Erwin de Gier

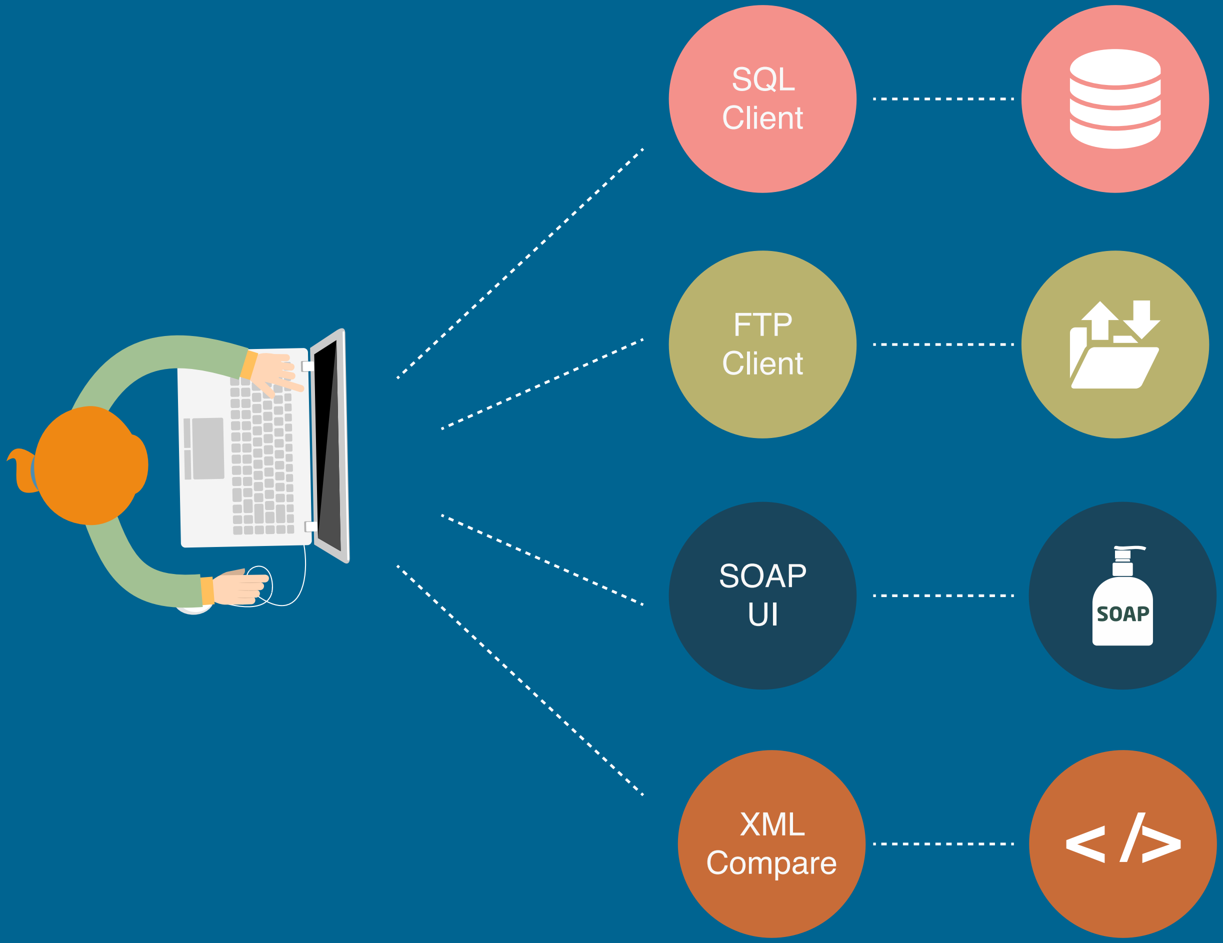
Creating a polyglot test framework with reactive technology

