

LP4J

A song of Java, MIDI
and websockets

Olivier Croisier



Breizh C@mp



Olivier Croisier

Freelance @ Moka Technologies

Expert Java certifié
Formateur
Speaker
Blogger



TheCodersBreakfast.net



olivier.croisier@mokatech.net



@OlivierCroisier



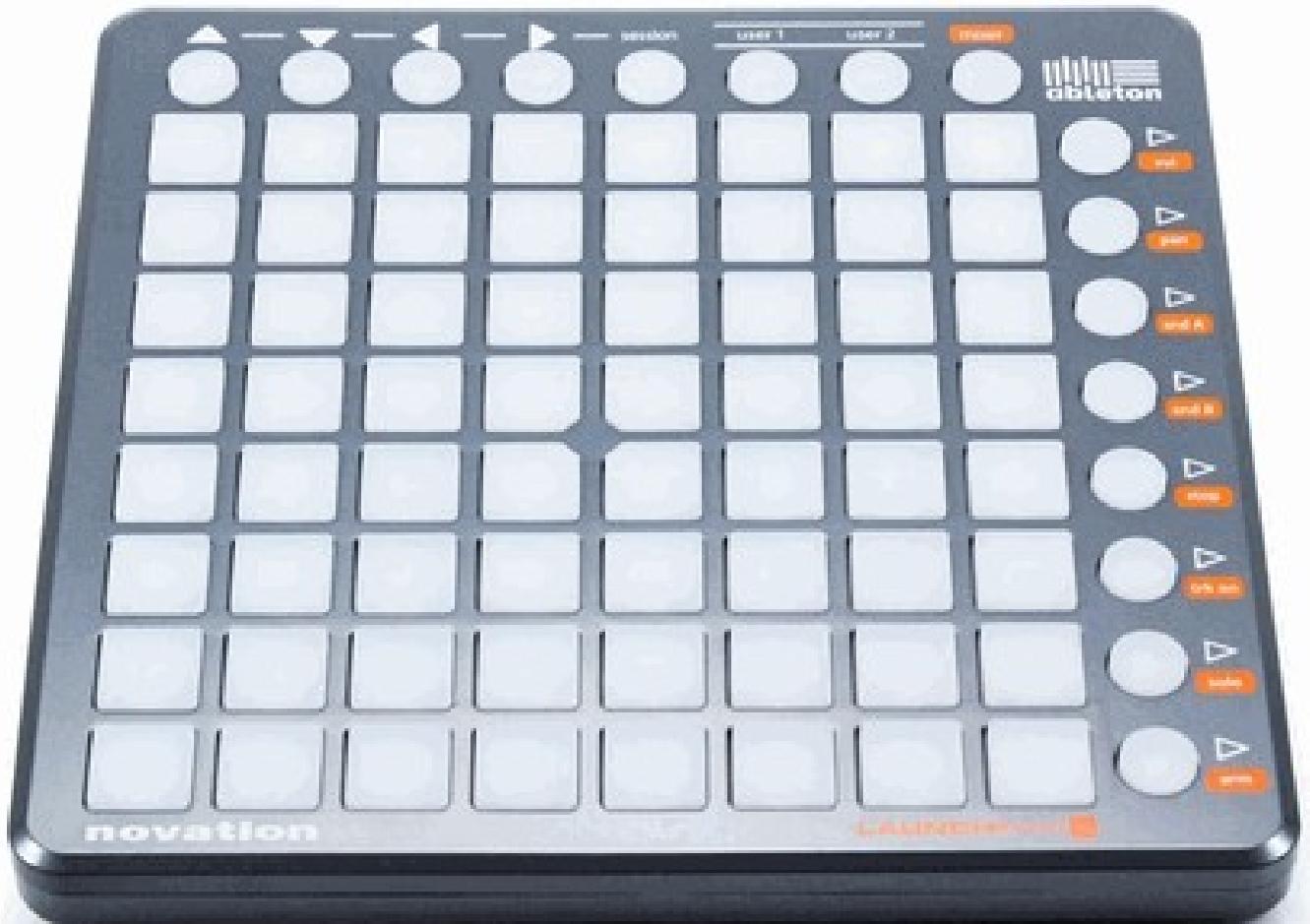
github.com/OlivierCroisier

Launchpad ?

Contrôleur MIDI
64 pads + 16 boutons lumineux

Pilotage de MAO

Daft Punk
Madeon
...
(moi?)



