Breathing Life Back into Factory Methods with Lambdas

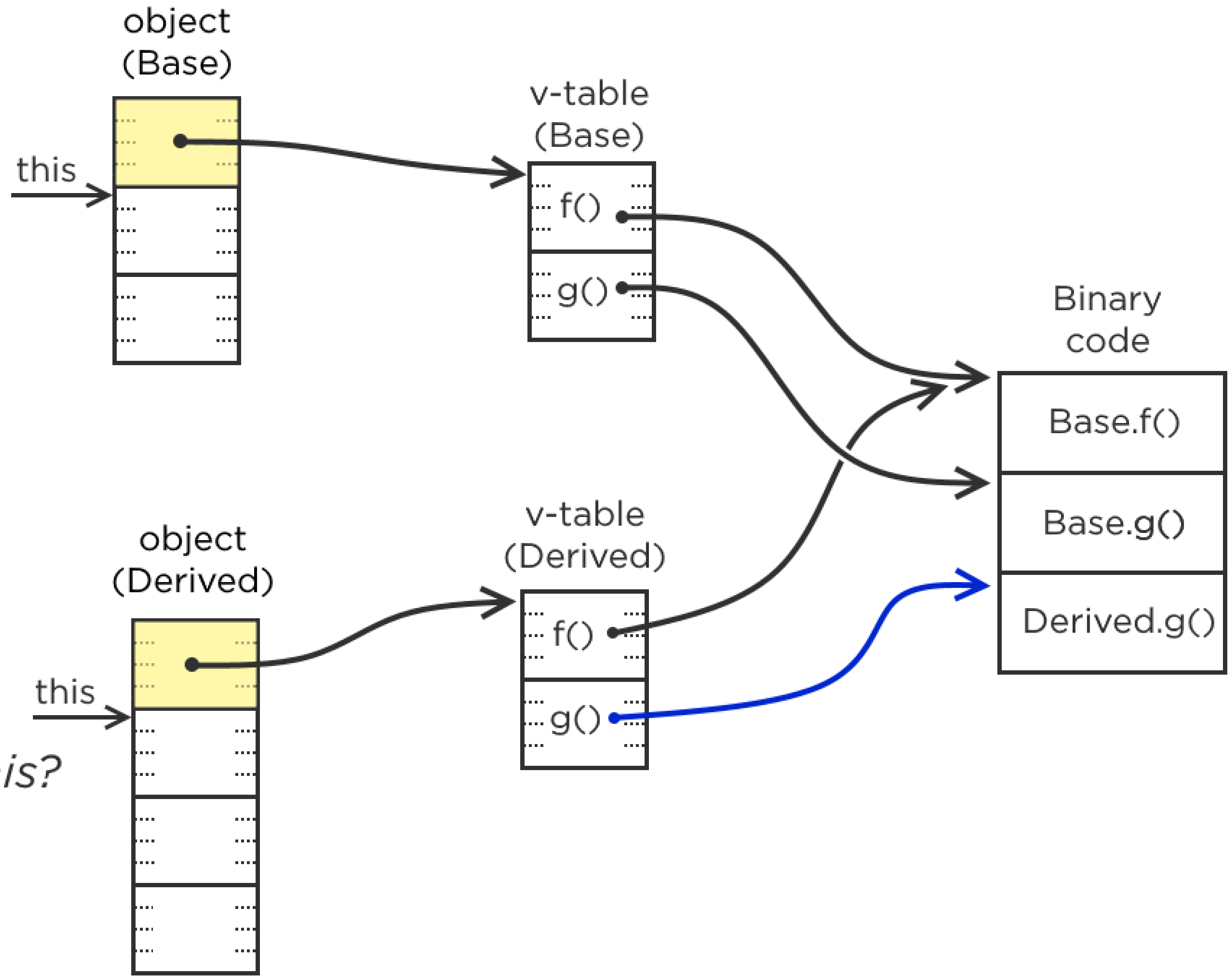
Zoran Horvat
OWNER AT CODING HELMET CONSULTANCY
@zoranh75 www.codinghelmet.com

Factory Method Pattern

In this module:

