- **Drawbacks**

- State of the singleton must be **shareable** between program executions

Conclusion

PAGE 21

- **Singleton pattern is great**
 - Manage **number of instances** at runtime
 - Provide **single unique access**

Elqoo

www.elqoo.com

- Builder -
Building complex objects



.....

Problem Statement

PAGE 23

Brad (Software Developer) says:

- Hello Brad
- Hi Suzy
- We need a flexible search screen for our customer application.
- I'll get on it.

Suzy (Project Manager) says:

Elqoo — www.elqoo.com

.....

Problem Statement Overview

Designing a search screen that connects to multiple data sources

PAGE 24

Multiple Search Options

Data is spread

Three different databases

Search Screen

Option X Option X
Option X Option X
Option X Option X

SQL

mongoDB

NO SQL

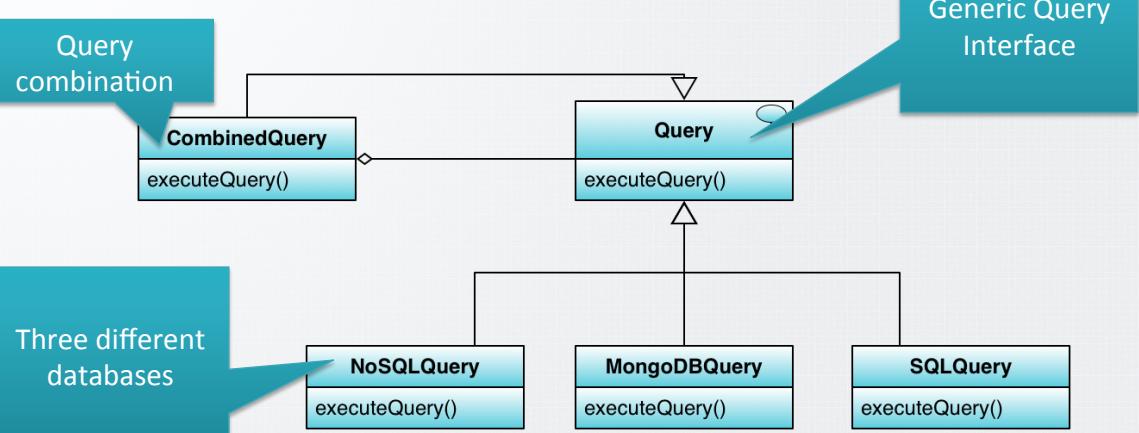
Elqoo — www.elqoo.com

....
How would you implement the query?



Problem Statement Detail (1)

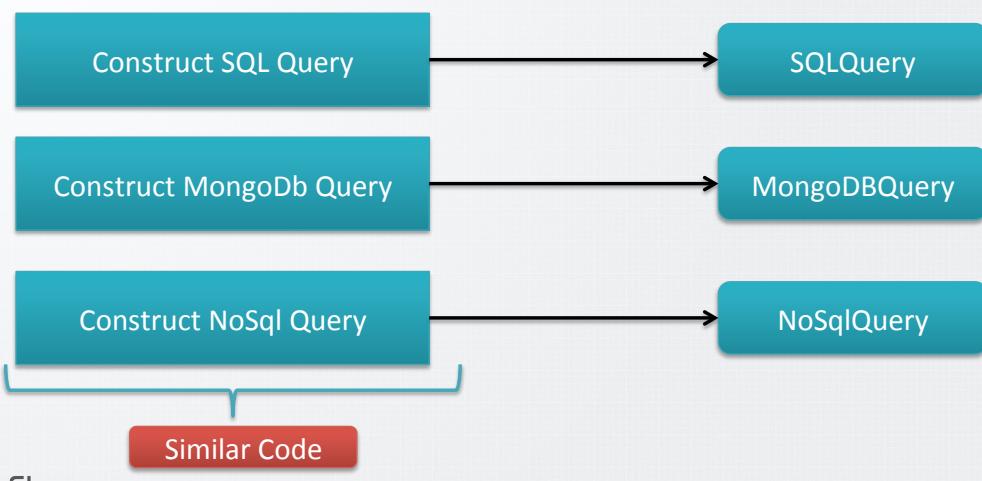
UML representation of the search screen



Problem Statement Detail (2)

Copy of the construction logic

PAGE 27



Problem Statement Detail (3)

Detailed explanation of the screen design problems

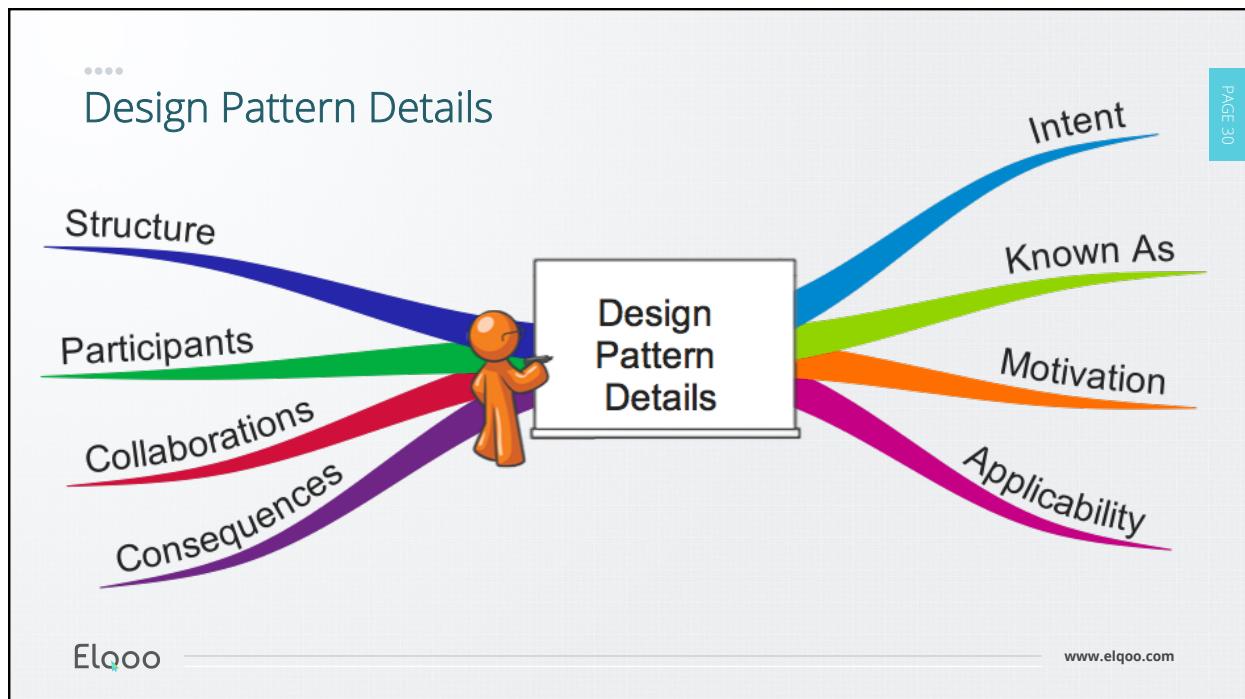
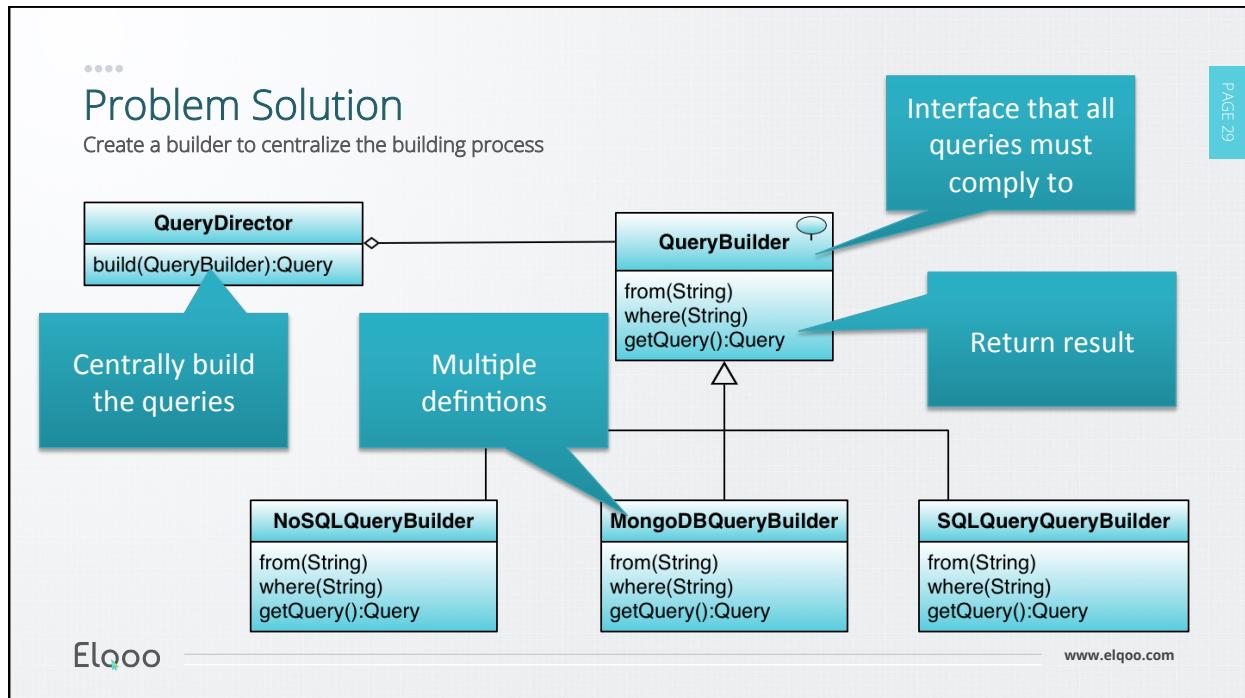
PAGE 28

