

JavaFX Days

e(fx)clipse

Tom Schindl

e: tom.schindl@bestsolution.at

t: [@tomsontom](https://twitter.com/tomsontom)

The 3 faces

The 3 faces



Eclipse IDE-Plugin

The 3 faces



Eclipse IDE-Plugin



FX-Libraries

The 3 faces



Eclipse IDE-Plugin



FX-Libraries



FX-App-Framework

The 3 faces



Eclipse IDE-Plugin

- ❖ FXML-Editor
- ❖ CSS-Editor
- ❖ FXML & CSS Preview
- ❖ Project Wizards
- ❖ ...



FX-Libraries



FX-App-Framework

The 3 faces



Eclipse IDE-Plugin

- ❖ FXML-Editor
- ❖ CSS-Editor
- ❖ FXML & CSS Preview
- ❖ Project Wizards
- ❖ ...



FX-Libraries

- ❖ Binding-Extensions
- ❖ EMF-Integration
- ❖ Controls
- ❖ Layout-Container
- ❖ ...



FX-App-Framework

The 3 faces



Eclipse IDE-Plugin

- ❖ FXML-Editor
- ❖ CSS-Editor
- ❖ FXML & CSS Preview
- ❖ Project Wizards
- ❖ ...



FX-Libraries

- ❖ Binding-Extensions
- ❖ EMF-Integration
- ❖ Controls
- ❖ Layout-Container
- ❖ ...



FX-App-Framework

- ❖ OSGi-Integration
- ❖ e4-Framework
- ❖ ...

Libraries

Libraries



Core-Libraries

Libraries



Core-Libraries



UI-Libraries

Libraries



Core-Libraries

- ❖ Advanced Service System
- ❖ EventBus
- ❖ Advanced ThreadSynchronization
- ❖ Logger abstraction (incl FluentLogger)
- ❖ AnnotatedString
- ❖ Listener Utilities
- ❖ Binding Extensions
- ❖ Eclipse Databinding integration
- ❖ Eclipse Modeling Framework integration
- ❖ ...



UI-Libraries

Libraries



Core-Libraries

- ❖ Advanced Service System
- ❖ EventBus
- ❖ Advanced ThreadSynchronization
- ❖ Logger abstraction (incl FluentLogger)
- ❖ AnnotatedString
- ❖ Listener Utilities
- ❖ Binding Extensions
- ❖ Eclipse Databinding integration
- ❖ Eclipse Modeling Framework integration
- ❖ ...



UI-Libraries

- ❖ Layout containers
- ❖ Filesystem Viewers
- ❖ Multi colored text
- ❖ Preference System
- ❖ Docking support for TabPane
- ❖ Lightweight Dialog API
- ❖ Icon-FontNode
- ❖ List/Tree/Table Utilities
- ❖ Advanced ImageViewer
- ❖ ..



Core Libraries

Advanced Service System

Advanced Service System

- ❖ Need service system who works flawlessly in Java and OSGi

Advanced Service System

- ❖ Need service system who works flawlessly in Java and OSGi
- ❖ Need more advanced system than ServiceLoader

Advanced Service System

- ❖ Need service system who works flawlessly in Java and OSGi
- ❖ Need more advanced system than ServiceLoader
- ❖ **Solution:** Provide implementation who support OSGi-Style-Declarative Services in Plain Java

Advanced Service System

Advanced Service System

```
public interface HelloService {  
    public void sayHello(String name);  
}  
  
public interface TranslationService {  
    public String translate(String value, String fromLang, String toLang);  
}
```

Advanced Service System

```
public interface HelloService {  
    public void sayHello(String name);  
}  
  
public interface TranslationService {  
    public String translate(String value, String fromLang, String toLang);  
}
```

```
public class ConsoleHelloService implements HelloService {  
    private TranslationService translationService;  
  
    @Override  
    public void sayHello(String name) {  
        System.out.println(translationService.translate("Hello", "en", "de")+ " " + name + "!");  
    }  
}
```

Advanced Service System

```
public interface HelloService {  
    public void sayHello(String name);  
}  
  
public interface TranslationService {  
    public String translate(String value, String fromLang, String toLang);  
}
```

```
@Component(xmlns = "http://www.osgi.org/xmlns/scr/v1.2.0")  
public class ConsoleHelloService implements HelloService {  
    private TranslationService translationService;  
  
    @Reference(cardinality=ReferenceCardinality.MANDATORY)  
    public void setTranslationService(TranslationService translationService) {  
        this.translationService = translationService;  
    }  
  
    @Override  
    public void sayHello(String name) {  
        System.out.println(translationService.translate("Hello", "en", "de")+ " " + name + "!");  
    }  
}
```

Advanced Service System

Advanced Service System

❖ Service Discovery

Advanced Service System

❖ Service Discovery

```
public class ServiceSample {  
    public static void main(String[] args) {  
        ServiceUtils.getService(HelloService.class).ifPresent((s) -> {  
            s.sayHello("Zürich");  
        });  
    }  
}
```

EventBus

EventBus

- ❖ Publish and Subscribe EventBus API

EventBus

- ❖ Publish and Subscribe EventBus API
- ❖ Needs to work out- and inside OSGi (and e4)

EventBus

- ❖ Publish and Subscribe EventBus API
- ❖ Needs to work out- and inside OSGi (and e4)
- ❖ Needs to provide typesafety

EventBus

EventBus

```
private static final Topic<String> COUNT_MESSAGE = new Topic<>("javafxdays/count");
```

EventBus

```
private static final Topic<String> COUNT_MESSAGE = new Topic<>("javafxdays/count");

private static long COUNT;

public Sample(EventBus bus) {
    HBox box = new HBox();

    Button b = new Button("Count");
    Label l = new Label("0");
    box.getChildren().addAll(b,l);
}
```

EventBus

```
private static final Topic<String> COUNT_MESSAGE = new Topic<>("javafxdays/count");

private static long COUNT;

public Sample(EventBus bus) {
    HBox box = new HBox();

    Button b = new Button("Count");
    Label l = new Label("0");
    box.getChildren().addAll(b,l);

    b.setOnAction( evt -> {
        COUNT += 1;
        bus.publish(COUNT_MESSAGE, "Count: " + COUNT, true);
    });
}
```

EventBus

```
private static final Topic<String> COUNT_MESSAGE = new Topic<>("javafxdays/count");

private static long COUNT;

public Sample(EventBus bus) {
    HBox box = new HBox();

    Button b = new Button("Count");
    Label l = new Label("0");
    box.getChildren().addAll(b,l);

    b.setOnAction( evt -> {
        COUNT += 1;
        bus.publish(COUNT_MESSAGE, "Count: " + COUNT, true);
    });
    bus.subscribe(COUNT_MESSAGE, evt -> {
        l.setText(evt.getData());
    });
}
```

EventBus

```
private static final Topic<String> COUNT_MESSAGE = new Topic<>("javafxdays/count");

private static long COUNT;

public Sample(EventBus bus) {
    HBox box = new HBox();

    Button b = new Button("Count");
    Label l = new Label("0");
    box.getChildren().addAll(b,l);

    b.setOnAction( evt -> {
        COUNT += 1;
        bus.publish(COUNT_MESSAGE, "Count: " + COUNT, true);
    });
    bus.subscribe(COUNT_MESSAGE, EventBus.data(l::setText));
}
```

Advanced ThreadSynchronization

Advanced ThreadSynchronization

- ❖ Abstraction above Platform.runLater? Why?

