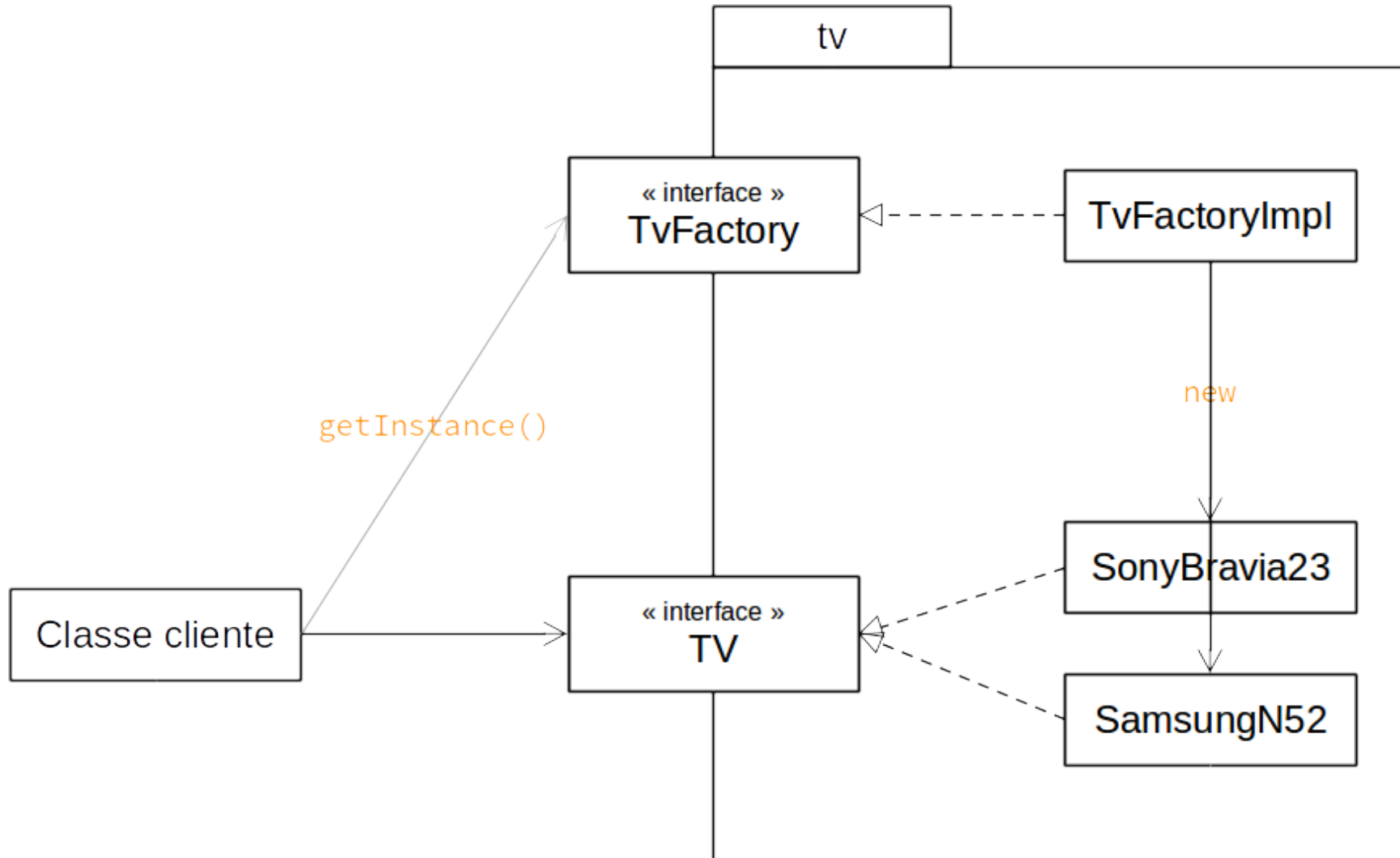


# Service Objects, Configuration

Leleux Laurent

2016 - 2017

# Factory



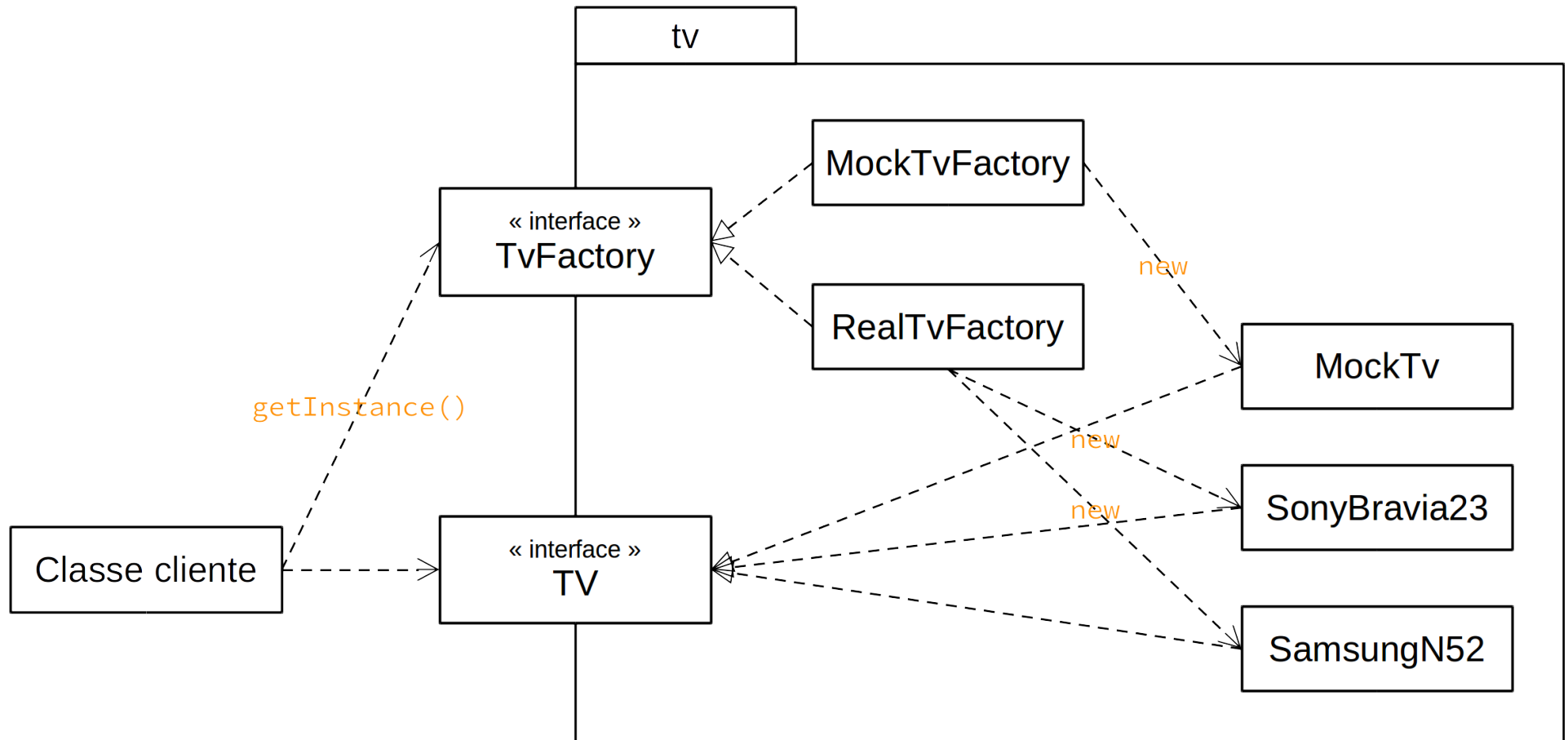
# Example

```
public interface TvFactory {  
  
    public static final TvFactory INSTANCE = new TvFactoryImpl();  
  
    Tv getTv();  
  
}  
  
public class TvFactoryImpl implements TvFactory {  
  
    @Override  
    public Tv getTv() {  
        return new SamsungN52();  
    }  
  
}  
  
// -----  
  
package client;  
  
import tv.Tv;  
import tv.TvFactory;  
  
public class Client {  
  
    public static void main(String[] args) {  
        Tv tv = TvFactory.INSTANCE.getTv();  
        tv.on();  
    }  
  
}
```

# Remarques

- Interface = Factory du Service Object
- Stateless = Immuable = Thread-safe
- Quasi-Singleton

# Multiple implementations



# Contexte

- Développement
- Production
- Changement dynamique
- Au lancement ?

# Properties

- Système clef/valeur
- Map<String, String>

```
#Fri Jan 17 22:37:45 MYT 2014  
dbpassword=password  
database=localhost  
dbuser=gerard
```

# Example

```
import java.io.FileInputStream;
import java.io.FileNotFoundException;
import java.io.IOException;
import java.util.Properties;

public class Main {

    public static void main(String[] args) {
        // Create and load properties
        Properties props = new Properties();
        FileInputStream input = null;
        try {
            input = new FileInputStream("src/props.properties");
            props.load(input);
        } catch (FileNotFoundException e) {
            e.printStackTrace();
            System.exit(1);
        } catch (IOException e) {
            e.printStackTrace();
            System.exit(2);
        } finally {
            if (input != null) {
                try {
                    input.close();
                } catch (IOException ignore) {
                }
            }
        }
        // Print properties
        System.out.println(props.getProperty("dbpassword"));
    }
}
```



# Documentation

- Recherche dans la JAVADOC...