ıto an injectable delegate

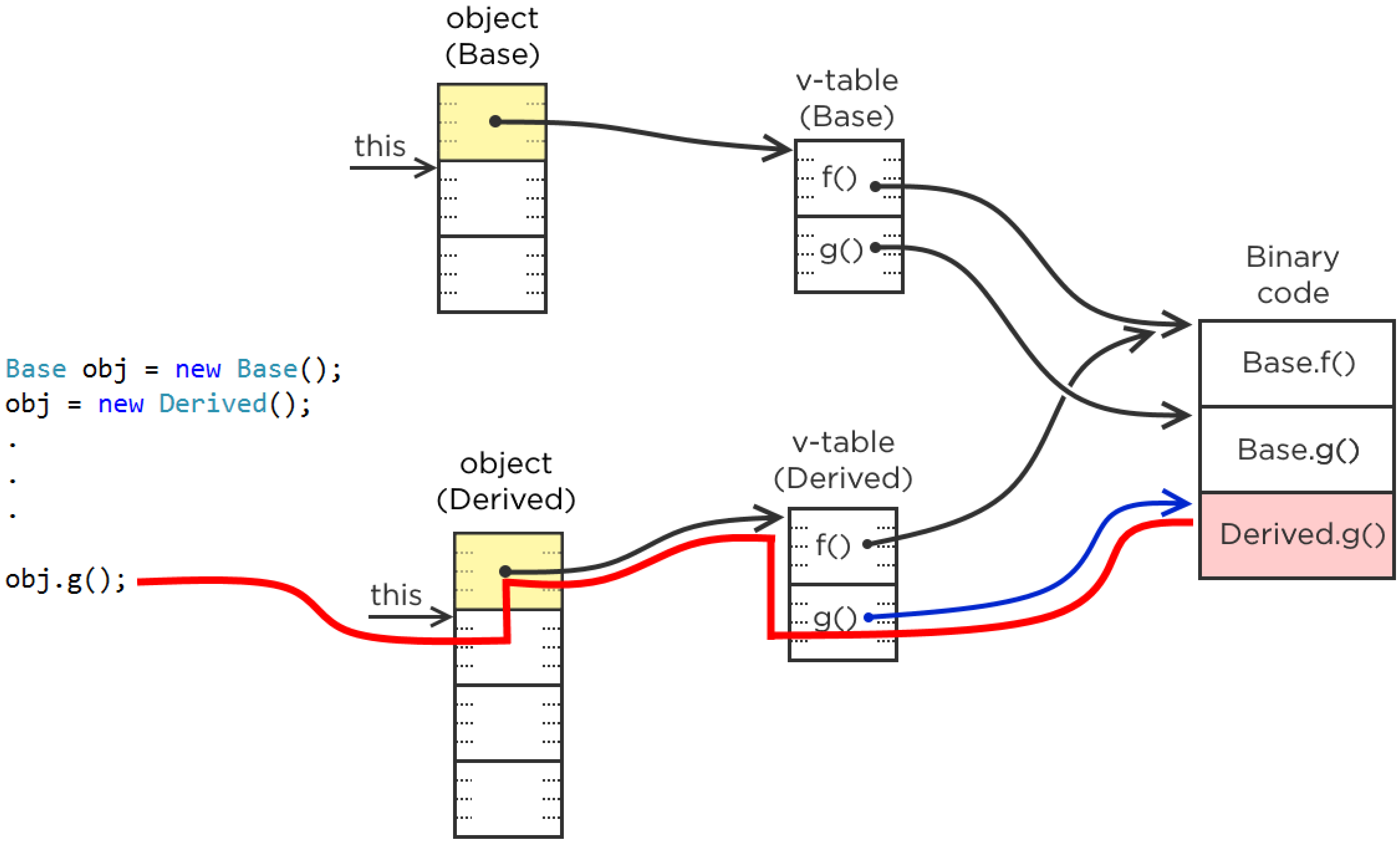
ɱposition (similar to Strategy pattern)



```
Base obj = new Base();
obj = new Derived();
.
.
.
```