- **Issues**

- Search screen results in three objects
 - One for each database
- Need to **construct** the three objects **over and over again**

- **Future proof**

- Ideally we would like to **support a fourth database** without changing too much code



Builder Pattern

Intent and known as

PAGE 31



Intent

Separate the construction of a complex object from its representation so that the same construction process can create different representations.

Elqoo

www.elqoo.com

Apply Builder Pattern

Applicability

PAGE 32

- **Use**

- Separate construction with internal representation
- One process → multiple object representation
- Object construction <> object assembling

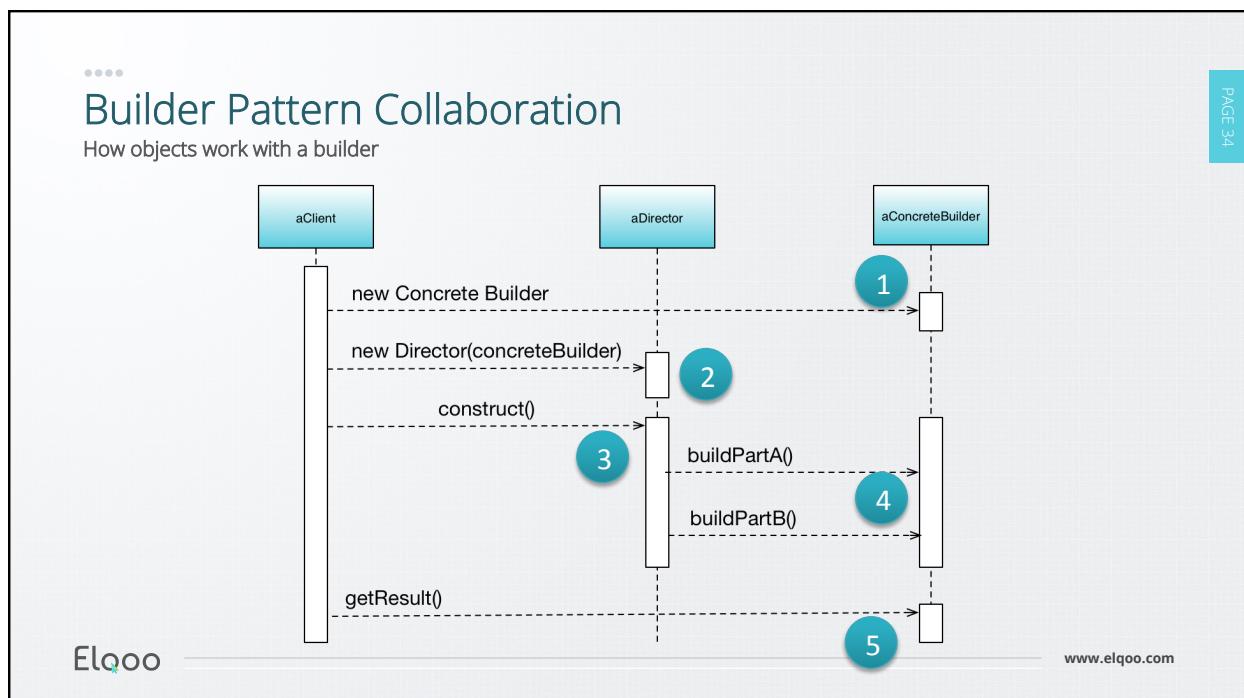
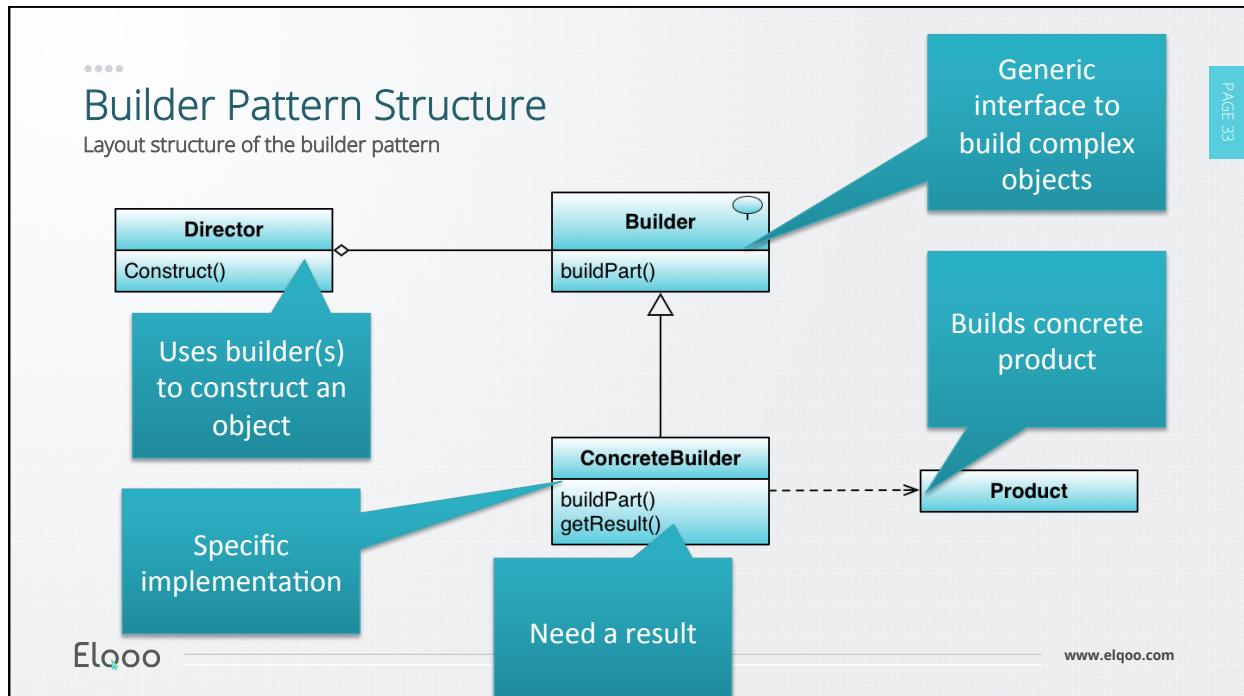
New Object

