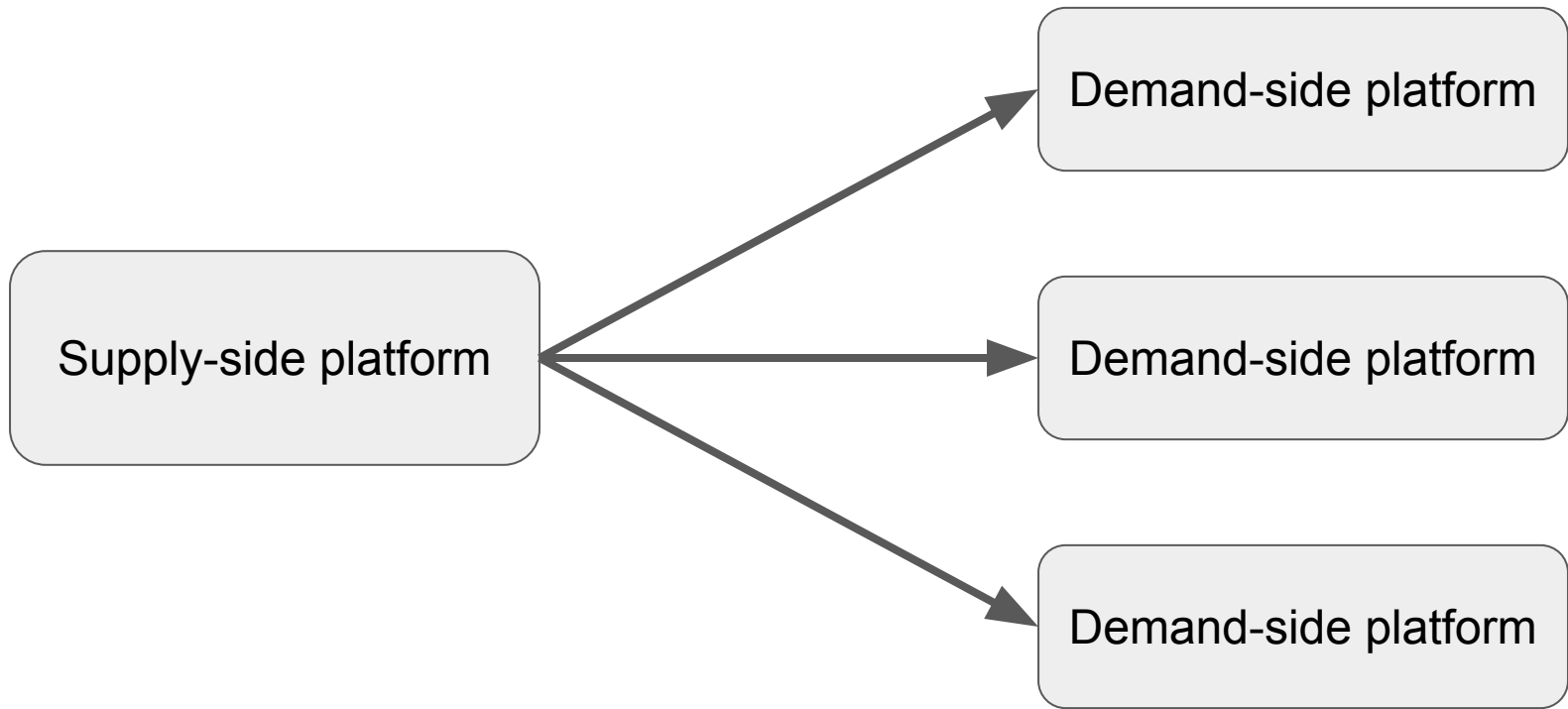
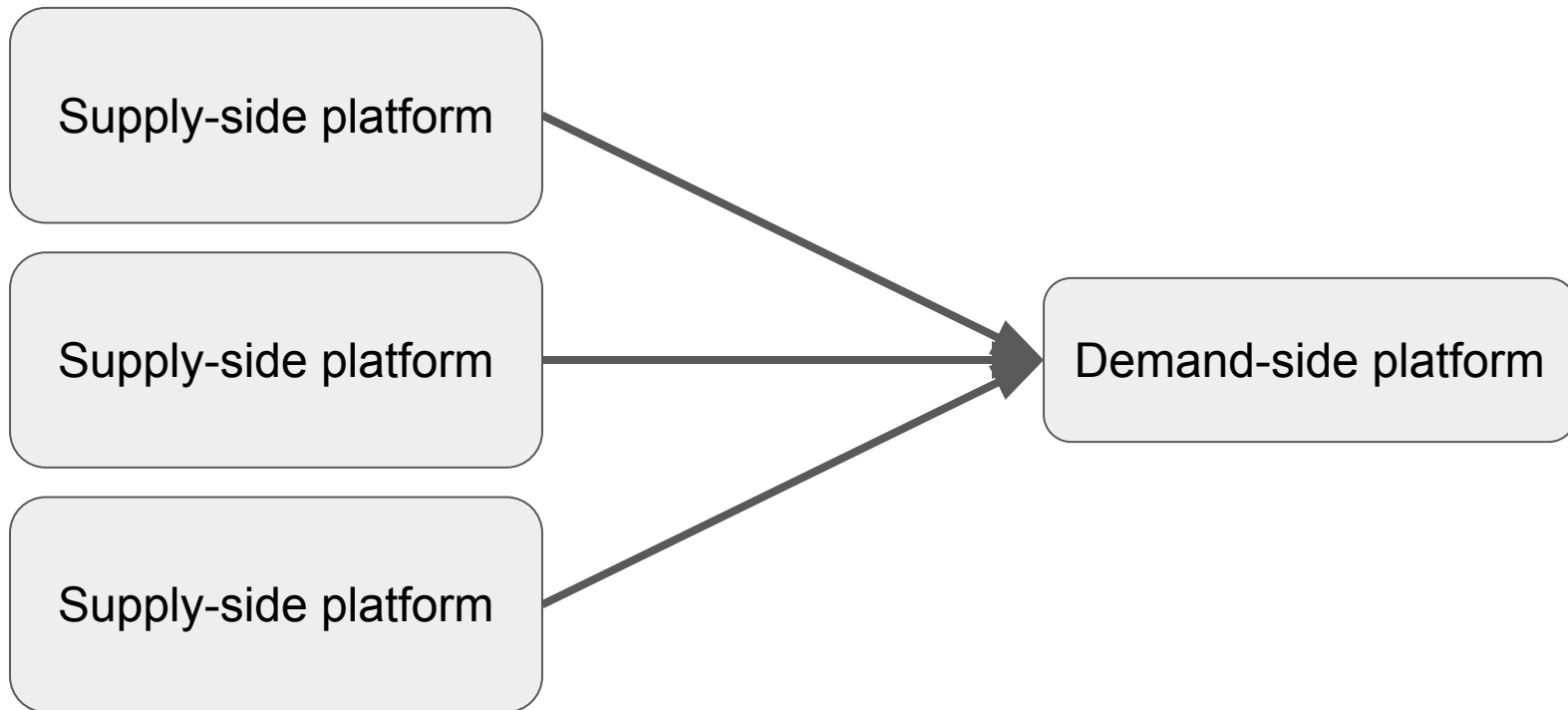
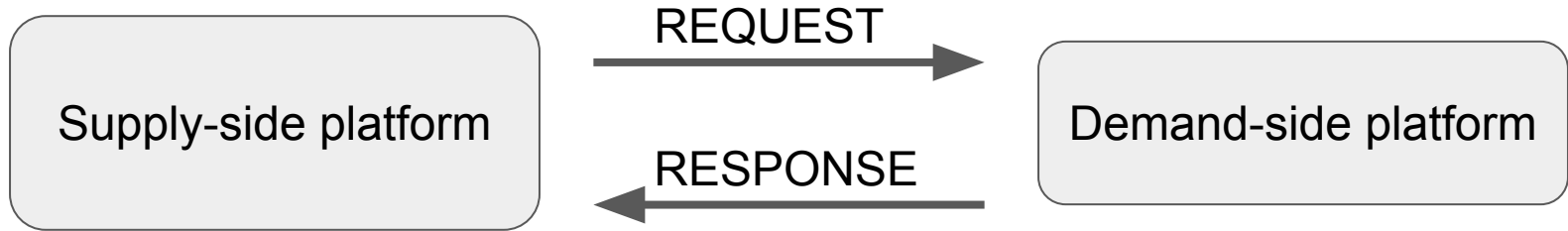