Advanced ThreadSynchronization

- ❖ Abstraction above Platform.runLater? Why?
- ❖ Need more features

Advanced ThreadSynchronization

- ❖ Abstraction above Platform.runLater? Why?
- ❖ Need more features
 - ❖ Halting program flow but processing events (for backwards compatibility reasons)

Advanced ThreadSynchronization

- ❖ Abstraction above Platform.runLater? Why?
- ❖ Need more features
 - ❖ Halting program flow but processing events (for backwards compatibility reasons)
 - ❖ Delayed Executions

Advanced ThreadSynchronization

- ❖ Abstraction above Platform.runLater? Why?
- ❖ Need more features
 - ❖ Halting program flow but processing events (for backwards compatibility reasons)
 - ❖ Delayed Executions
 - ❖ Synchronous Executions (for backwards compatibility reasons)

Advanced ThreadSynchronization

- ❖ Abstraction above Platform.runLater? Why?
- ❖ Need more features
 - ❖ Halting program flow but processing events (for backwards compatibility reasons)
 - ❖ Delayed Executions
 - ❖ Synchronous Executions (for backwards compatibility reasons)
- ❖ ...

Advanced ThreadSynchronization

Advanced ThreadSynchronization

Schedule Execution

Advanced ThreadSynchronization

Schedule Execution

```
import org.eclipse.fx.core.ThreadSynchronize;

threadSync.scheduleExecution(1000*60, () -> {
    // Run in a minute from now on FX-Thread
});
```

Advanced ThreadSynchronization

Schedule Execution

```
import org.eclipse.fx.core.ThreadSynchronize;

threadSync.scheduleExecution(1000*60, () -> {
    // Run in a minute from now on FX-Thread
});
```

Delayed Bindings

Advanced ThreadSynchronization

Schedule Execution

```
import org.eclipse.fx.core.ThreadSynchronize;

threadSync.scheduleExecution(1000*60, () -> {
    // Run in a minute from now on FX-Thread
});
```

Delayed Bindings

```
StringProperty name = new SimpleStringProperty();
name.addListener( (ob, ol, ne) -> {
    // do some complex computations
} );
TextField f = new TextField();
name.bind(f.textProperty());
```

Advanced ThreadSynchronization

Schedule Execution

```
import org.eclipse.fx.core.ThreadSynchronize;

threadSync.scheduleExecution(1000*60, () -> {
    // Run in a minute from now on FX-Thread
});
```

Delayed Bindings

```
StringProperty name = new SimpleStringProperty();
name.addListener( (ob, ol, ne) -> {
    // do some complex computations
} );
TextField f = new TextField();
threadSync.delayedChangeExecution(200, f.textProperty(), name::set);
```

Advanced ThreadSynchronization

Advanced ThreadSynchronization

Sync back on UI-Thread

Advanced ThreadSynchronization

Sync back on UI-Thread

```
private static final Topic<String> COUNT_MESSAGE = new Topic<>("javafxdays/count");

private static long COUNT;

public Sample(EventBus bus) {
    HBox box = new HBox();

    Button b = new Button("Count");
    Label l = new Label("0");
    box.getChildren().addAll(b,l);

    b.setOnAction( evt -> {
        COUNT += 1;
        bus.publish(COUNT_MESSAGE, "Count: " + COUNT, true);
    });
    bus.subscribe(COUNT_MESSAGE, EventBus.data(l::setText));
}
```

Advanced ThreadSynchronization

Sync back on UI-Thread

```
private static final Topic<String> COUNT_MESSAGE = new Topic<>("javafxdays/count");

private static long COUNT;

public Sample(EventBus bus) {
    HBox box = new HBox();

    Button b = new Button("Count");
    Label l = new Label("0");
    box.getChildren().addAll(b,l);

    b.setOnAction( evt -> {
        COUNT += 1;
        bus.publish(COUNT_MESSAGE, "Count: " + COUNT, false);
    });
    bus.subscribe(COUNT_MESSAGE, EventBus.data(l::setText));
}
```

Advanced ThreadSynchronization

Sync back on UI-Thread

```
private static final Topic<String> COUNT_MESSAGE = new Topic<>("javafxdays/count");

private static long COUNT;

public Sample(EventBus bus) {
    HBox box = new HBox();

    Button b = new Button("Count");
    Label l = new Label("0");
    box.getChildren().addAll(b,l);

    b.setOnAction( evt -> {
        COUNT += 1;
        bus.publish(COUNT_MESSAGE, "Count: " + COUNT, false);
    });
    Consumer<String> wrappedAccess = threadSync.wrap(l::setText);
    bus.subscribe(COUNT_MESSAGE, EventBus.data(wrappedAccess));
}
```

Listener Utilities

Listener Utilities

Problem domain

Listener Utilities

Problem domain

```
StringProperty name = new SimpleStringProperty();
name.addListener( (ob, ol, ne) -> {
    // do some complex computations
} );

ObservableList<String> l = FXCollections.observableArrayList();
l.addListener( c -> {
    // process changes
});
```

Listener Utilities

Problem domain

```
StringProperty name = new SimpleStringProperty();
name.addListener( (ob, ol, ne) -> {
    // do some complex computations
} );

ObservableList<String> l = FXCollections.observableArrayList();
l.addListener( c -> {
    // process changes
});
```

```
StringProperty name = new SimpleStringProperty();
FXObservableUtil.onChange(name, ne -> {
    // do some complex computations
});

ObservableList<String> l = FXCollections.observableArrayList();
FXObservableUtil.onChange(l, c -> {
    // process changes
});
```