New Object

Set part A, set part B

Elqoo

www.elqoo.com



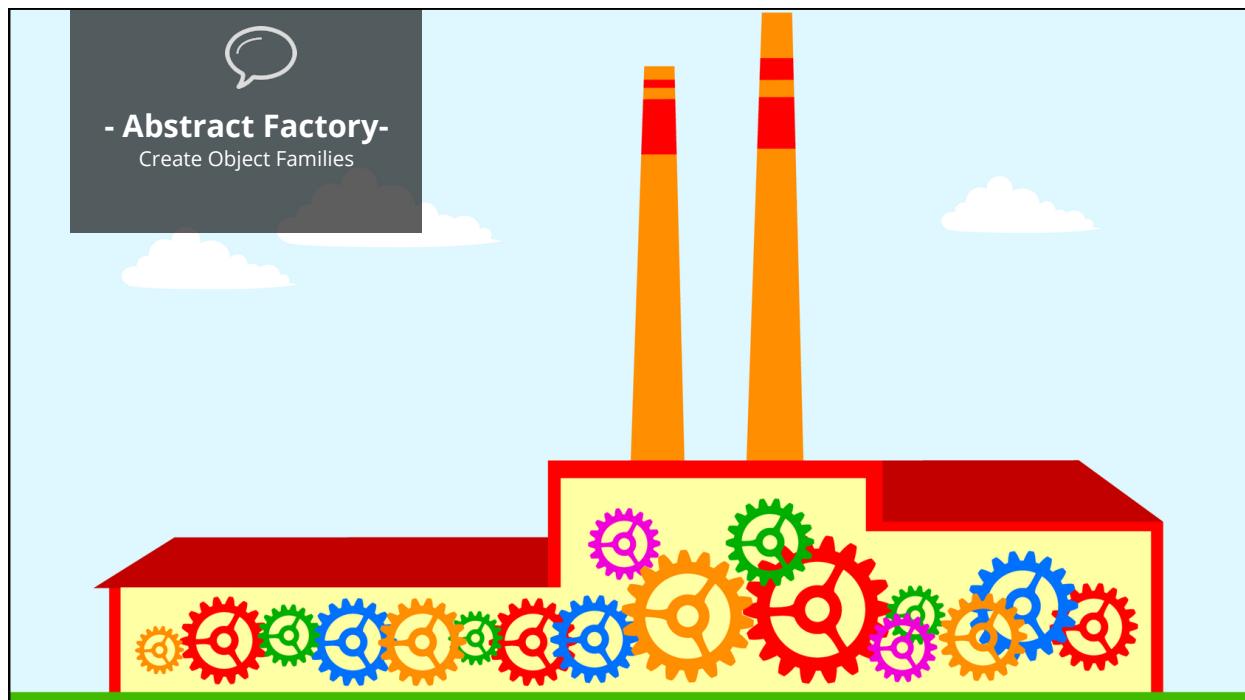
Builder Pattern Consequences

Benefits of the builder pattern

- **Benefits**
 - Uniform production creation via an interface
 - Abstract building process
 - **Loose coupling**
 - Construction
 - Representation
 - **Finer control** on the build process → Allow multiple steps

Conclusion

- **Builder pattern is great**
 - One build process for multiple similar objects
 - Advance control over the build process



.....

Problem Statement

PAGE 38



SD

Brad

- Hello Brad
- Hi Suzy
- We want to use different themes in our application
- No problem, pink too?



PM

Suzy

Elqoo

www.elqoo.com

Problem Statement Overview

PAGE 39



- **Requirement**
 - Easy change theme
 - Add new themes
- **Avoid**
 - Hard coded themes