LP4J



LP4J

API Java pour piloter un Launchpad

Launchpad → LP4J

Pression sur un pad ou un bouton

LP4J → Launchpad

Eclairage

Défilement de texte

Buffers, luminosité, reset

API + 2 implémentations

MIDI, émulateur web

API

Launchpad
Listener

Launchpad

Launchpad
Client

Pad

Button

Color

Brightness

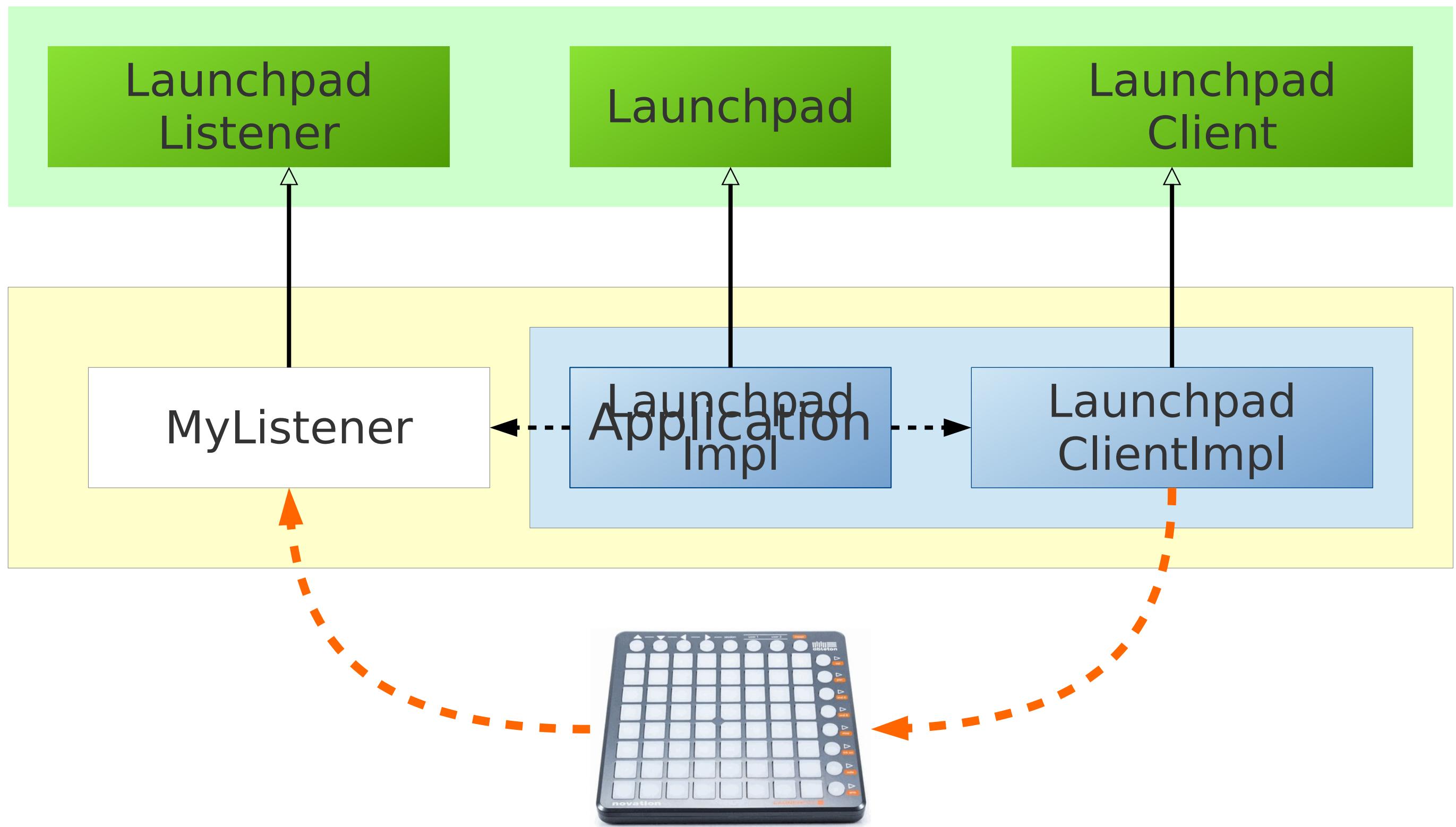
LightIntensity

Buffer

BackBufferOperation

ScrollSpeed

API



API

Launchpad

getClient

setListener

LaunchpadListener

onXXXPressed|Released(Pad/Button)