github.com/erwindeg

@erwindeg

edegier.nl

SOGETI

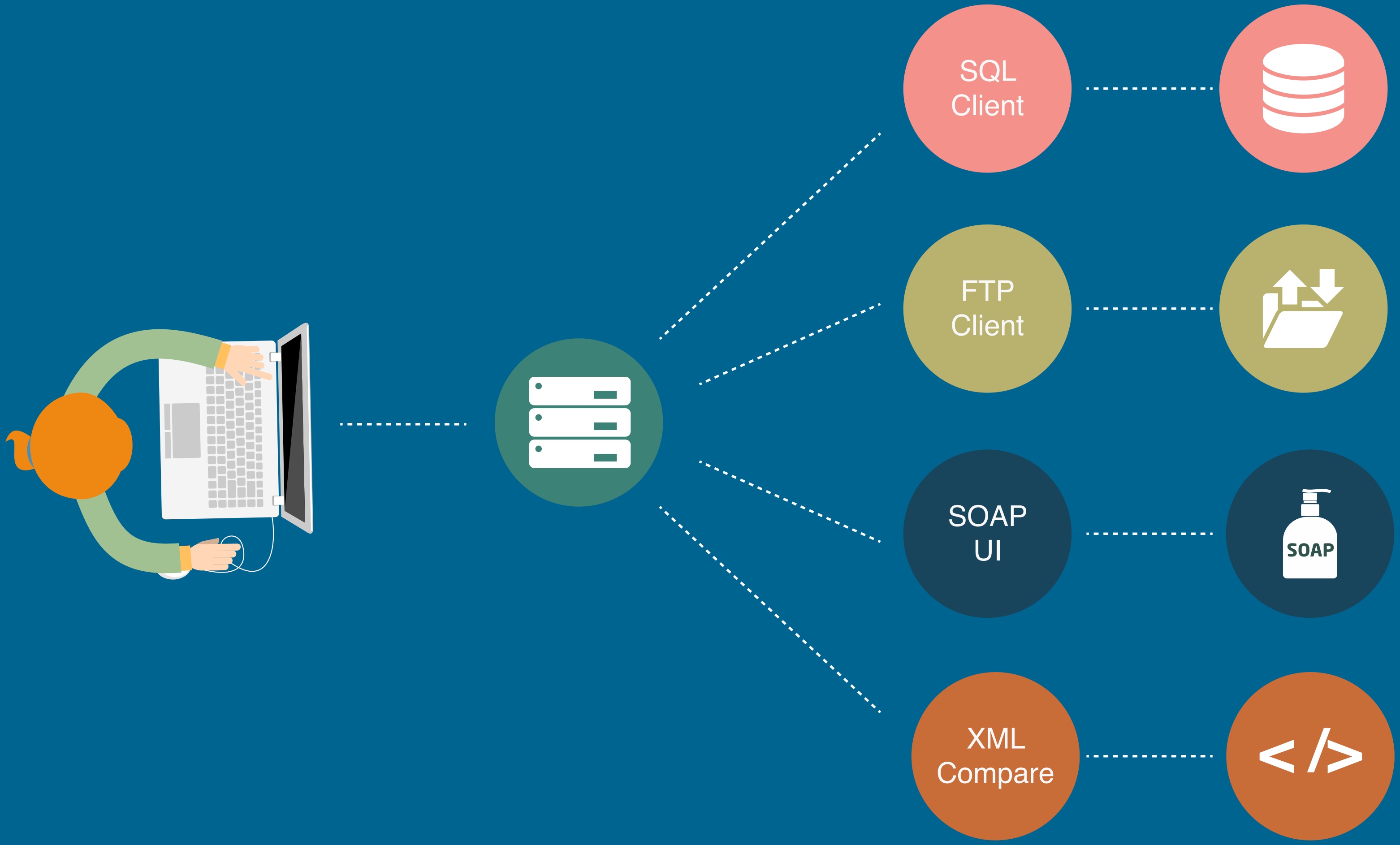


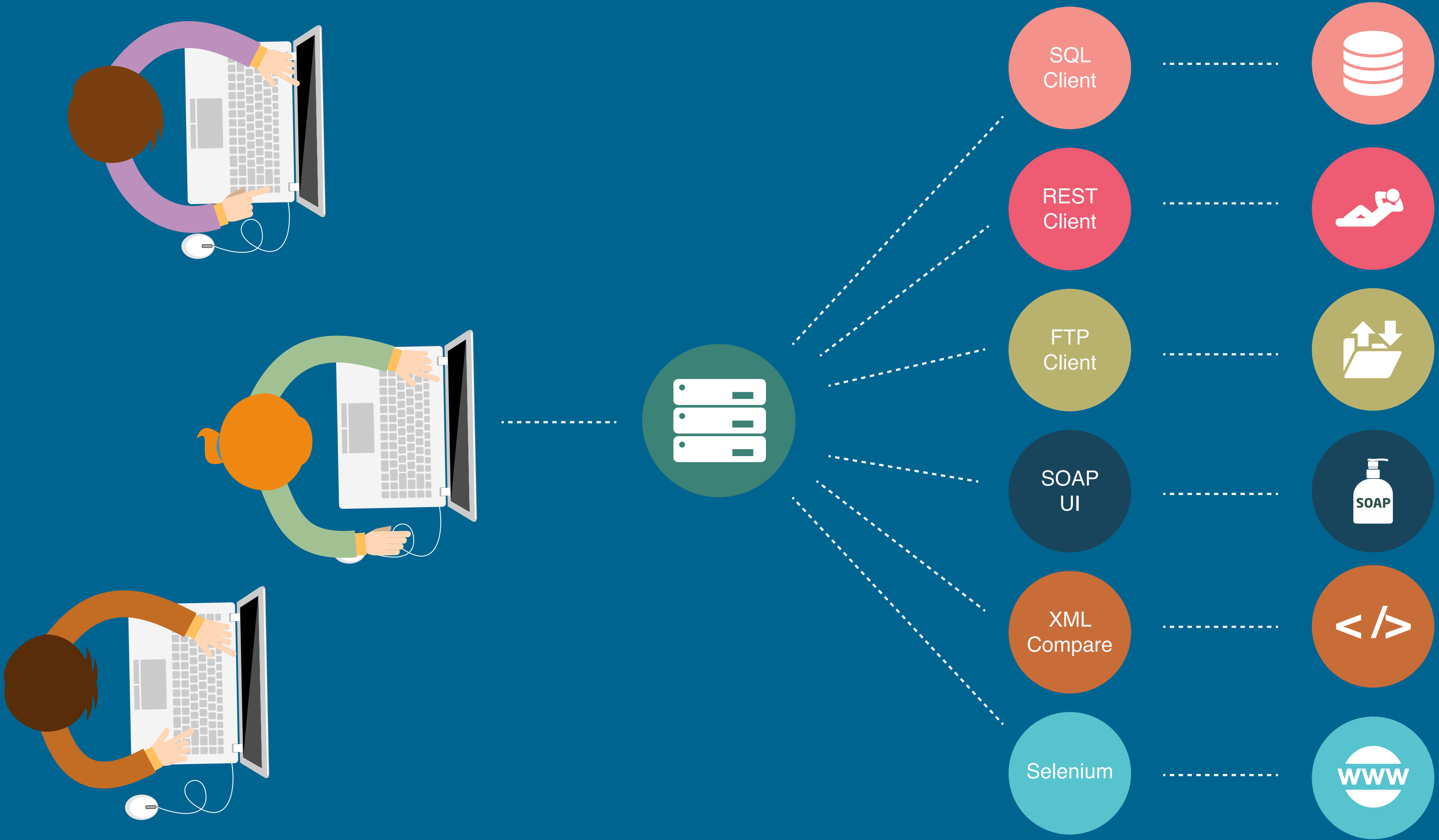