Elqoo

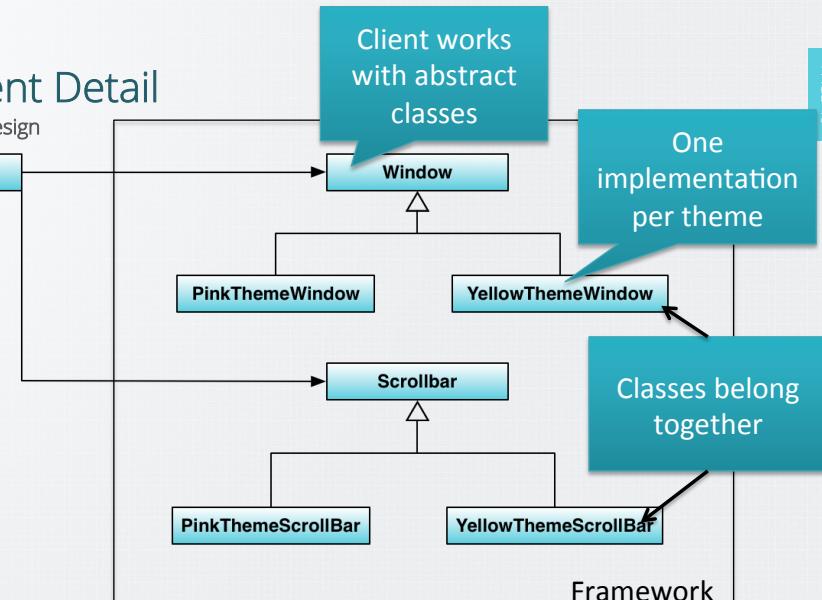
www.elqoo.com

Problem Statement Detail

PAGE 40

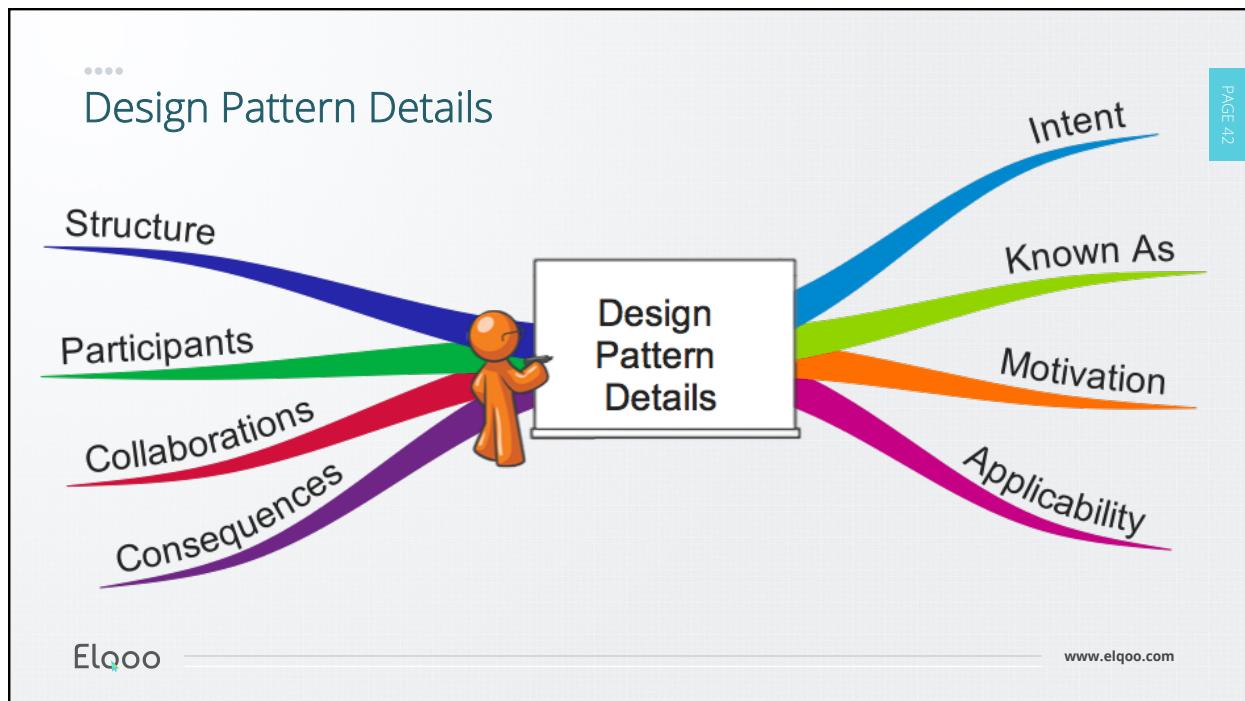
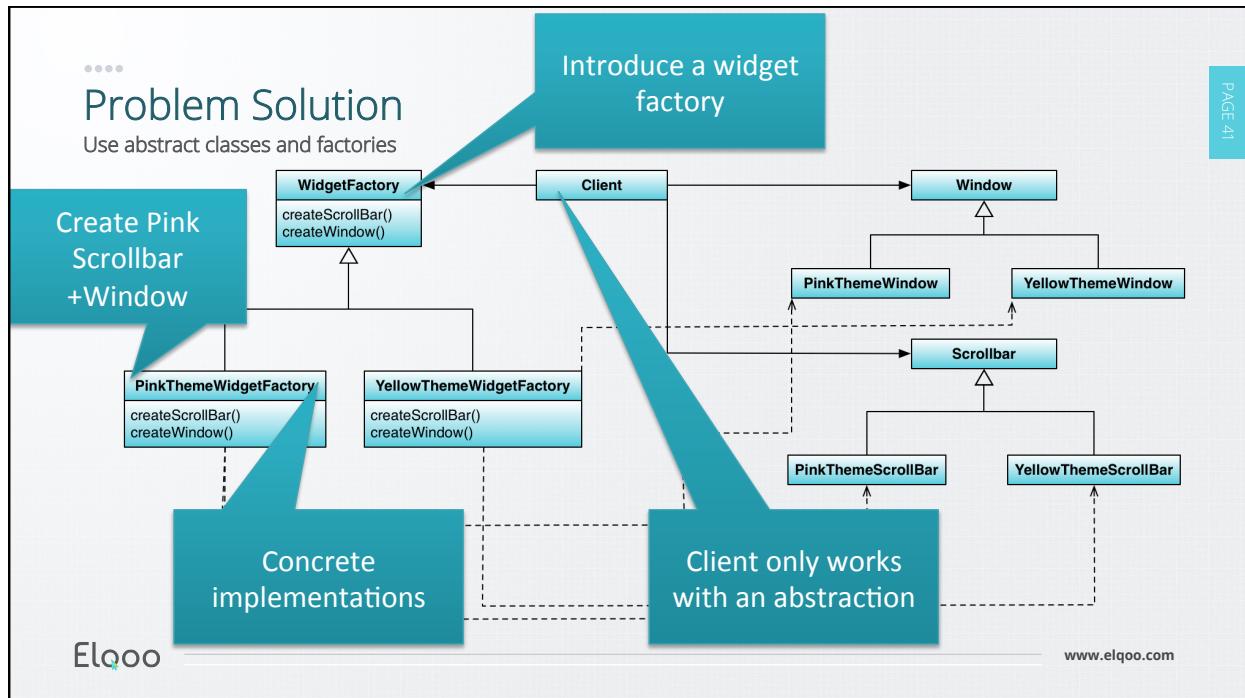
UML representation of a theme design

How will the client
create the right
classes?



Elqoo

www.elqoo.com



Abstract Factory Pattern

Intent and known as



Intent

Provide an **interface** for creating families of related or dependent objects **without specifying their concrete classes.**

Known As

Kit

Elqoo

www.elqoo.com

Apply Abstract Factory Pattern

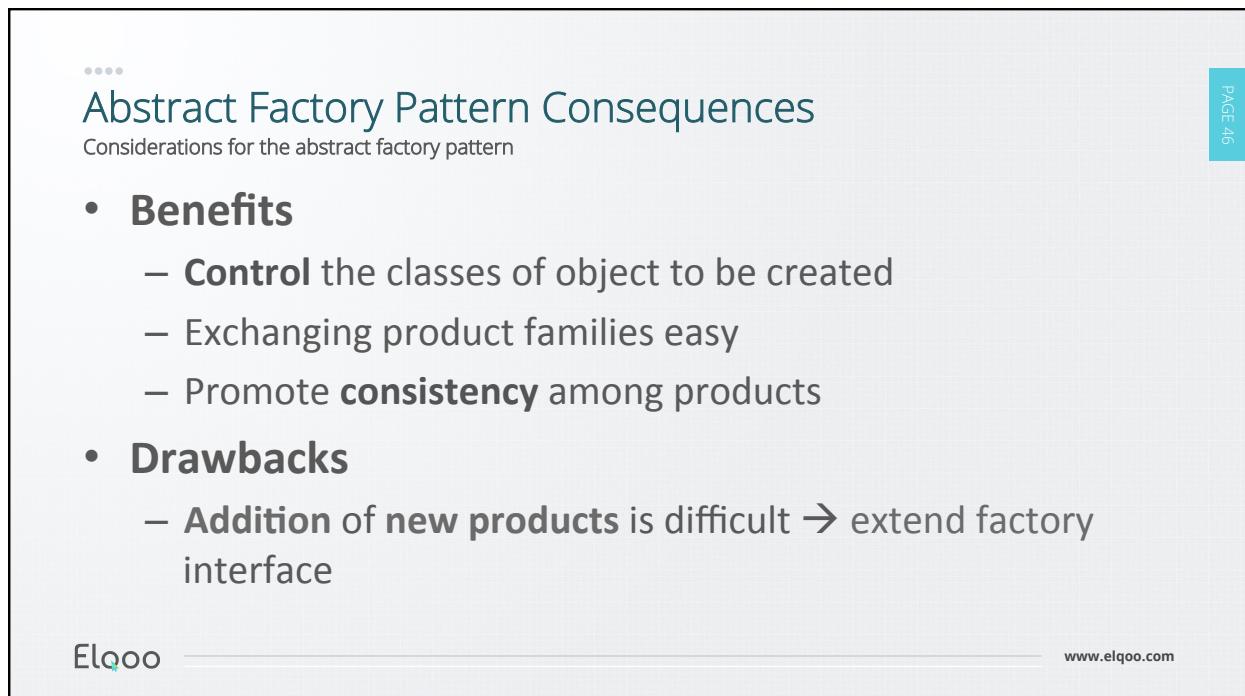
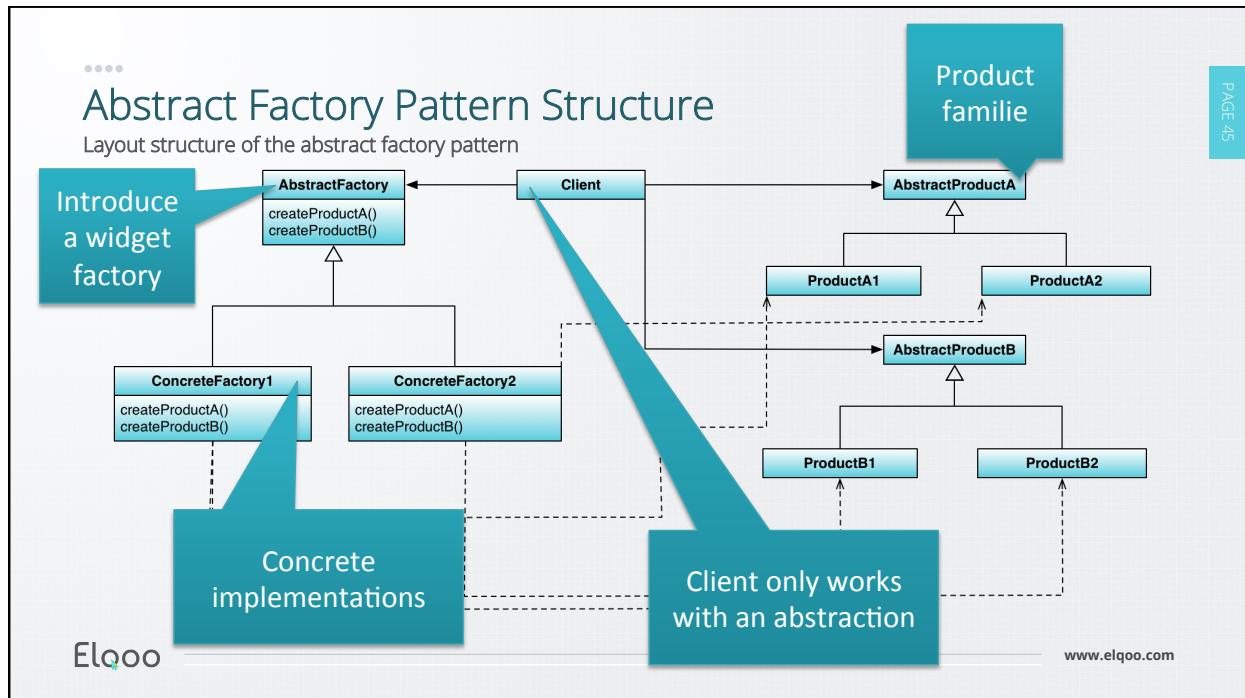
Applicability