onTextScrolled

LaunchpadClient

reset

setPadLight / setButtonLight

scrollText

setBuffers

Démo

```
Launchpad launchpad = new <implémentation>;
LaunchpadClient client = launchpad.getClient();
launchpad.setListener(new MyListener(client));

class MyListener extends LaunchpadListenerAdapter {
    private LaunchpadClient client;

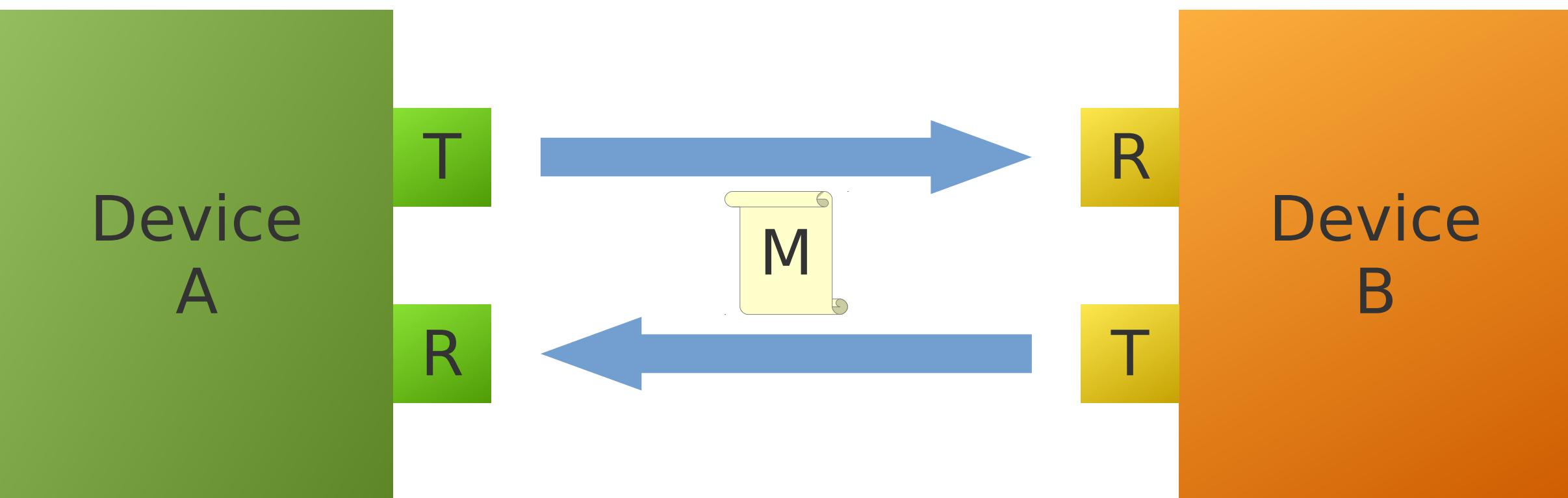
    public MyListener(LaunchpadClient client) {
        this.client = client;
    }

    public void onPadPressed(Pad pad, long timestamp) {
        client.setPadLight(pad, Color.RED, NONE);
    }
}
```

Protocole MIDI

`javax.sound.midi`

MidiDevice
Transmitter / Receiver
MidiMessage



Messages MIDI

`javax.sound.midi.MidiMessage`

`ShortMessage` → notes

`MetaMessage` → paroles, copyright, tempo, clé...

`SysexMessage` → messages systèmes exclusifs

```
MidiMessage message = new ShortMessage(  
    ShortMessage.NOTE_ON, note, velocity);
```

Protocole propriétaire

Envoi d'un message

Récupération d'un **Receiver**
Auprès du device / logiciel destinataire

```
MidiDevice device = MidiSystem.getMidiDevice(deviceId);  
if (device.getMaxReceivers() == -1) {  
    return device.getReceiver();  
}
```

Receiver.send(message)

```
MidiMessage message = ... ;  
Receiver receiver = device.getReceiver();  
receiver.send(message)
```