SQL Client

FTP Client

SOAP UI

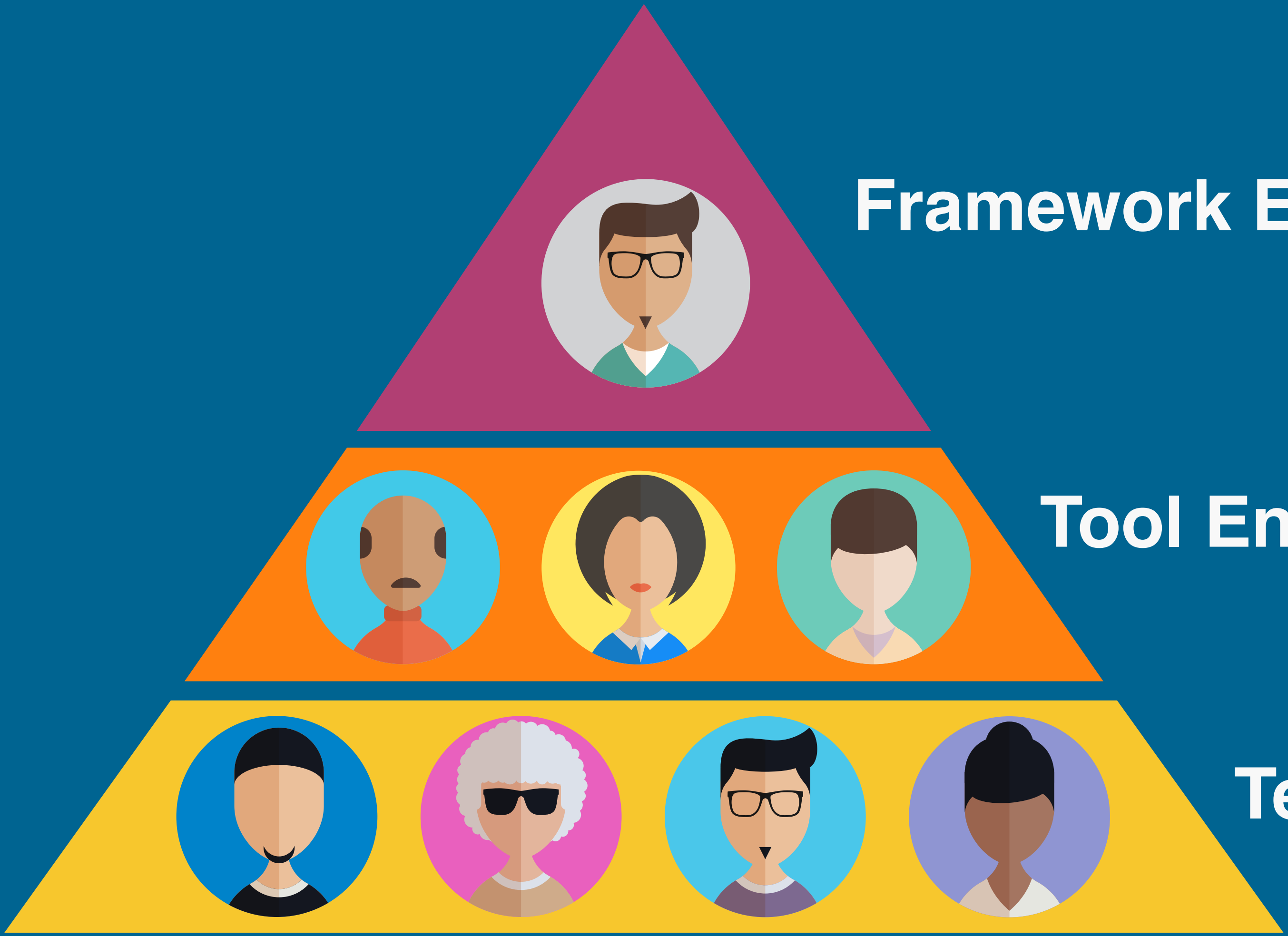
XML Compare


SQL Client
FTP Client
SOAP UI
SOAP
XML Compare