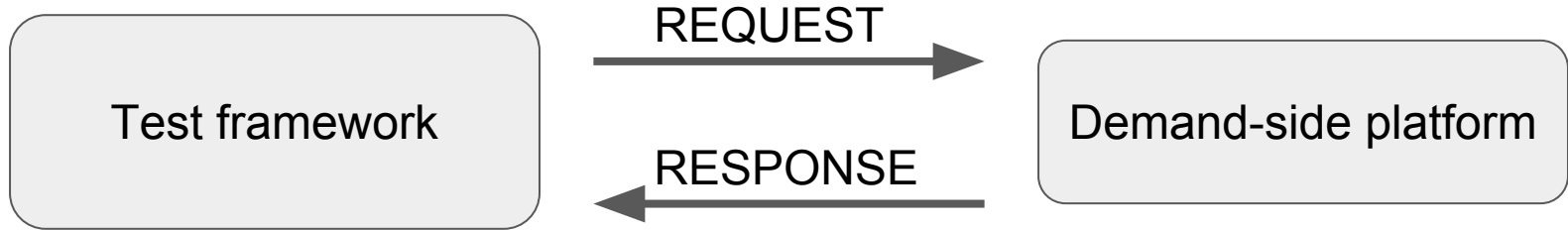
INTEGRATION TEST FRAMEWORK FOR G4 DSP

Ilya Medvedev









Test

1. Create request
2. Send request from AppNexus
3. Test response

```
cerateRequest() {  
    request = new Request();  
}
```

```
sendRequest(String sspName) {  
    requester = getRequester(sspName);  
  
    response = requester.send(request);  
}
```

```
testResponse() {  
    If (response.isNoBid()) {  
        throw new NoBidResponseException(response);  
    }  
}
```

Test

1. Create request with American IP address
2. Send request from AppNexus
3. Test response


```
cerateRequest() {  
    request = new Request();  
}
```

```
sendRequest(String sspName) {  
    requester = getRequester(sspName);  
  
    response = requester.send(request);  
}
```

```
testResponse() {  
    If (response.isNoBid()) {  
        throw new NoBidResponseException(response);  
    }  
}
```

```
cerateRequest() {  
    request = new Request();  
}
```

```
cerateRequestWithIpAddress(String country) {  
    request = new Request()  
        .setIP(geolp.getIp(country));  
}
```

```
sendRequest(String sspName) {  
    requester = getRequester(sspName);  
  
    response = requester.send(request);  
}
```

```
testResponse() {  
    If (response.isNoBid()) {  
        throw new NoBidResponseException(response);  
    }  
}
```

Test

1. Create request with Verizon mobile carrier
2. Send request from AppNexus
3. Test response

```
cerateRequest() {  
    request = new Request();  
}
```

```
cerateRequestWithIpAddress(String country) {  
    request = new Request()  
        .setIP(geolp.getIp(country));  
}
```

```
sendRequest(String sspName) {  
    requester = getRequester(sspName);  
  
    response = requester.send(request);  
}
```

```
testResponse() {  
    If (response.isNoBid()) {  
        throw new NoBidResponseException(response);  
    }  
}
```

```
cerateRequest() {  
    request = new Request();  
}  
  
cerateRequestWithIpAddress(String country) {  
    request = new Request()  
        .setIP(geolp.getIp(country));  
}  
  
cerateRequestWithMobileCarrier(String carrier)) {  
    request = new Request()  
        .setMobileCarrier(carrier);  
}  
  
sendRequest(String sspName) {  
    requester = getRequester(sspName);  
  
    response = requester.send(request);  
}  
  
testResponse() {  
    If (response.isNoBid()) {  
        throw new NoBidResponseException(response);  
    }  
}
```

```
cerateRequest() {  
    request = new Request();  
}  
  
sendRequest(String sspName) {  
    requester = getRequester  
(sspName);  
  
    response = requester.send  
(request);  
}  
  
testResponse() {  
    If (response.isNoBid()) {  
        throw new  
NoBidResponseException(response);  
    }  
}
```

```
cerateRequestWithIpAddress(String  
country) {  
    request = new Request()  
        .setIP(geolp.getIp(country));  
}  
  
sendRequest(String sspName) {  
    requester = getRequester  
(sspName);  
  
    response = requester.send  
(request);  
}  
  
testResponse() {  
    If (response.isNoBid()) {  
        throw new  
NoBidResponseException(response);  
    }  
}
```

```
cerateRequestWithMobileCarrier(String  
carrier)) {  
    request = new Request()  
        .setMobileCarrier(carrier);  
}  
  
sendRequest(String sspName) {  
    requester = getRequester  
(sspName);  
  
    response = requester.send  
(request);  
}  
  
testResponse() {  
    If (response.isNoBid()) {  
        throw new  
NoBidResponseException(response);  
    }  
}
```