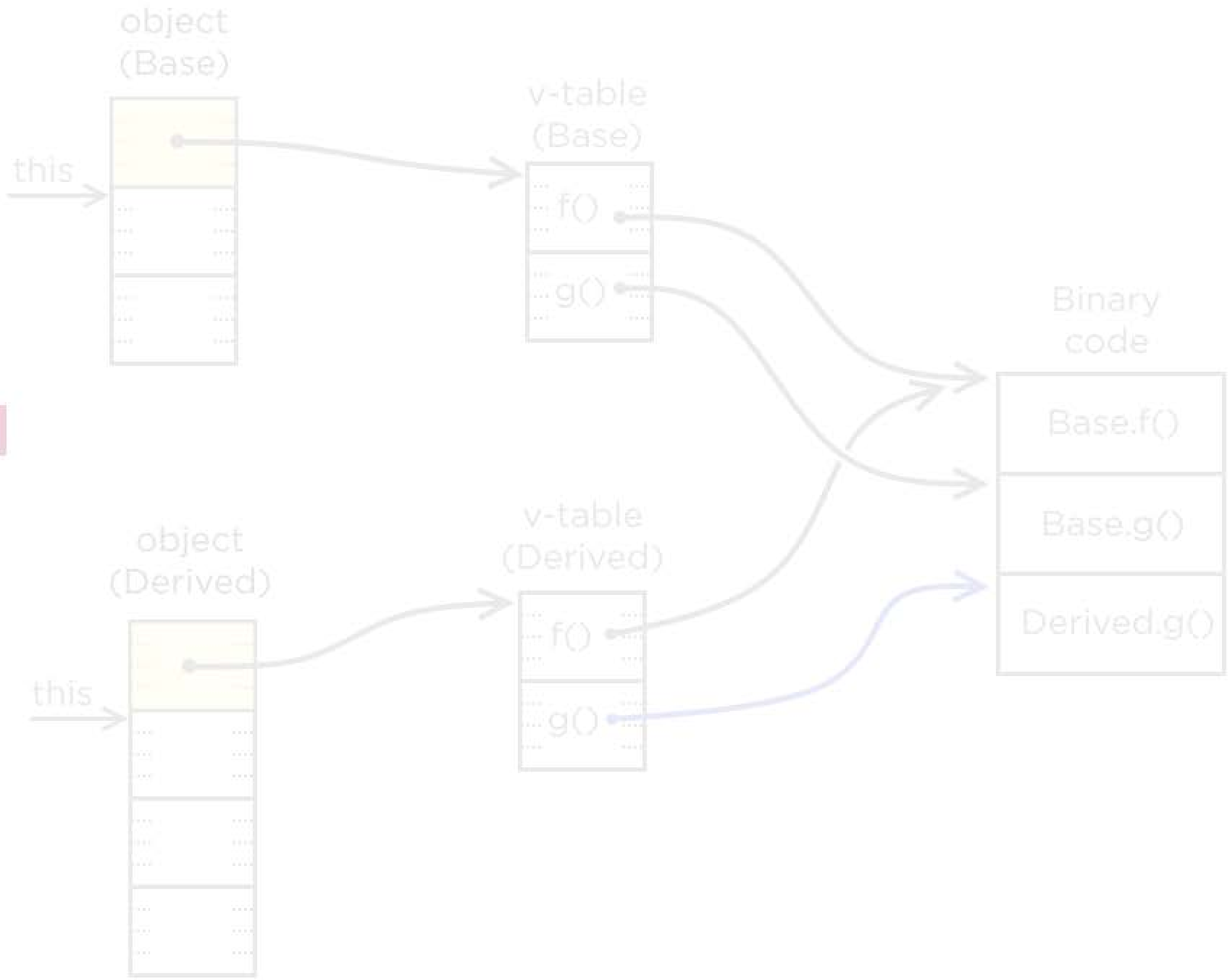
```
obj.g();
```

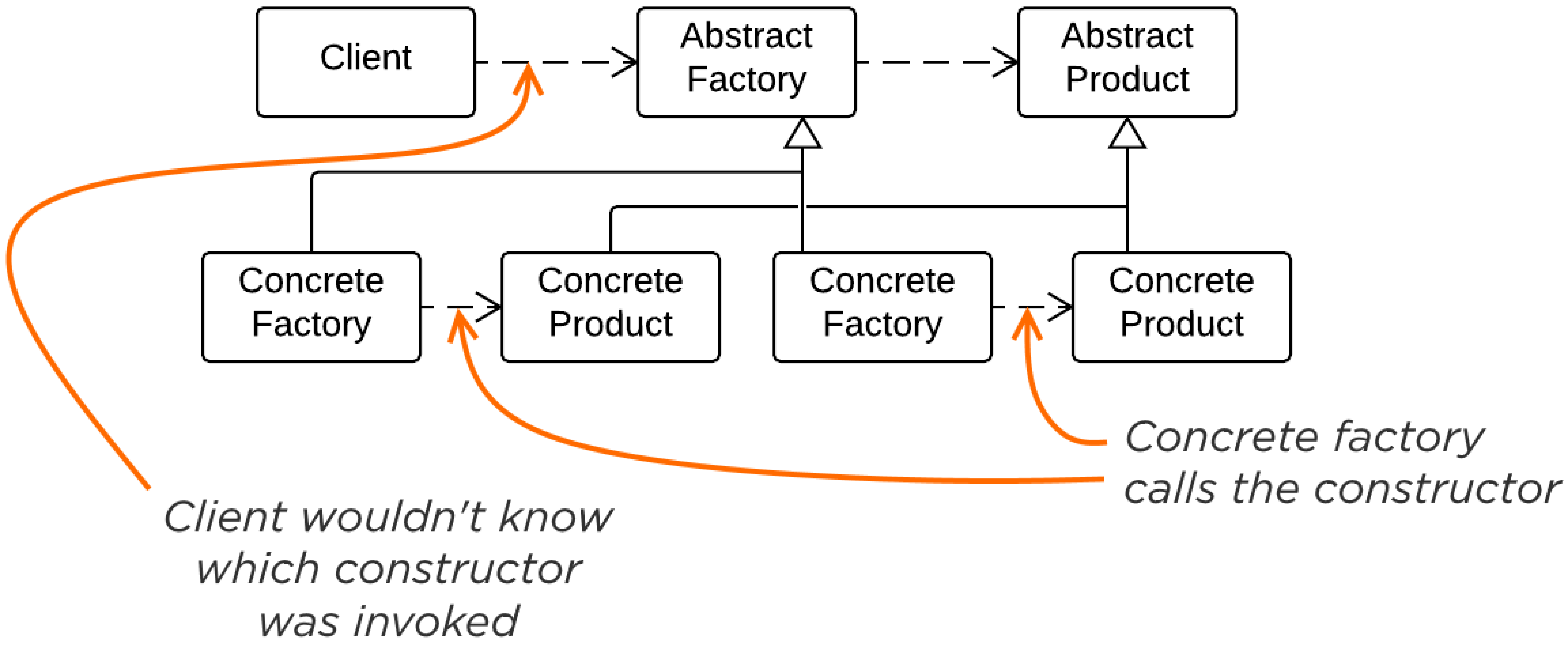
Which object is this?
What is its type?