</>


SQL
Client
REST
Client
FTP
Client
SOAP
UI
SOAP
XML
Compare
</>
Selenium
WWW

Requirements

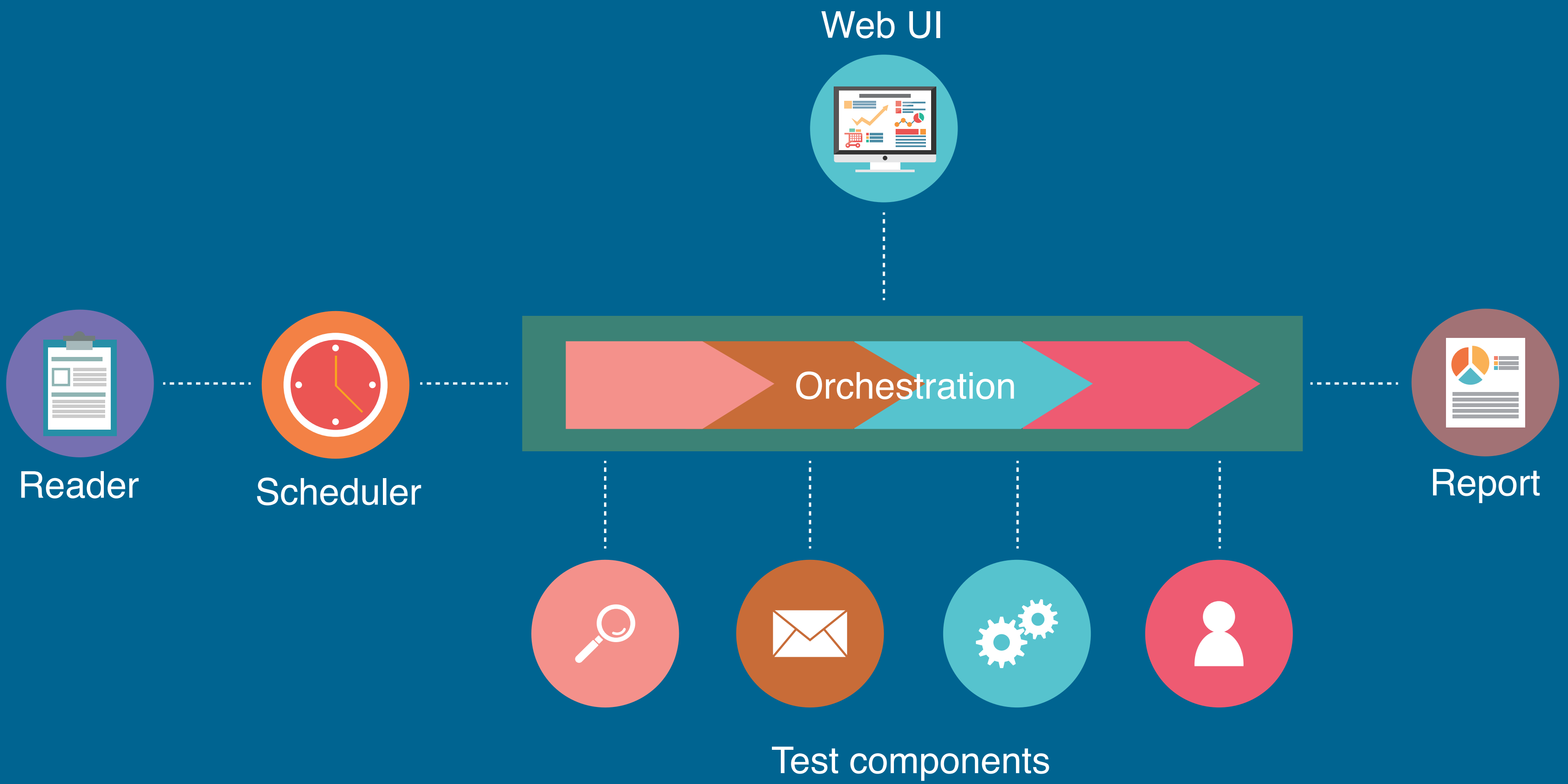
- ✓ Integrate existing test libraries
- ✓ Scripting language support
- ✓ Extensive logging
- ✓ Scalable over multiple servers