- **Use**

- **Creation** of products **independent** from the **application**
- Configuration of product families is required
- Hide product implementation → only provide interface

Elqoo

www.elqoo.com



Conclusion

PAGE 47

- **Abstract Factory pattern is great**
 - Creating product families
 - Centralize creation logic

Elqoo

www.elqoo.com

- Factory Method-

Delegate object creation

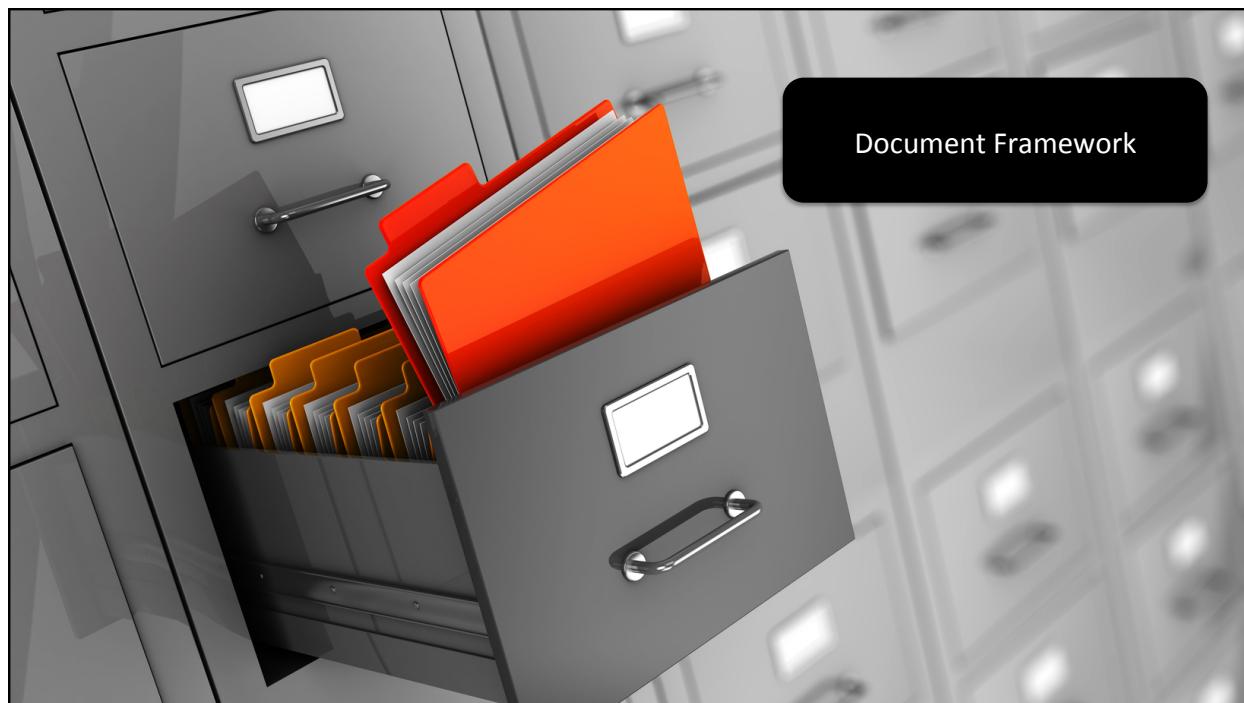


Problem Statement

Brad (SD) is on the left, and Suzy (PM) is on the right. A blue speech bubble contains their conversation:

- Hello Brad
- Hi Suzy
- Can you help our developers only focus on the new functionality to implement for our document framework?
- No problem

Elqoo www.elqoo.com



Problem Statement Overview

Required functionality for the document framework

PAGE 51

- **Framework Requirement**

- **Flexible** create new documents
- **Document type per application**
- Re-use document code