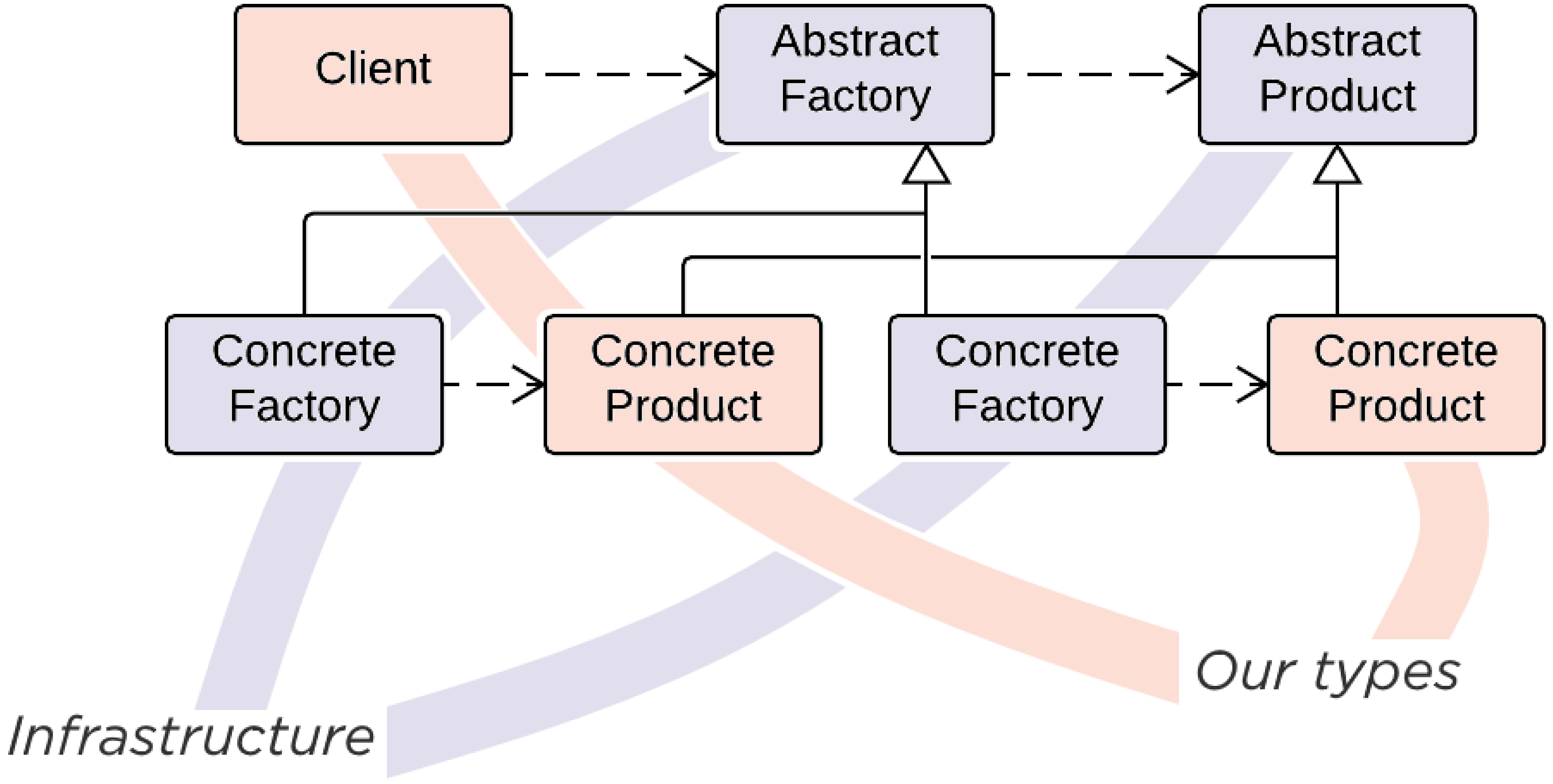


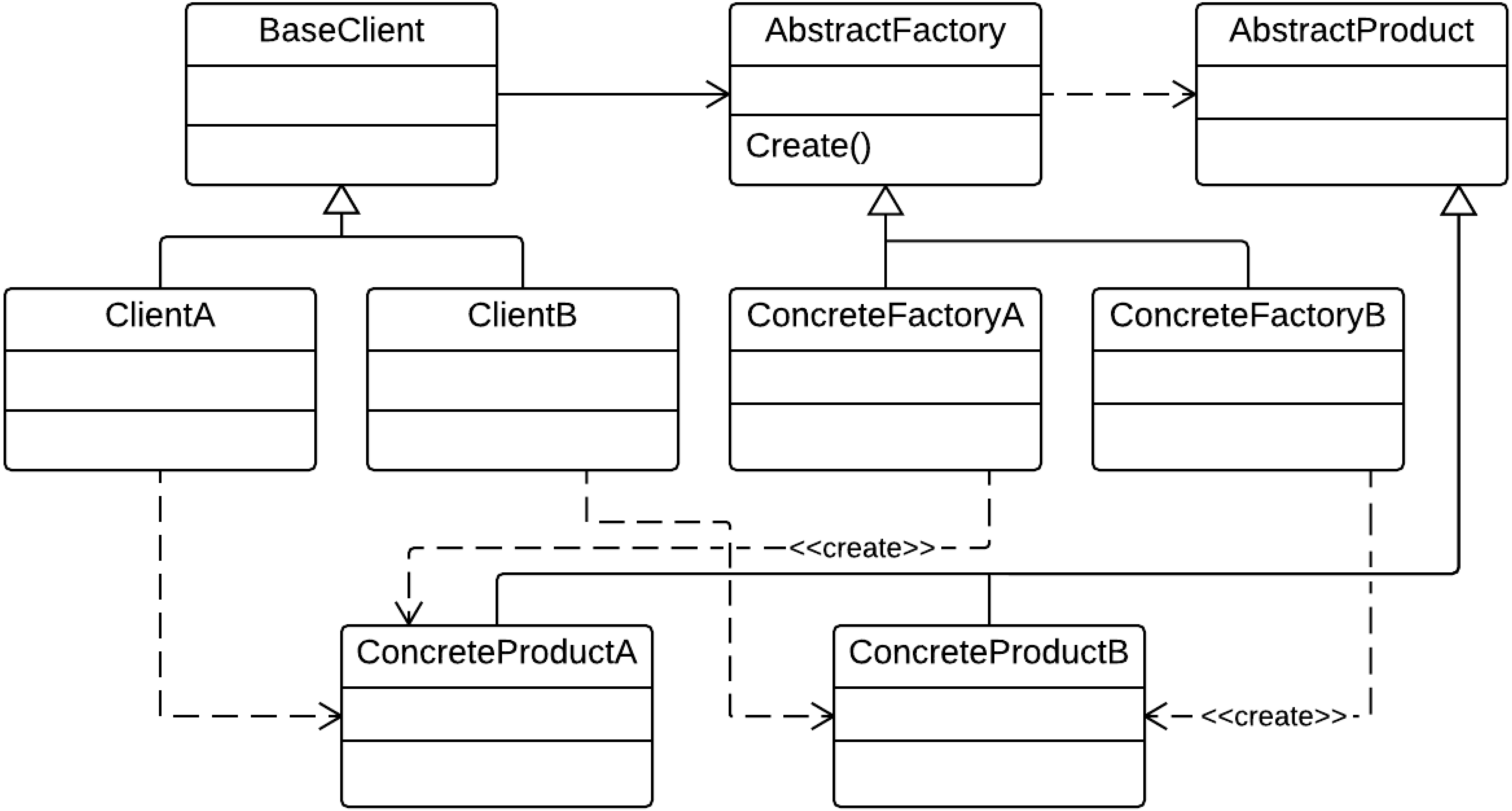
```
Base obj = new Base();
obj = new Derived();