Réception d'un message

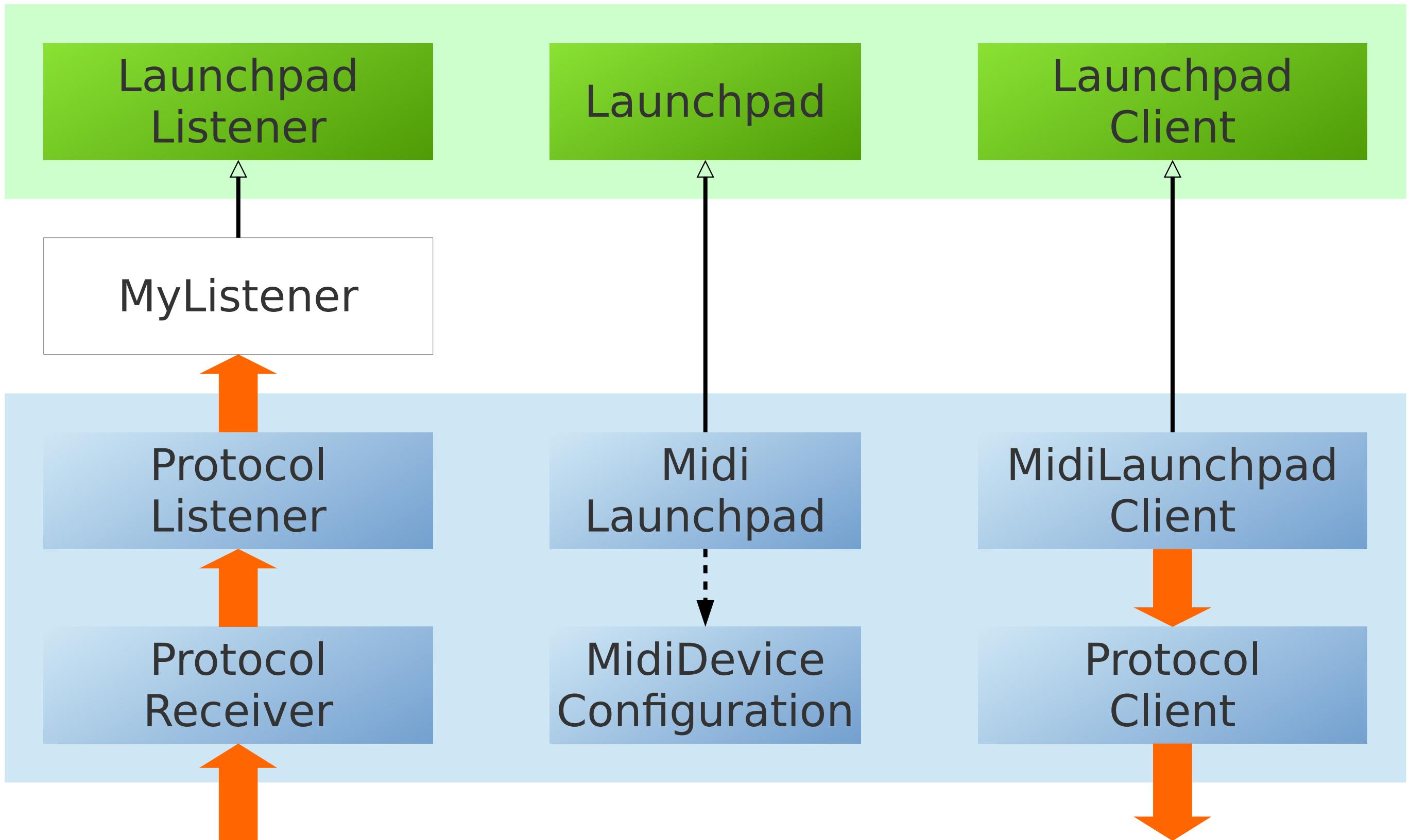
Récupération d'un **Transmitter**
Auprès du device / logiciel destinataire

```
MidiDevice device = MidiSystem.getMidiDevice(deviceId);  
if (device.getMaxTransmitters() == -1) {  
    return device.getTransmitter();  
}
```

Association d'un Receiver au Transmitter

```
Receiver midiReceiver = new MyReceiver();  
transmitter.setReceiver(midiReceiver);
```

LP4J-midi



LP4J-midi

MidiDeviceConfiguration
Configuration manuelle
Auto-détection (« Launchpad S »)

MidiLaunchpad
Implémentation de l'interface Launchpad

```
MidiDeviceConfiguration configuration =  
MidiDeviceConfiguration.autodetect();
```

```
Launchpad launchpad = new MidiLaunchpad(configuration);
```