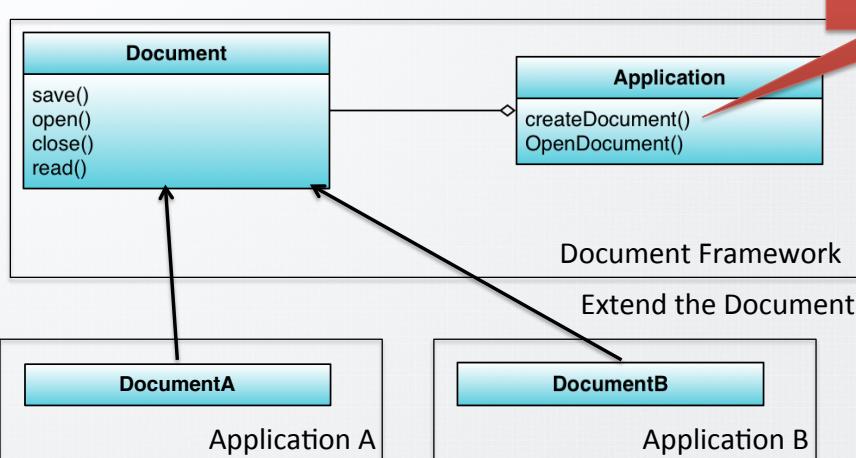
Elqoo

www.elqoo.com

Problem Statement Detail

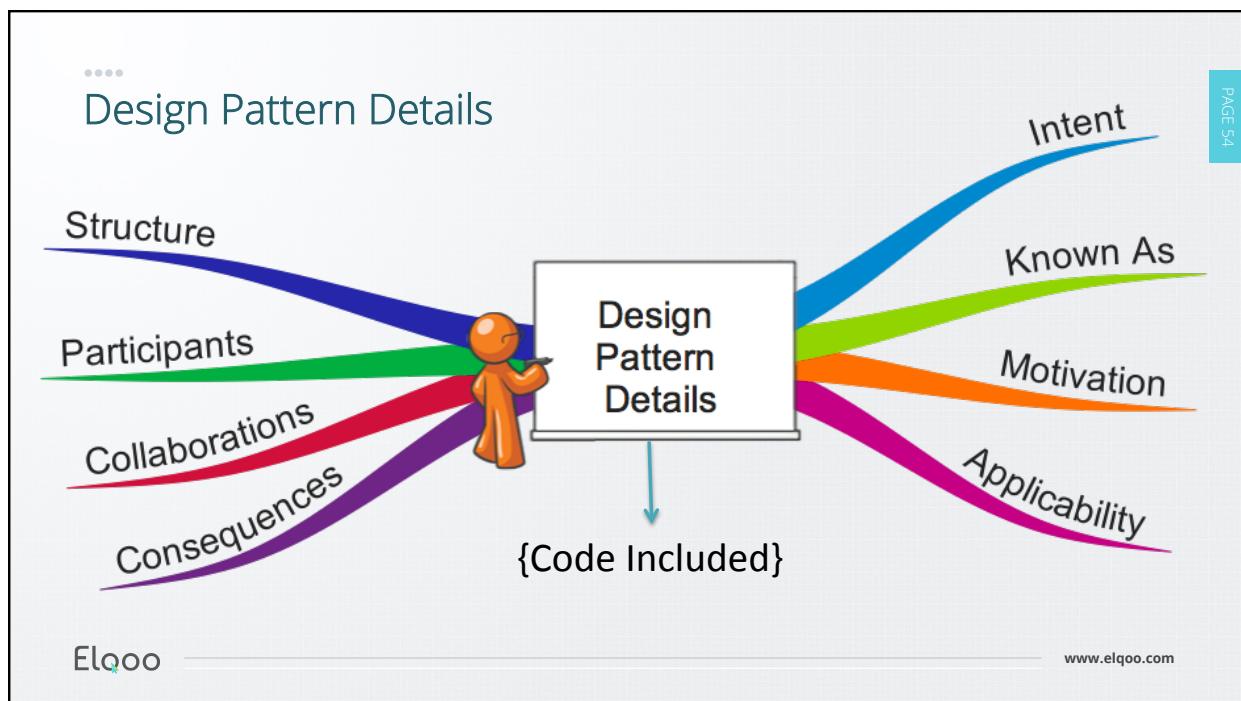
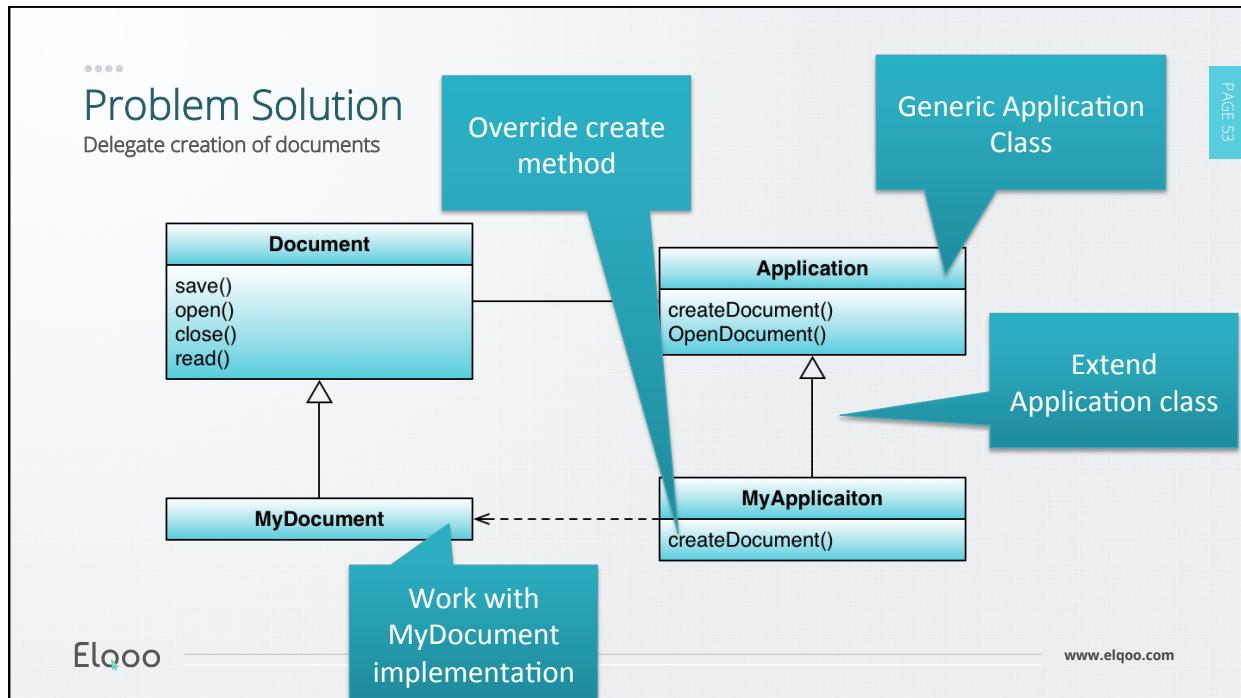
UML representation of the document framework

PAGE 52



Elqoo

www.elqoo.com



Factory Method Pattern

Intent and known as

PAGE 55



Intent

Define an **interface** for creating an object but let subclasses decide which class to instantiate. Factory Method lets a class **defer instantiation to subclasses**.