Binding Extensions

Binding Extensions

```
public class Person {
    ObjectProperty<Address> primaryAddress = new SimpleObjectProperty<>();

    public final ObjectProperty<Address> primaryAddressProperty() {
        return this.primaryAddress;
    }

    // ....
}

public class Address {
    private IntegerProperty streeNr = new SimpleIntegerProperty();

    public final IntegerProperty streeNrProperty() {
        return this.streeNr;
    }

    // ...
}
```

Binding Extensions

Binding Extensions

```
TextField houseNumber = new TextField();
```

Binding Extensions

```
TextField houseNumber = new TextField();  
  
public void bindMaster(ObjectProperty<Person> master) {
```

Binding Extensions

```
TextField houseNumber = new TextField();

public void bindMaster(ObjectProperty<Person> master) {
    IntegerProperty hnrProperty = FXBindings.bindStream( master )
}
```

Binding Extensions

```
TextField houseNumber = new TextField();

public void bindMaster(ObjectProperty<Person> master) {
    IntegerProperty hnrProperty = FXBindings.bindStream( master )
        .map(Person::primaryAddressProperty)
```

Binding Extensions

```
TextField houseNumber = new TextField();

public void bindMaster(ObjectProperty<Person> master) {
    IntegerProperty hnrProperty = FXBindings.bindStream( master )
        .map(Person::primaryAddressProperty)
        .collect(FXCollectors.toIntegerProperty(Address::streetNrProperty));
}
```

Binding Extensions

```
TextField houseNumber = new TextField();

public void bindMaster(ObjectProperty<Person> master) {
    IntegerProperty hnrProperty = FXBindings.bindStream( master )
        .map(Person::primaryAddressProperty)
        .collect(FXCollectors.toIntegerProperty(Address::streeNrProperty));

    StatusBinding bindingStatus = FXBindings.bindBidirectional(houseNumber.textProperty(),
```

Binding Extensions

```
TextField houseNumber = new TextField();

public void bindMaster(ObjectProperty<Person> master) {
    IntegerProperty hnrProperty = FXBindings.bindStream( master )
        .map(Person::primaryAddressProperty)
        .collect(FXCollectors.toIntegerProperty(Address::streetNrProperty));

    StatusBinding bindingStatus = FXBindings.bindBidirectional(houseNumber.textProperty(),
        hnrProperty,
```

Binding Extensions

```
TextField houseNumber = new TextField();

public void bindMaster(ObjectProperty<Person> master) {
    IntegerProperty hnrProperty = FXBindings.bindStream( master )
        .map(Person::primaryAddressProperty)
        .collect(FXCollectors.toIntegerProperty(Address::streetNrProperty));

    StatusBinding bindingStatus = FXBindings.bindBidirectional(houseNumber.textProperty(),
        hnrProperty,
        Integer::parseInt,
```

Binding Extensions

```
TextField houseNumber = new TextField();

public void bindMaster(ObjectProperty<Person> master) {
    IntegerProperty hnrProperty = FXBindings.bindStream( master )
        .map(Person::primaryAddressProperty)
        .collect(FXCollectors.toIntegerProperty(Address::streetNrProperty));

    StatusBinding bindingStatus = FXBindings.bindBidirectional(houseNumber.textProperty(),
        hnrProperty,
        Integer::parseInt,
        String::valueOf);
}
```

Binding Extensions

```
TextField houseNumber = new TextField();

public void bindMaster(ObjectProperty<Person> master) {
    IntegerProperty hnrProperty = FXBindings.bindStream( master )
        .map(Person::primaryAddressProperty)
        .collect(FXCollectors.toIntegerProperty(Address::streeNrProperty));

    StatusBinding bindingStatus = FXBindings.bindBidirectional(houseNumber.textProperty(),
        hnrProperty,
        Integer::parseInt,
        String::valueOf);

    FXObservableUtil.onChange(bindingStatus, v -> {
        houseNumber.pseudoClassStateChanged(ERROR, v.isNotOk());
    });
}
```

Annotated String

- ❖ For UIs we need a CharSequence who can carry meta information (eg Styleing)
- ❖ AnnotatedString is an IMMUTABLE CharSequence like a String but allows to attach meta data to ranges

Annotated String

Annotated String

```
import org.eclipse.fx.core.text.AnnotatedString;
```

Annotated String

```
import org.eclipse.fx.core.text.AnnotatedString;

AnnotatedString<String> val = AnnotatedString.create(String.class, 30)
    .add("public ", "modifier")
    .add("static ", "modifier")
    .add("void ", "keyword")
    .add("main", "method","identifier")
    .add("(", "brace")
    .add("String[] ", "type")
    .add("args", "parameter","identifier")
    .add(")", "brace")
    .build();
```

Annotated String

```
import org.eclipse.fx.core.text.AnnotatedString;
```

```
AnnotatedString<String> val = AnnotatedString.create(String.class, 30)
    .add("public ", "modifier")
    .add("static ", "modifier")
    .add("void ", "keyword")
    .add("main", "method","identifier")
    .add("(", "brace")
    .add("String[] ", "type")
    .add("args", "parameter","identifier")
    .add(")", "brace")
    .build();
```

```
AnnotatedString<Meta> val2 = AnnotatedString.create(Meta.class, 30)
    .add("public ", Meta.MODIFIER)
    .add("static ", Meta.MODIFIER)
    .add("void ", Meta.KEYWORD)
    .add("main", Meta.METHOD, Meta.IDENTIFIER)
    .add("(", Meta.BRACE)
    .add("String[] ", Meta.TYPE)
    .add("args", Meta.PARAMETER, Meta.IDENTIFIER)
    .add(")", Meta.BRACE)
    .build();
```

Logging

- ❖ What logging API should a framework depend on?
- ❖ Our answer: **None** we ship a custom API and provide the users integration points into their logging system

Logging API

- ❖ 2 different APIs available
 - ❖ Traditional log4j like API
 - ❖ Fluent API inspired by Google Fluent Logger

Logging API

Logging API

Traditional Log API

Logging API

Traditional Log API

```
private static Logger LOGGER = LoggerCreator.createLogger(LoggerSample.class);

int i = 0;
try {
    for( i = 0; i < 1_000_000; i++ ) {
        LOGGER.debugf("Progress %s", i);
    }
} catch( Throwable t ) {
    LOGGER.errorf("Failure while processing Item %s", t, i);
}
```

Logging API

Traditional Log API

```
private static Logger LOGGER = LoggerCreator.createLogger(LoggerSample.class);

int i = 0;
try {
    for( i = 0; i < 1_000_000; i++ ) {
        LOGGER.debugf("Progress %s", i);
    }
} catch( Throwable t ) {
    LOGGER.errorf("Failure while processing Item %s", t, i);
}
```