LP4J-emu-web

Utiliser LP4J sans Launchpad physique

Implémentation

Client lourd

Application web

Contraintes

Simplicité de mise en place

Principe de substitution

Latence

LP4J-emu-web

Communication bi-directionnelle temps réel
Sockets

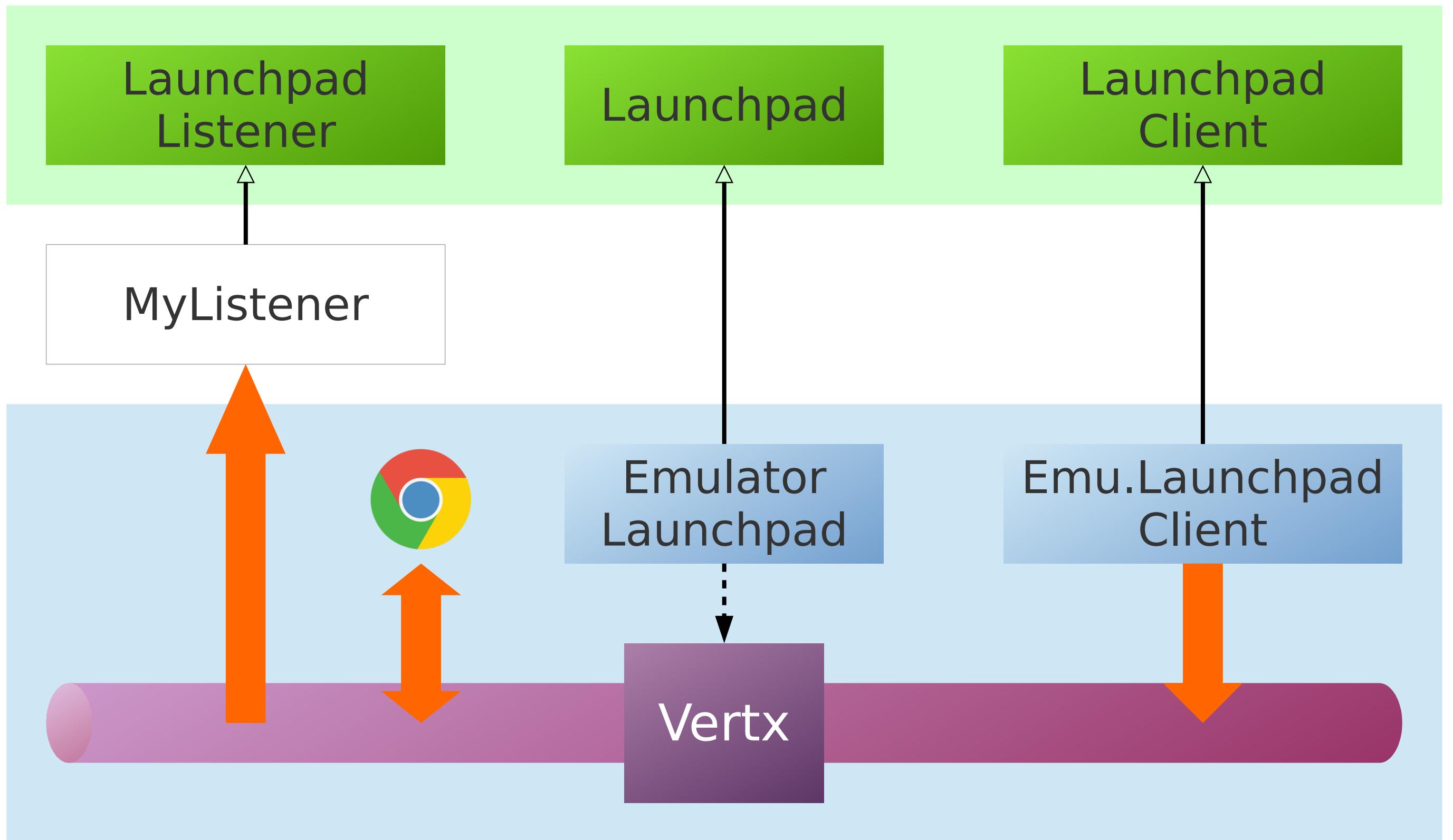
~~RMI~~

~~HTTP~~

Websockets !