Framework Engineers

Tool Engineers

Test automation



Web UI
Reader
Scheduler
Orchestration
Report
Test components

Demo

VERT.X

ANGULARJS

mongoDB

VERT.X

Reactive

Non-blocking

Message driven

Components

Polyglot

Runnable jar

Distributed eventbus

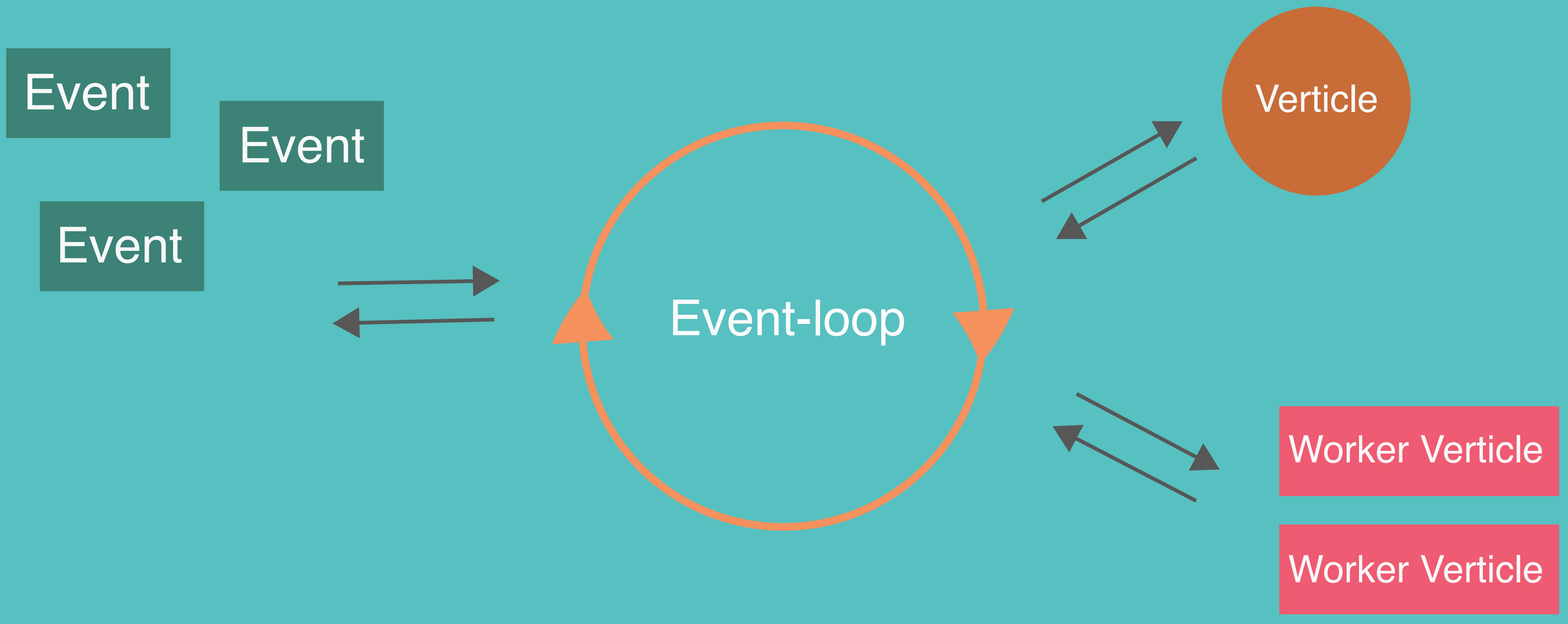
Service Registry

Fun

Metrics

Distributed storage