Known As

Virtual Constructor

Elqoo

www.elqoo.com

Apply Factory Method Pattern

Applicability

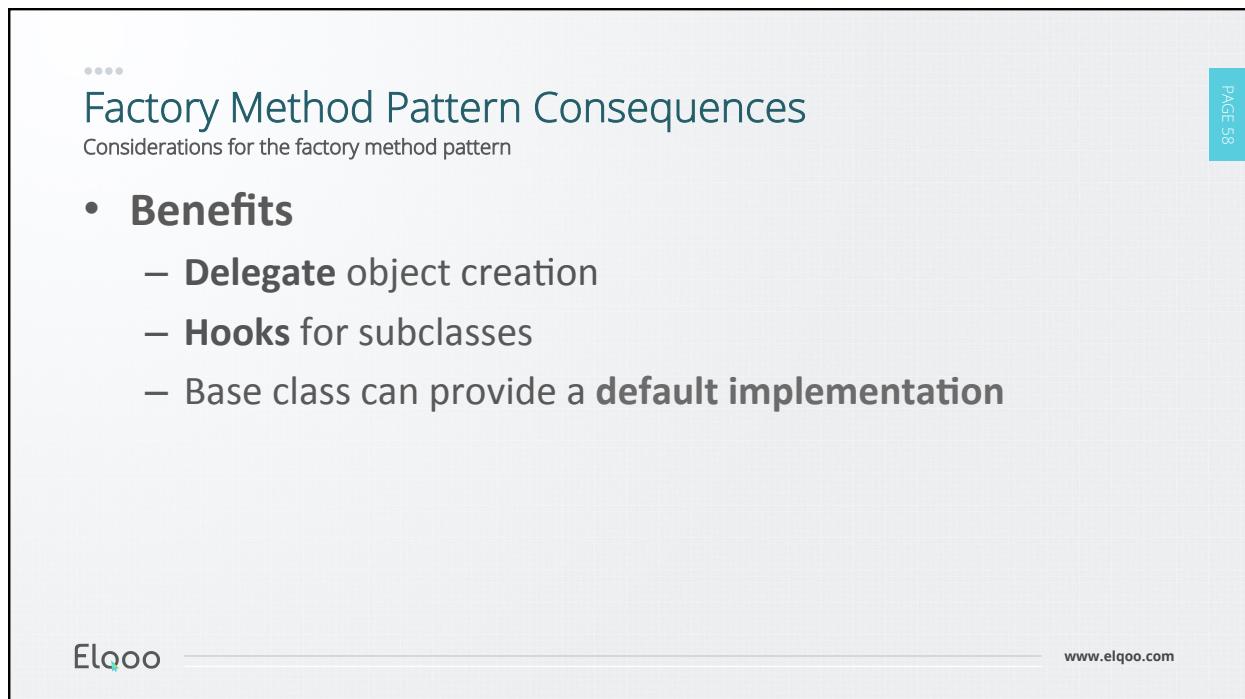
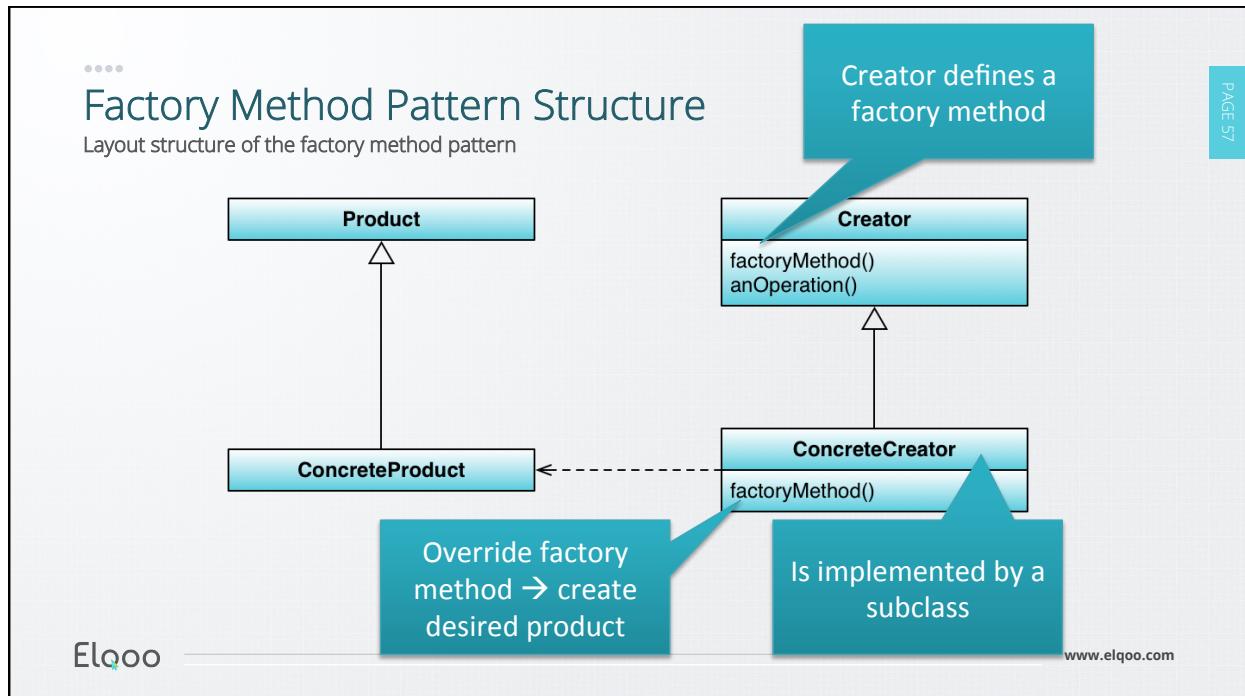
PAGE 56

- **When**

- Class can't **expect** the type of object it must **create**
- **Subclasses** must decide what types of **objects are created**

Elqoo

www.elqoo.com



Conclusion

PAGE 59

- **Factory Method pattern is great**
 - Delegate object creation at runtime
 - Don't know what type of class you need to create

Elqoo

www.elqoo.com

- Prototype-
Create object on
prototype



.....

Problem Statement

PAGE 61

Brad (SD)

Suzy (PM)

Elqoo

www.elqoo.com

Brad's Problem Statement:

- Hello Brad
- Hi Suzy
- Our Graphic framework isn't as usable as we thought.
- Strange, I thought it was well designed

.....

Problem Statement Overview

UML representation of the graphics framework

PAGE 62

Framework

GraphicTool

operation(graphicType)

Graphic

Image

Video

Application Specific

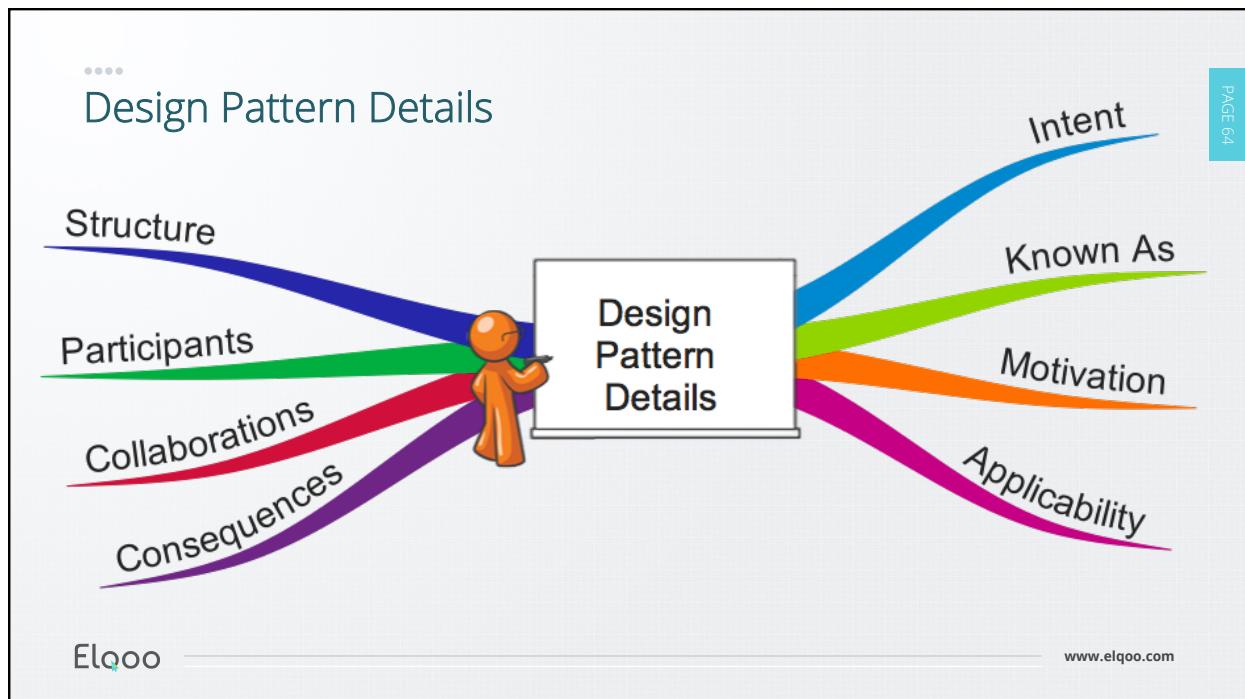
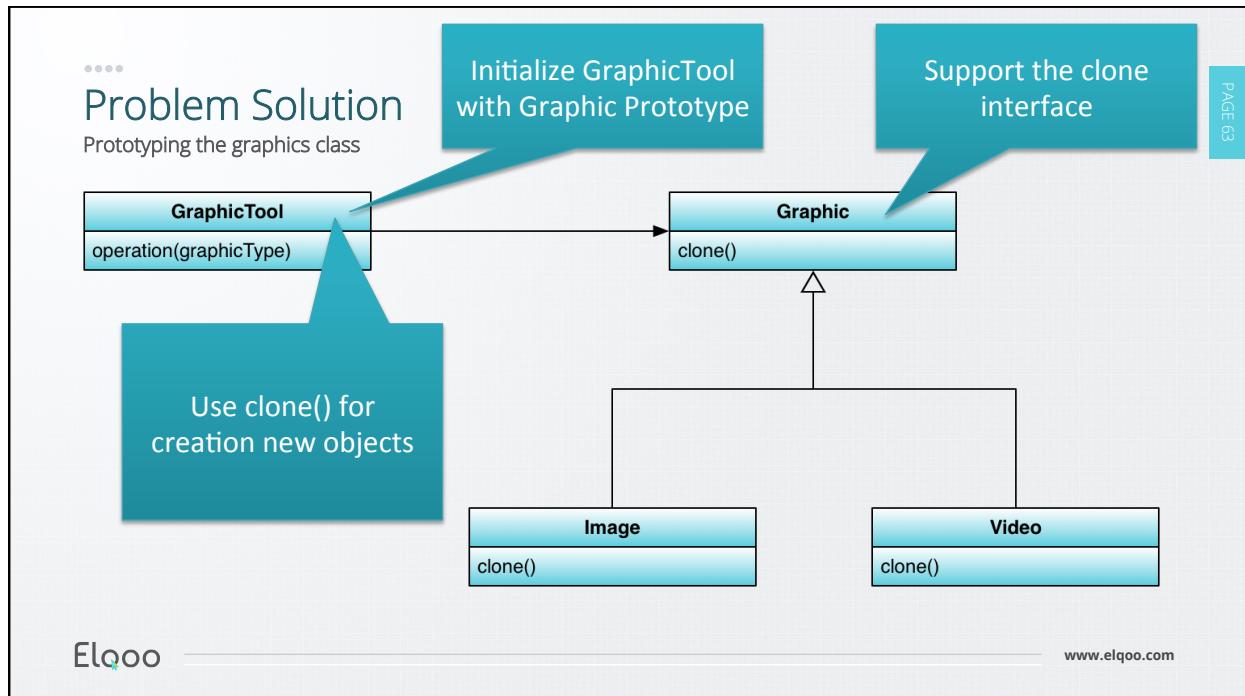
Base Graphic Object