Emulateur « embedded »
Vertx core, websockets, SVG

LP4J-emu-web



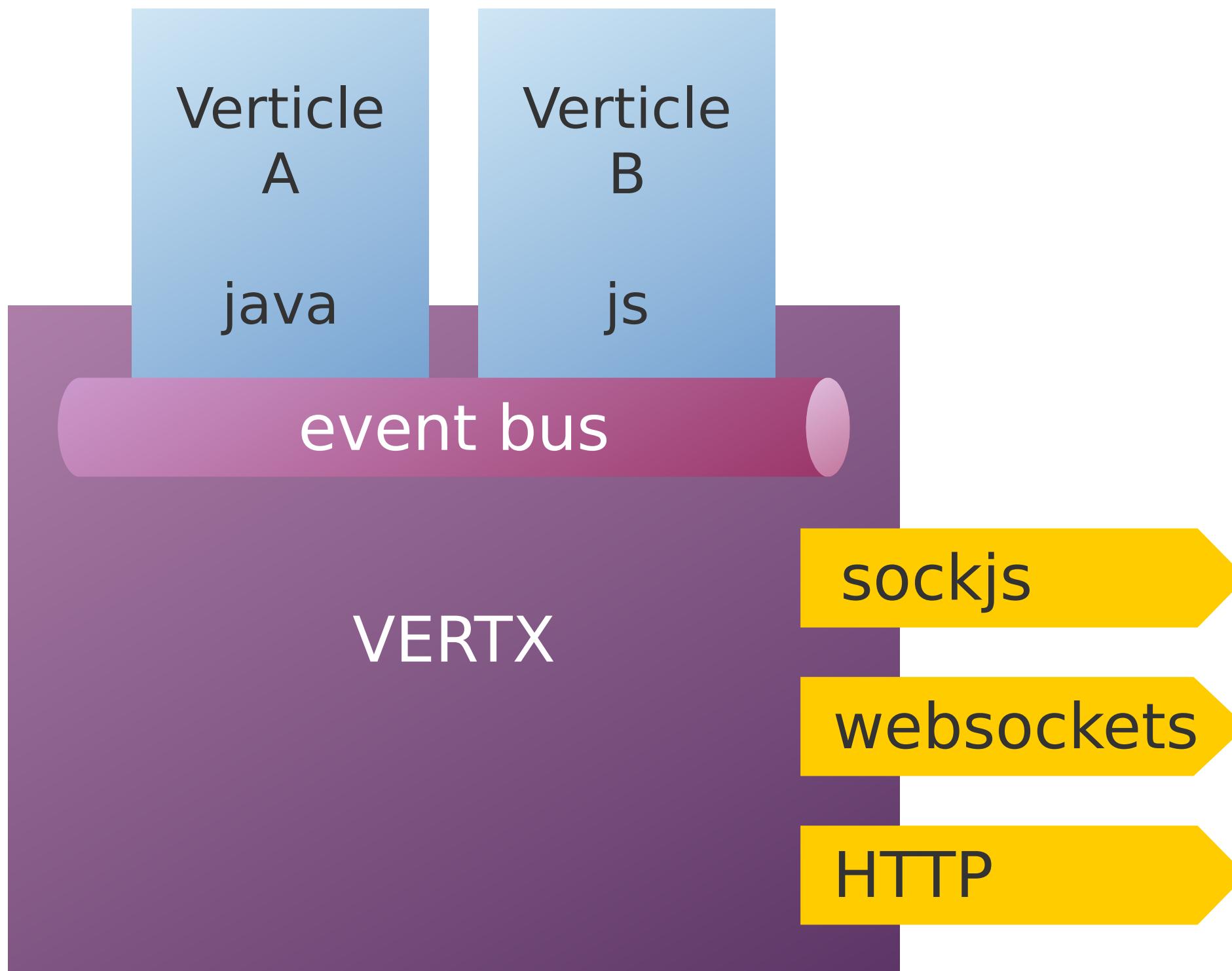
Vertx

Serveur léger pour la JVM
Réactif (~NodeJS)
Polyglotte
Bus d'événements

Exécute des Verticles (« modules »)

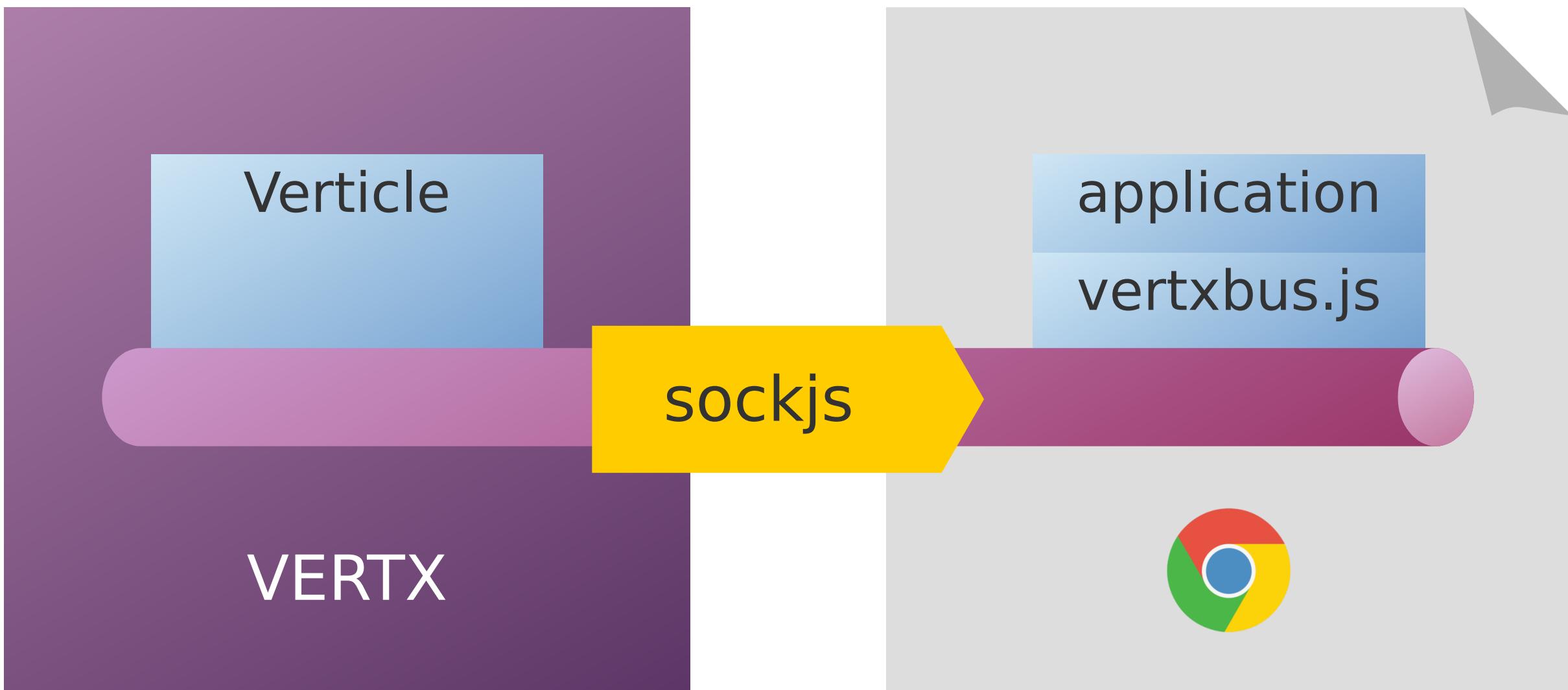
Connecteurs externes
Serveur HTTP
Serveur websockets
Serveur SockJS