.
.
.
obj.g();
```

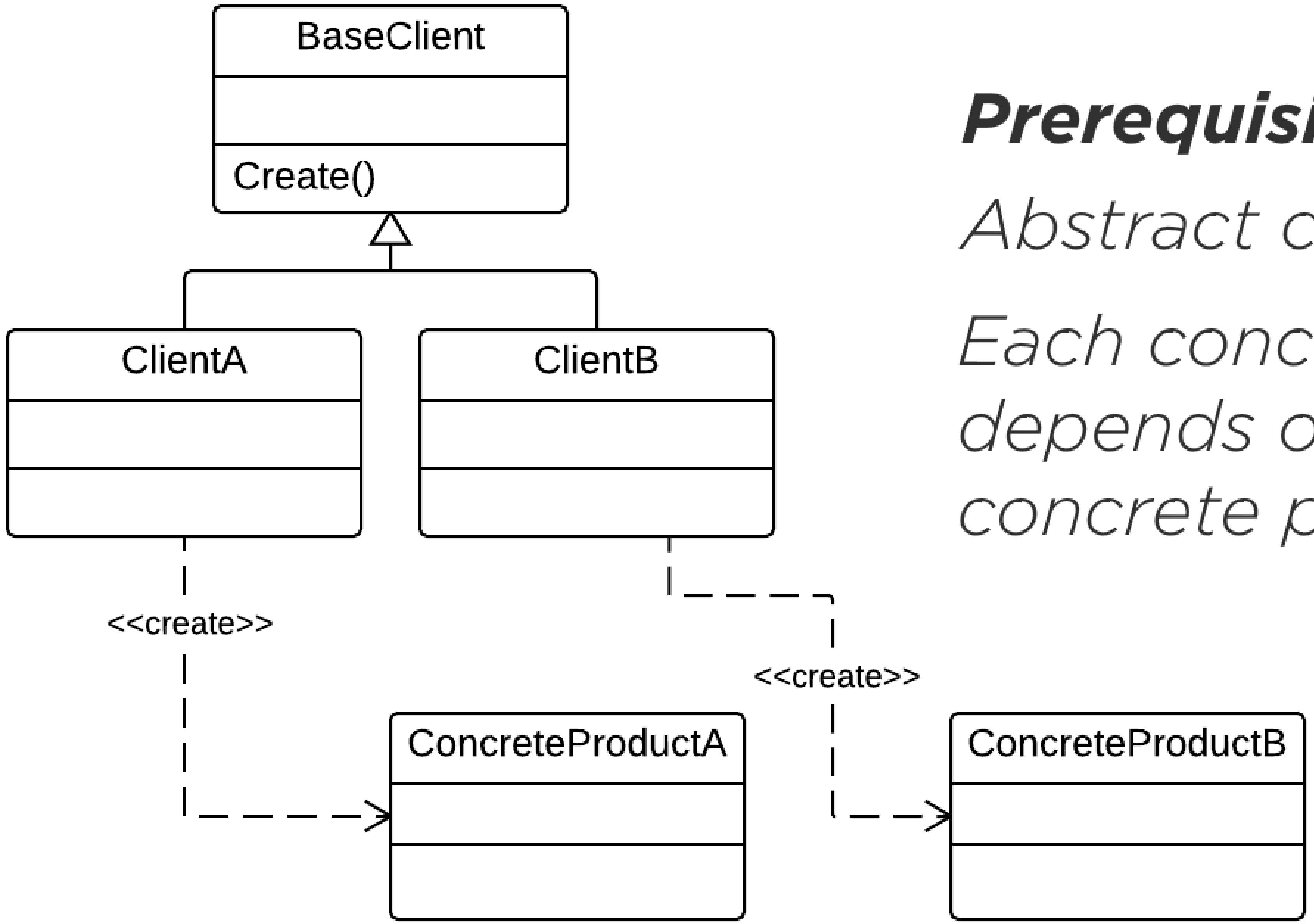
```
Base obj = new Base();
obj = new Derived();
.
.
.

obj.g();
```