Fluent Log API

Logging API

Traditional Log API

```
private static Logger LOGGER = LoggerCreator.createLogger(LoggerSample.class);

int i = 0;
try {
    for( i = 0; i < 1_000_000; i++ ) {
        LOGGER.debugf("Progress %s", i);
    }
} catch( Throwable t ) {
    LOGGER.errorf("Failure while processing Item %s", t, i);
}
```

Fluent Log API

```
private static Logger LOGGER = LoggerCreator.createLogger(LoggerSample.class);
private static FluentLogger FLOGGER = FluentLogger.of(LOGGER);

FluentLogContext debug = FLOGGER.atDebug().throttleByCount(1_000);
FluentLogContext error = FLOGGER.atError();
int i = 0;
try {
    for( i = 0; i < 1_000_000; i++ ) {
        debug.log("Progress %s", i);
    }
} catch( Throwable t ) {
    error.withException(t).log("Failure while processing Item %s", i);
}
```



Frameworks

JavaFX-8 in OSGi

JavaFX-8 in OSGi

- ❖ Problem 1: JavaFX is on the extension class path

JavaFX-8 in OSGi

- ❖ Problem 1: JavaFX is on the extension class path
- ❖ Solution: Run with
`org.osgi.framework.bundle.parent=ext`

JavaFX-8 in OSGi

- ❖ Problem 1: JavaFX is on the extension class path
- ❖ Solution: Run with
`org.osgi.framework.bundle.parent=ext`
- ❖ Problem 2: swt-fx is on no classpath (because it depends on SWT)

JavaFX-8 in OSGi

- ❖ Problem 1: JavaFX is on the extension class path
- ❖ Solution: Run with
`org.osgi.framework.bundle.parent=ext`
- ❖ Problem 2: swt-fx is on no classpath (because it depends on SWT)
- ❖ Use Equinox Adapter hooks to create a class loader who knows the SWT-Classloader

JavaFX-11 in OSGi

- ❖ JavaFX is not shipped with Java-11 anymore
- ❖ JavaFX expect to run as Java-Module on the module-path

JavaFX-11 in OSGi

- ❖ Use Equinox-Adapter Hooks and spin an extra Java-Module-Layer dynamically

Demo: <https://gist.github.com/tomsontom/5d8bd7fbfcbd9ec41090cf9a7a2d2b45>

JavaFX-11 in OSGi

- ❖ Use Equinox-Adapter Hooks and spin an extra Java-Module-Layer dynamically

Java-11-Root-Layer (java.base, ...)

Demo: <https://gist.github.com/tomsontom/5d8bd7fbfcbd9ec41090cf9a7a2d2b45>

JavaFX-11 in OSGi

- ❖ Use Equinox-Adapter Hooks and spin an extra Java-Module-Layer dynamically

OSGi-Bundles (unnamed-module)

Java-11-Root-Layer (java.base, ...)

Demo: <https://gist.github.com/tomsontom/5d8bd7fbfcbd9ec41090cf9a7a2d2b45>

JavaFX-11 in OSGi

- ❖ Use Equinox-Adapter Hooks and spin an extra Java-Module-Layer dynamically

OSGi-Bundles (unnamed-module)

**JavaFX-Layer
(javafx.base,
...)**

Java-11-Root-Layer (java.base, ...)

Demo: <https://gist.github.com/tomsontom/5d8bd7fbfcbd9ec41090cf9a7a2d2b45>

JavaFX-11 in OSGi

- ❖ Locating JavaFX-Modules
 - ❖ Use a command line option `-Dfxclipse.java-modules.dir=...`
 - ❖ Ship JavaFX-Bundles wrapped in an OSGi-Bundle with a special MANIFEST.MF-Header `Java-Module: ...`
- ❖ Prepackaged versions:
 - ❖ p2: <http://downloads.efxclipse.bestsolution.at/p2-repos/openjfx-11/repository/>
 - ❖ maven: <http://maven.bestsolution.at/efxclipse-releases/at/bestsolution/openjfx/>

JavaFX-11 in OSGi

JavaFX-11 in OSGi

- ❖ Support for Advanced Features via MANIFEST.MF

JavaFX-11 in OSGi

- ❖ Support for Advanced Features via MANIFEST.MF
 - ❖ Java-Module-AddOpens

JavaFX-11 in OSGi

- ❖ Support for Advanced Features via MANIFEST.MF
 - ❖ Java-Module-AddOpens
 - ❖ Java-Module-AddExports

JavaFX-11 in OSGi

- ❖ Support for Advanced Features via MANIFEST.MF
 - ❖ Java-Module-AddOpens
 - ❖ Java-Module-AddExports
 - ❖ Java-Module-AddReads

JavaFX-11 in OSGi

- ❖ Support for Advanced Features via MANIFEST.MF
 - ❖ Java-Module-AddOpens
 - ❖ Java-Module-AddExports
 - ❖ Java-Module-AddReads

```
Manifest-Version: 1.0
Bundle-ManifestVersion: 2
Bundle-Name: Fx
Bundle-SymbolicName: org.eclipse.fx.ui.workbench.fx;singleton:=true
Java-Module-AddExports: javafx.graphics/com.sun.javafx.application=.
```

e4 on JavaFX

- ❖ Application Model: Keeps the complete state of your application (windows-sizes, ...)
- ❖ Dependency Injection Container: Provides your components access to state and services
- ❖ Everything is a services: Rendering, Selection-Propagation, ModelPersistence, ...

e4 on JavaFX

- ❖ e4 designed from the beginning to be UI-Toolkit agnostic
- ❖ ships with default service implementations who can be replaced

Future e4 on JavaFX

Future e4 on JavaFX

- ❖ Development Support for none Eclipse IDEs
 - ❖ VS-Code, Netbeans, IntelliJ
 - ❖ In Beta-State

Future e4 on JavaFX

- ❖ Development Support for none Eclipse IDEs
 - ❖ VS-Code, Netbeans, IntelliJ
 - ❖ In Beta-State
- ❖ Make use of OSGi optional
 - ❖ PoC are done but no time plan yet unless we get funding

Demo

e(fx)clipse - Drift

What is $e(fx)$ clipse Drift

What is $e(fx)$ clipse Drift

Direct

What is $e(fx)$ clipse Drift

Direct Rendering

What is e(fx)clipse Drift

Direct Rendering Infrastructure

What is e(fx)clipse Drift

DirectRenderingInFrastrucTure

Drift started as a **research** project of multiple companies to integrate **existing** native rendering engines (most importantly OpenGL) into JavaFX applications to provide a business UI

Partner companies

Partner companies



Netallied - NetAllied develops 3D visualization software for the automotive and aviation industry

Partner companies



Netallied - NetAllied develops 3D visualization software for the automotive and aviation industry



EclipseSource Munich - We support our customers in efficiently creating tools for their specific domain.