Vertx

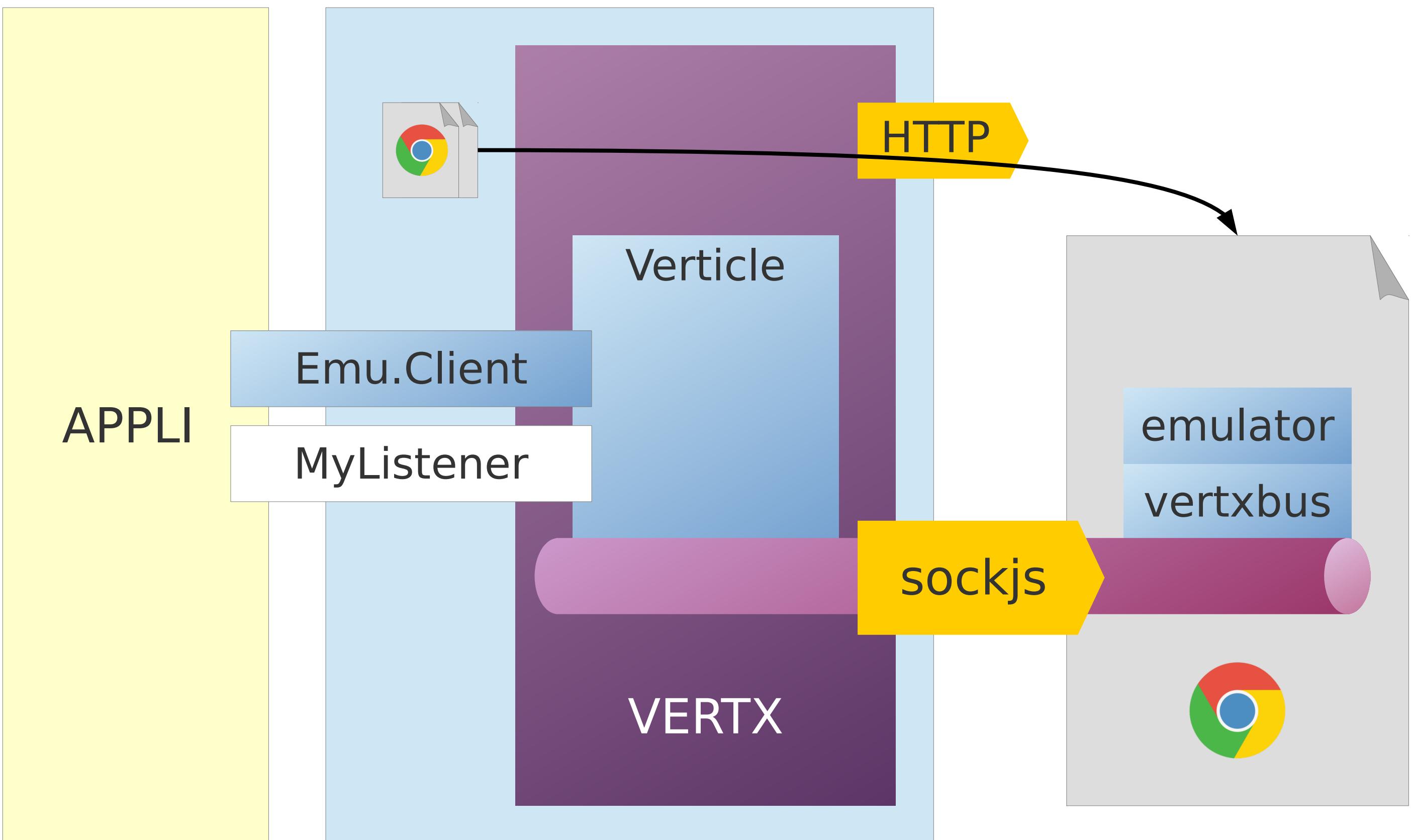


Eventbus bridge

Extension du bus via SockJS
Temps réel (websockets)



LP4J-emu-web



LP4J-emu-web

```
Vertx vertx = VertxFactory.newVertx();

// Static files
HttpServer httpServer = vertx.createHttpServer();
httpServer.requestHandler(new WebResourceHandler());

// Eventbus bridge
vertx.createSockJSServer(httpServer).bridge(...);
vertx.eventBus()
    .registerLocalHandler("lp4j:server", handler);

// Start
httpServer.listen(httpPort);
```

Conception d'API

Une expérience intéressante
Challenge !