Test

1. Create request
2. Send request from AppNexus
3. Test response

```
class RequestCreator {  
    Context context;  
  
    RequestCreator(Context context) {  
        this.context = context;  
    }  
}
```


<dependency>

<groupId>info.cukes**</groupId>**

<artifactId>cucumber-picocontainer**</artifactId>**

<version>\${dependency.cucumber.version}**</version>**

</dependency>

```
class RequestCreator {  
    Context context;  
  
    RequestCreator(Context context) {  
        this.context = context;  
    }  
}
```

```
cerateRequest() {  
    request = new Request();  
  
    context.setValue(Context.REQUEST, request);  
}
```

```
cerateRequestWithIpAddress(String country) {  
    request = new Request()  
        .setIP(geolp.getIp(country));  
  
    context.setValue(Context.REQUEST, request);  
}
```

```
cerateRequestWithMobileCarrier(String carrier) {  
    request = new Request()  
        .setMobileCarrier(carrier);  
  
    context.setValue(Context.REQUEST, request);  
}
```

```
class RequestSender {  
    Context context;  
  
    RequestSender(Context context) {  
        this.context = context;  
    }  
}
```

```
sendRequest(String sspName) {  
    request = context.getValue(Context.REQUEST);  
    requester = getRequester(sspName);  
    response = requester.send(request);  
  
    context.setValue(Context.RESPONSE, response);  
}
```

```
class RequestTester {  
    Context context;  
  
    RequestTester(Context context) {  
        this.context = context;  
    }  
}
```

```
testResponse() {  
    response = context.getValue(Context.RESPONSE);  
  
    If (response.isNoBid()) {  
        throw new NoBidResponseException(response);  
    }  
}
```

```
public class Context {  
    public static final Context<Request> REQUEST = new Context<>();  
    public static final Context<Response> RESPONSE = new Context<>();  
  
    private final Map<Context, Object> context = new HashMap<>();  
  
    public <Value> Context setValue(final Context<Value> context, Value value) {  
        context.put(context, value);  
  
        return this;  
    }  
  
    public <Value> Value getValue(final Context<Value> context) {  
        return (Value) context.get(context)  
    }  
}
```



```
cerateRequest() {  
    request = new Request();  
  
    context.setValue(Context.REQUEST, request);  
}
```

```
cerateRequestWithIpAddress(String country) {  
    request = new Request()  
        .setIP(geolp.getIp(country));  
  
    context.setValue(Context.REQUEST, request);  
}
```

```
cerateRequestWithMobileCarrier(String carrier) {  
    request = new Request()  
        .setMobileCarrier(carrier);  
  
    context.setValue(Context.REQUEST, request);  
}
```

```
sendRequest(String sspName) {  
    request = context.getValue(Context.REQUEST);  
    requester = getRequester(sspName);  
    response = requester.send(request);  
  
    context.setValue(Context.RESPONSE, response);  
}
```

```
testResponse() {  
    response = context.getValue(Context.RESPONSE);  
  
    If (response.isNoBid()) {  
        throw new NoBidResponseException(response);  
    }  
}
```

```
public class SharedBrowser {  
    private static final Browser BROWSER = new Browser();  
  
    static {  
        final ShutdownHook shutdownHook = new ShutdownHook(BROWSER);  
  
        Runtime.getRuntime()  
            .addShutdownHook(shutdownHook);  
    }  
}
```

```
public class ShutdownHook extends Thread {  
    private final Closeable closeable;  
  
    ShutdownHook(final Closeable closeable) {  
        this.closeable = closeable;  
    }  
  
    void run() {  
        closeable.close();  
    }  
}
```

Questions?