Partner companies



Netallied - NetAllied develops 3D visualization software for the automotive and aviation industry



EclipseSource Munich - We support our customers in efficiently creating tools for their specific domain.



BestSolution

BestSolution - JavaFX and Eclipse consulting and development company

What can be built with it?

What can be built with it?

Drift

Drift

JVM

Drift

JVM

JavaFX

Drift

JVM

JavaFX

**Dirft
Java-API**

Drift

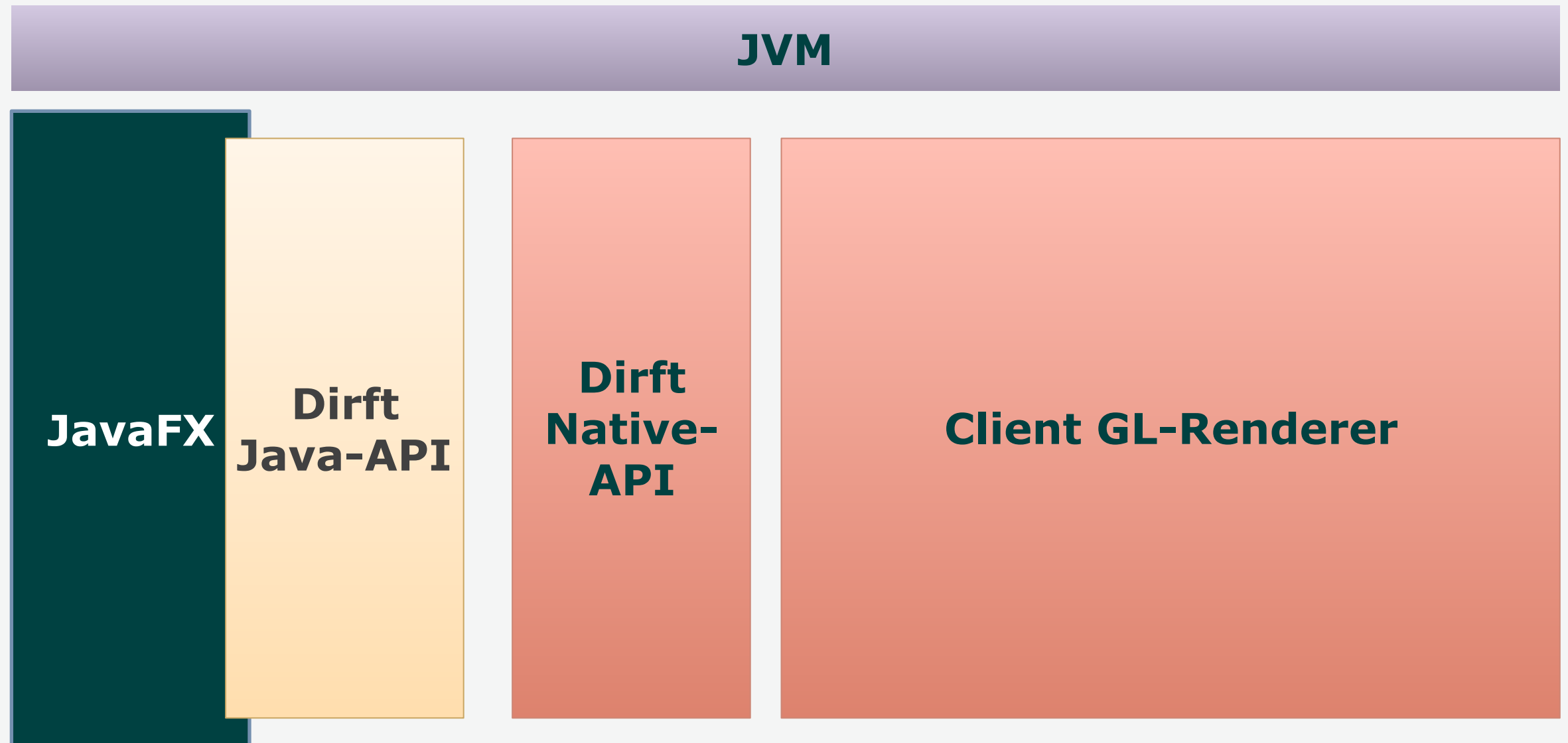
JVM

JavaFX

**Dirft
Java-API**

**Dirft
Native-
API**

Drift



Drift

Drift

JavaFX

Drift

JavaFX

DriftFX

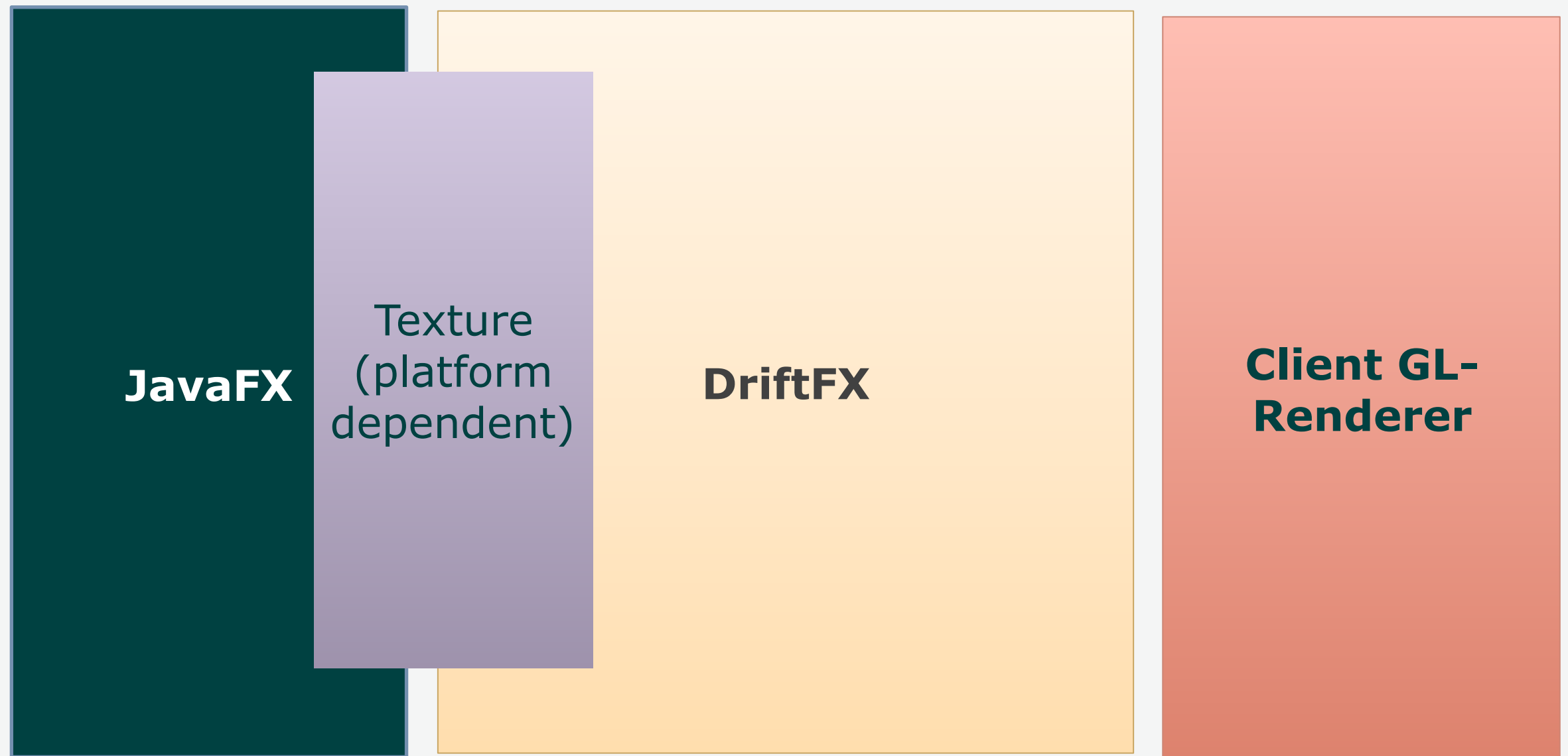
Drift

JavaFX