Client
Abstract Factory
Abstract Product
Concrete Factory
Concrete Product
Concrete Factory
Concrete Product
Concrete factory calls the constructor
Client wouldn't know which constructor was invoked


Client
Abstract Factory
Abstract Product
Concrete Factory
Concrete Product
Concrete Factory
Concrete Product
Our types
Infrastructure

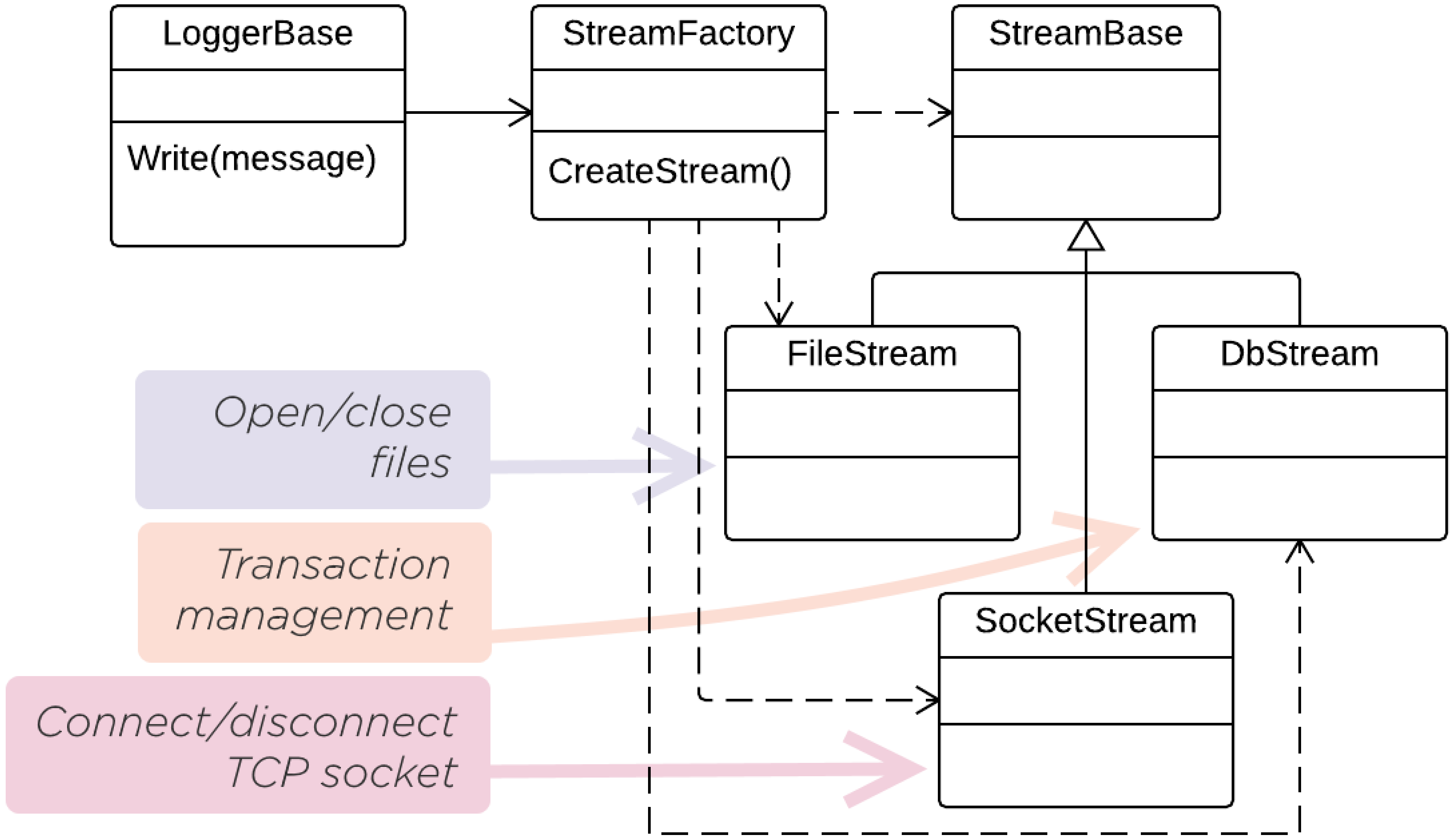

BaseClient
AbstractFactory
Create()
AbstractProduct
ClientA
ClientB
ConcreteFactoryA
ConcreteFactoryB
<<create>>
ConcreteProductA
ConcreteProductB
<<create>>