Two Types

Needs to initialize a new Graphic, but cannot say new "GraphicClass"

Elqoo

www.elqoo.com



..... Prototype Pattern

Intent and known as

PAGE 65



Intent

Specify the kinds of objects to create using a **prototypical instance**, and create new objects by **copying this prototype**.

Elqoo

www.elqoo.com

..... Apply Prototype Pattern

Applicability

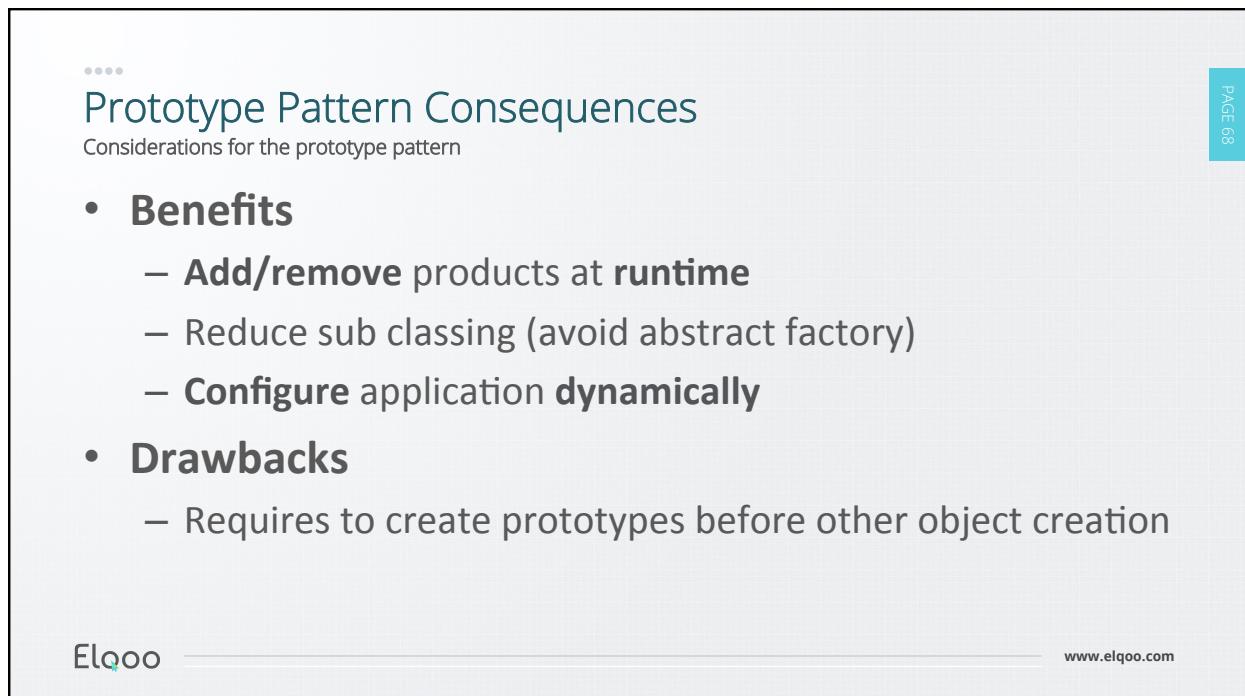
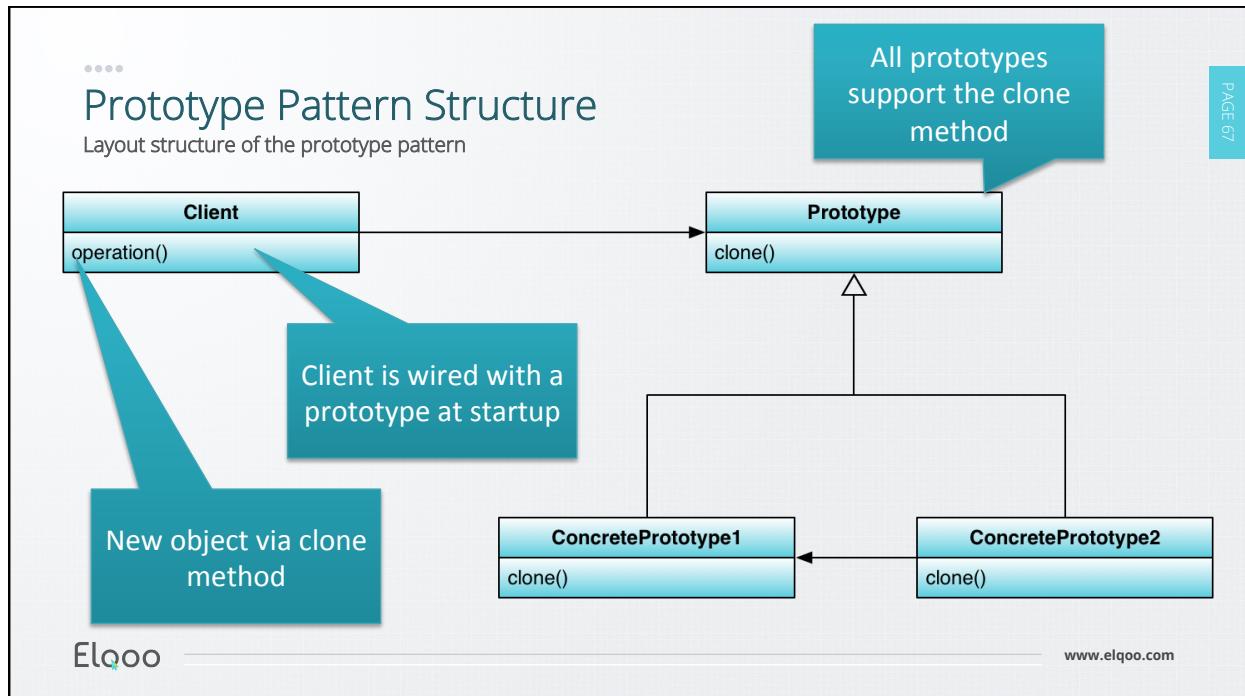
PAGE 66

• When

- Classes to instantiate are specific at **run-time**
- Avoid building class hierarchies (abstract factory pattern)
- A class can have **limited instances** of state
 - Cloning is more efficient

Elqoo

www.elqoo.com



Conclusion

PAGE 69

- **Prototype pattern is great**
 - Configure object creation by cloning
 - Dynamically change object creation