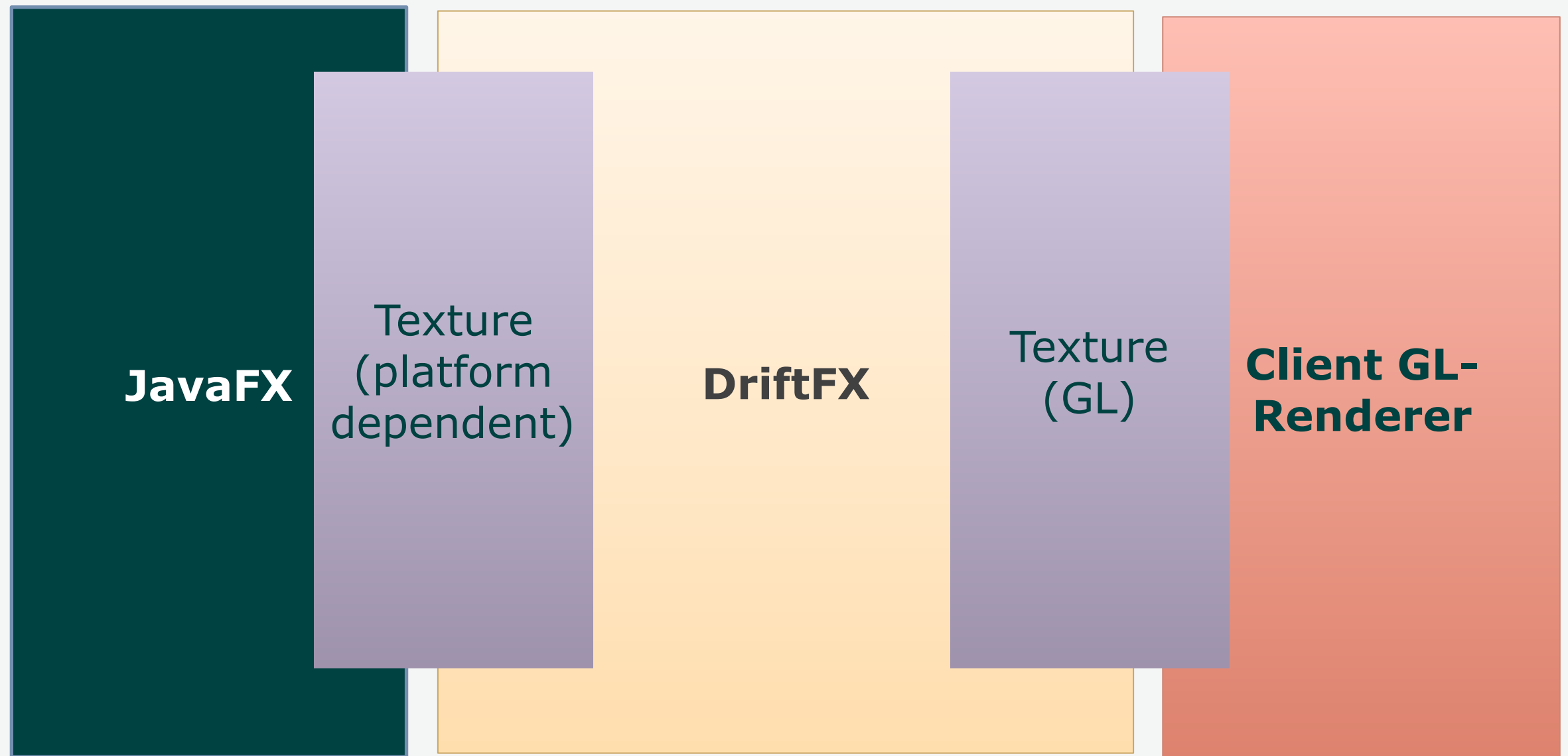
DriftFX

**Client GL-
Renderer**

Drift



Drift



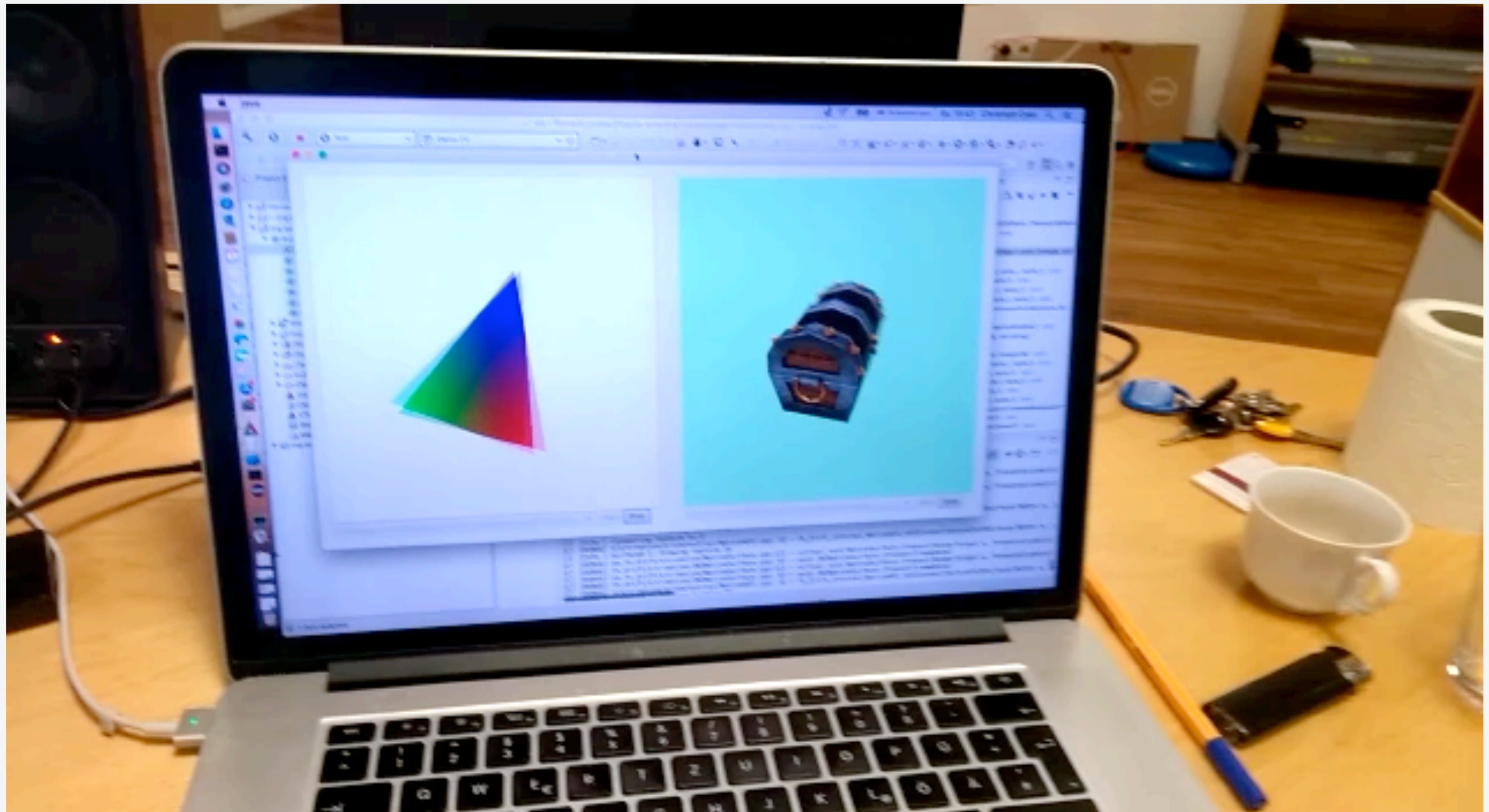
In-Process-Support

	D3D	OpenGL	Metal	Vulkan
Windows				
OS-X				
Linux				

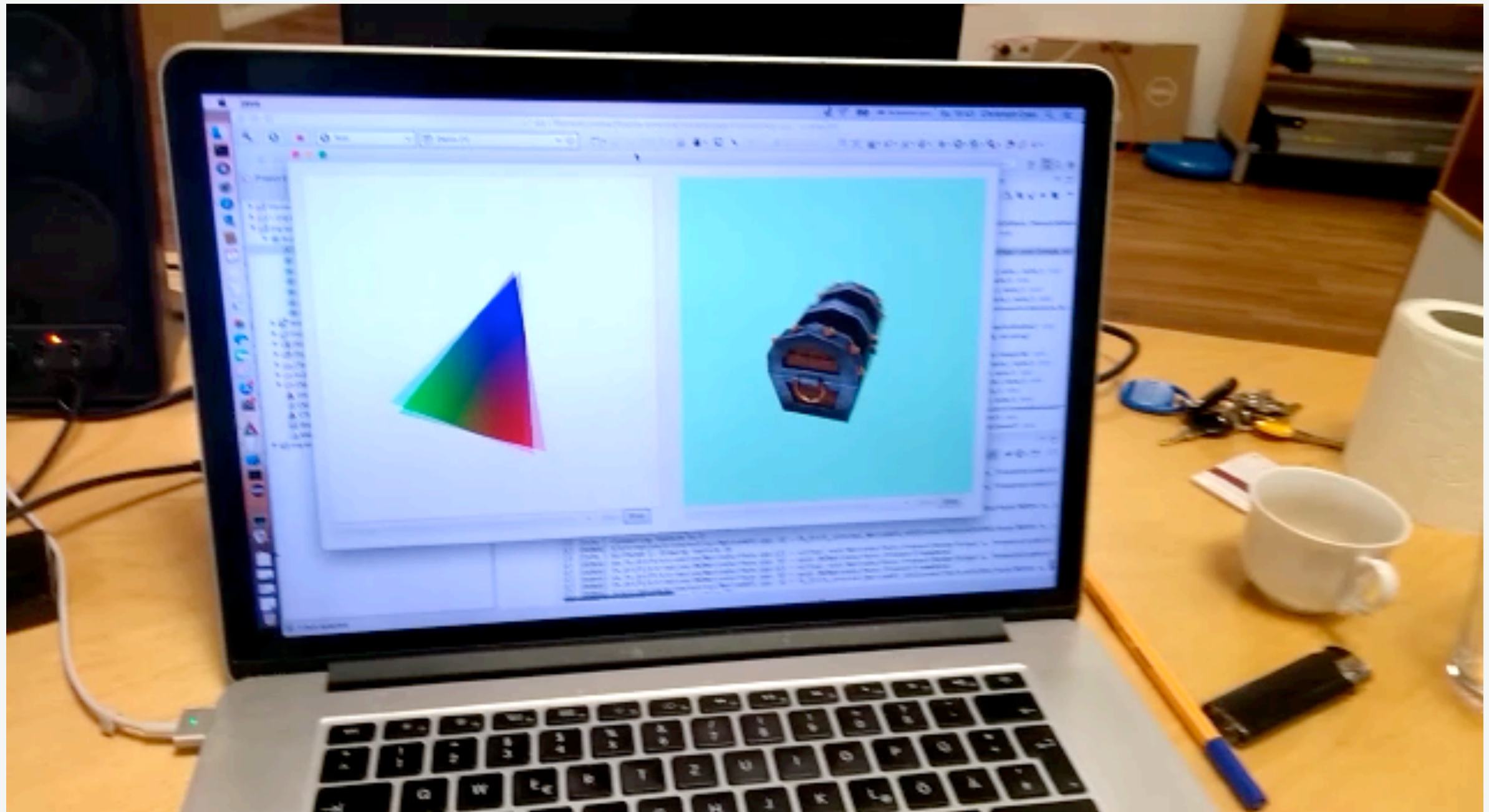
In-Process-Support

	D3D	OpenGL	Metal	Vulkan
Windows	✓	✓		?
OS-X		✓	?	?
Linux		✓		?