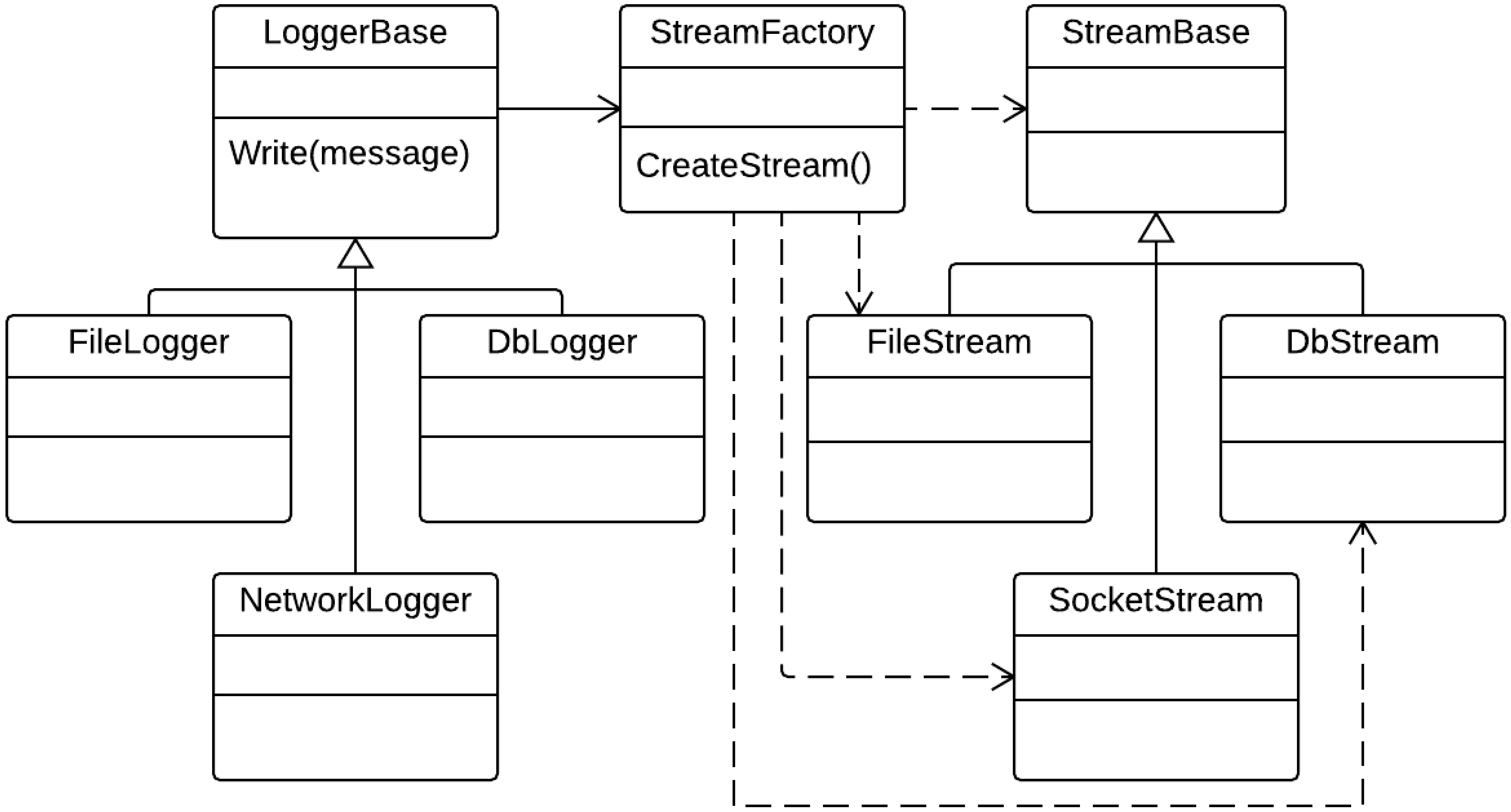


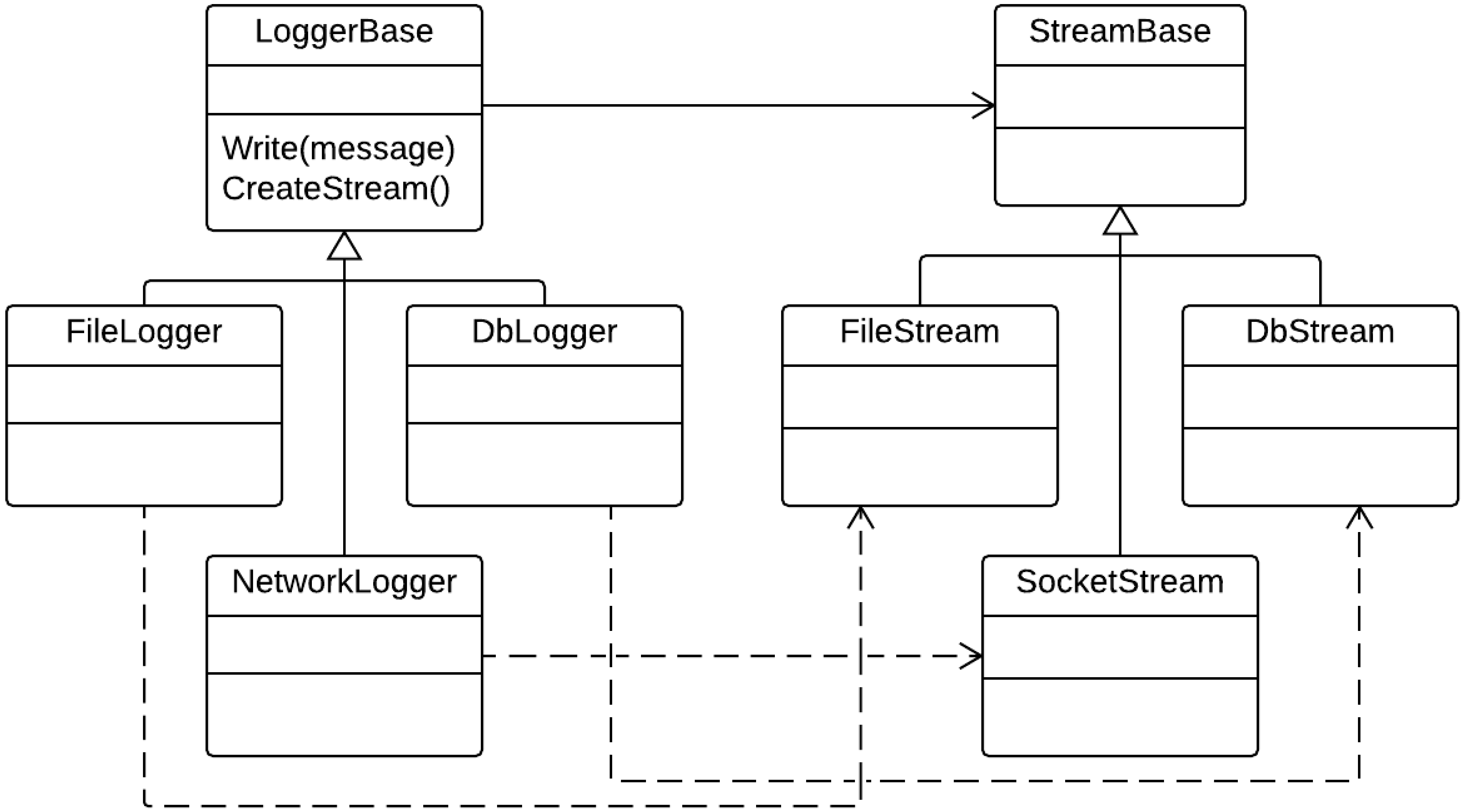
Prerequisites:

Abstract client

Each concrete client depends on one concrete product


LoggerBase
Write(message)
StreamFactory
CreateStream()
StreamBase
FileStream
DbStream
SocketStream
Open/close files
Transaction management
Connect/disconnect TCP socket


LoggerBase
Write(message)
StreamFactory
CreateStream()
StreamBase
FileLogger
DbLogger
NetworkLogger
FileStream
DbStream
SocketStream


LoggerBase
Write(message)
CreateStream()
StreamBase
FileLogger
DbLogger
FileStream
DbStream
NetworkLogger
SocketStream

Summary

Factory Method design pattern

- Original implementation relies on class inheritance
- Derived class provides concrete factory method implementation
- Useful if we already heave a hierarchy of classes
- Base class becomes Abstract Factory

Summary

Alternate implementation of the Factory Method

- Use lambda expressions to construct factory method on the fly
- Inject lambda expression into an object
- No need for derived classes

Summary

Alternate uses of the Factory Method

- Wrap Builder inside Factory Method
- Factory Method is covariant
- Builder was not covariant

Beyond Factory Method

- Constructing a complex object
- Would require a tree of nested Factory Methods

Next module –
Building Complex Objects with the Specification Pattern