Effort

POC : 1 jour

Implémentation midi : 1 semaine

Refactoring : 1 semaine

Implémentation web : 2 semaines

Polissage : 6 mois

L'interface Launchpad

N'existe pas initialement
Canaux in/out indépendants

```
LPCClient client = new LPCClient (midiOut);  
LPReceiver recv = new LPReceiver(midiIn,myListener);
```

Apparition d'une 2° implémentation (émulateur)
Canaux in/out liés (web)
Configuration et initialisation différentes

```
Vertx vertx = ... ;  
LPCClient client = new EmuLPCClient (vertx);  
LPReceiver recv = new EmuLPReceiver(vertx,myListener);
```

L'interface Launchpad

Introduction de l'interface Launchpad

Centralise la configuration

Configure le client et le listener

Inversion de dépendance pour le client

Principe de substitution

// Spécifique

```
Launchpad launchpad = new MidiLaunchpad(midiCfg);  
Launchpad launchpad = new EmulatorLaunchpad(port);
```

// Indépendant

```
LaunchpadClient client = launchpad.getClient();  
Launchpad.setListener(myListener);
```

Gestion des coordonnées

```
void setPadLight(int x, int y, Color color) {  
    checkPadCoords(x,y);  
    ...  
}  
  
void onPadPressed(int x, int y) {  
    checkPadCoords(x,y);  
    ...  
}  
  
void onPadReleased(int x, int y) {  
    checkPadCoords(x,y);  
    ...  
}
```

Gestion des coordonnées

DRY

Classes Pad et Button

Mise en cache des instances

```
public class Pad {  
    private static final Pad[][] PADS = new Pad[8][8];  
    private int x,y;  
    public static Pad at(int x, int y) {  
        if (x<X_MIN || x>X_MAX || y<Y_MIN || y>Y_MAX) {  
            throw new IllegalArgumentException(...);  
        }  
        return PADS[x][y];  
    }  
}  
  
void setPadLight(Pad pad, Color color) { ... }
```

Gestion des exceptions

LaunchpadException

Exception générique top-level

RuntimeException

Wrapping aux interfaces

Conserver l'exception initiale

```
public void reset() {  
    try {  
        midiProtocolClient.reset();  
    } catch (InvalidMidiDataException e) {  
        throw new LaunchpadException(e);  
    }  
}
```

Javadoc

Paramètres

Null / NotNull

Valeurs acceptables

Message d'exception explicatif

@param x The X coordinate of the pad. Must be in range
[{@link Pad#X_MIN},{@link Pad#X_MAX}].

@param color The color to use. Must not be null.

```
if (x < X_MIN || x > X_MAX) {  
    throw new IllegalArgumentException(  
        "Illegal X coordinate : " + x +  
        ".Acceptable values are in ["+X_MIN+".." +X_MAX+"].");  
}
```

Javadoc

Doclets

{@link <symbole>} : lien vers un symbole

{@inheritDoc} : recopie la javadoc parente

```
/**  
 * Parses low-level messages and notifies a high-level  
 * {@link net.thecodersbreakfast.lp4j.api.LaunchpadListener}.  
 */  
  
/**  
 * {@inheritDoc}  
 * @param brightness {@inheritDoc} Must not be null.  
 */  
public void setBrightness(Brightness brightness) { ... }
```

Licence

Obligatoire Sur tous les fichiers Maven-rat-plugin

```
/*
 * Copyright 2015 Olivier Croisier (thecodersbreakfast.net)
 *
 * Licensed under the Apache License, Version 2.0 (the "License");
 * you may not use this file except in compliance with the License.
 * You may obtain a copy of the License at
 *
 *     http://www.apache.org/licenses/LICENSE-2.0
 *
 * Unless required by applicable law or agreed to in writing, software
 * distributed under the License is distributed on an "AS IS" BASIS,
 * WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
 * See the License for the specific language governing permissions and
 * limitations under the License.
 */
```

Applications

Le Launchpad propose

Input : 64 touches + 16 boutons

Output : 8x8 LEDs colorées

Domaines d'application

MAO

Jeux (mineweeper, simon)

Aide à la saisie (keypad, authentification)

Affichage / supervision

...

Hackathon ?

LP4J

A song of Java, MIDI
and websockets

<http://github.com/OlivierCroisier/LP4J>

<http://github.com/OlivierCroisier/LP4J-examples>



TheCodersBreakfast.net



@OlivierCroisier



olivier.croisier@mokatech.net



github.com/OlivierCroisier

Crédits images

<http://kontroleryzm.pl>

<http://www.wmnstudio.com>

<http://consequenceofsound.wordpress.com>