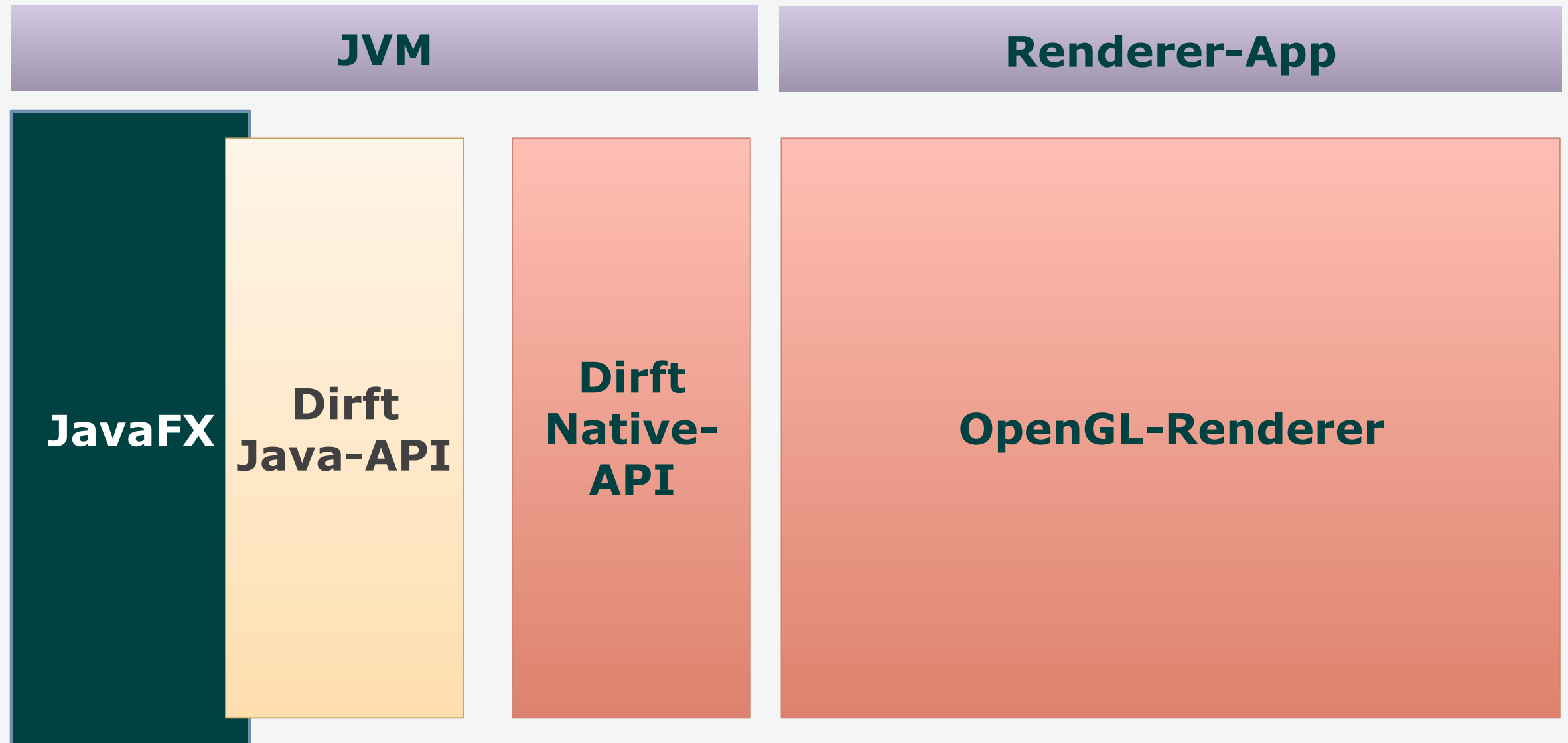
Drift in Action



Drift in Action



Drift Multi-Process



Multi-Process-Support

	D3D	OpenGL	Metal	Vulkan
Windows				
OS-X				
Linux				

Drift - Announcement

Drift - Announcement

The screenshot shows a GitHub pull request titled "intial implementation #1" by user redrezo. The pull request is open and ready for review, with all checks passed. The interface includes a top navigation bar with the GitHub logo, a search bar, and links to Pull requests, Issues, Marketplace, and Explore. The repository name "eclipse-efx / efxclipse-drift" is displayed at the top, along with statistics for Unwatch (6), Star (0), and Fork (4). The pull request details show that redrezo wants to merge 1 commit into eclipse-efx:master from eclipsesource:master. The pull request has 1 commit and 160 files changed, with a net change of +72,632 lines and -2 deletions. A comment by redrezo, posted 40 minutes ago, states "No description provided." The pull request is reviewed by tomsontom, who has requested changes. The pull request is assigned to no one. The pull request has no labels, projects, or milestones. The pull request has notifications enabled. The pull request is ready to be merged, with a green "Merge pull request" button. The pull request is also available in GitHub Desktop or via command line instructions.

intial implementation by redrezo

GitHub, Inc. (US) | <https://github.com/eclipse-efx/efxclipse-drift/pull/1>

Search or jump to... Pull requests Issues Marketplace Explore

eclipse-efx / efxclipse-drift

Unwatch 6 Star 0 Fork 4

Code Issues 0 Pull requests 1 Projects 0 Wiki Insights

intial implementation #1 Edit

Open redrezo wants to merge 1 commit into eclipse-efx:master from eclipsesource:master

Conversation 0 Commits 1 Checks 0 Files changed 160 +72,632 -2

redrezo commented 40 minutes ago Member

No description provided.

intial implementation 2dad2a

Add more commits by pushing to the **master** branch on eclipsesource/efxclipse-fxgl.

All checks have passed 1 successful check Show all checks

This branch has no conflicts with the base branch Merging can be performed automatically.

Merge pull request You can also open this in GitHub Desktop or view command line instructions.

Write Preview AA B i “ < > @ * ↩

Leave a comment

Reviewers Suggestions tomsontom Request

Assignees No one—assign yourself

Labels None yet

Projects None yet

Milestone No milestone

Notifications Unsubscribe You're receiving notifications because you're watching this

We are hiring!