Full stack



Non blocking single-threaded

Event

Event

Event

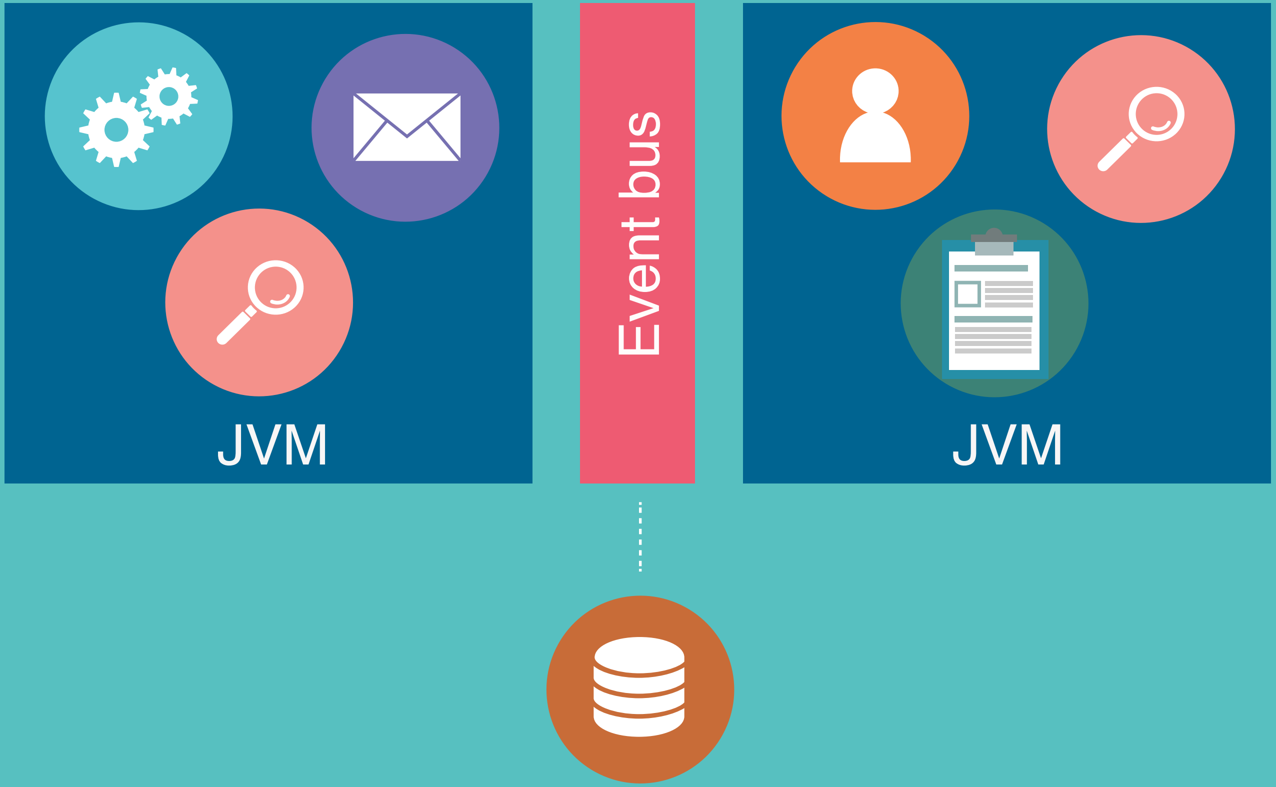
Event-loop

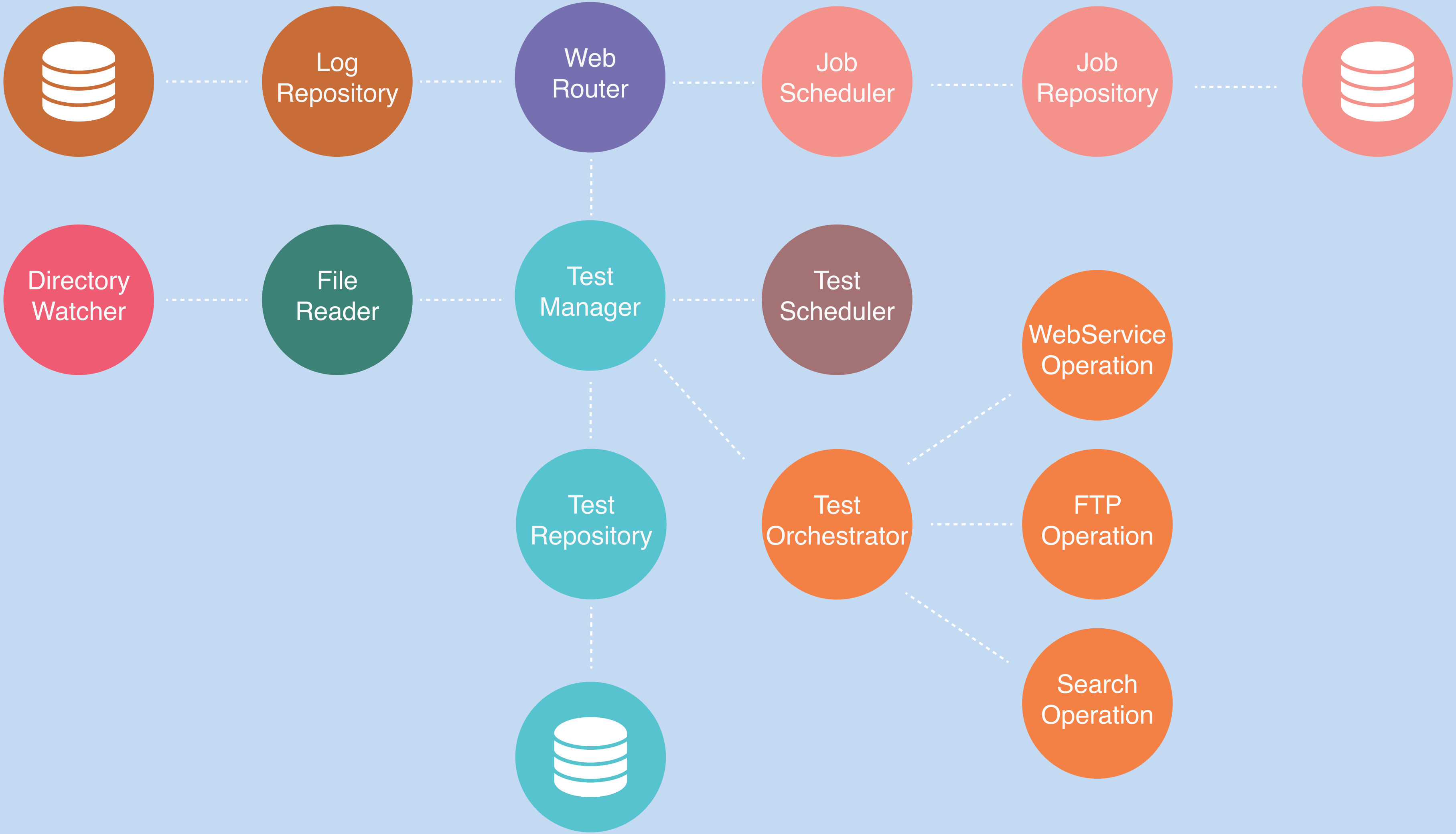
Verticle

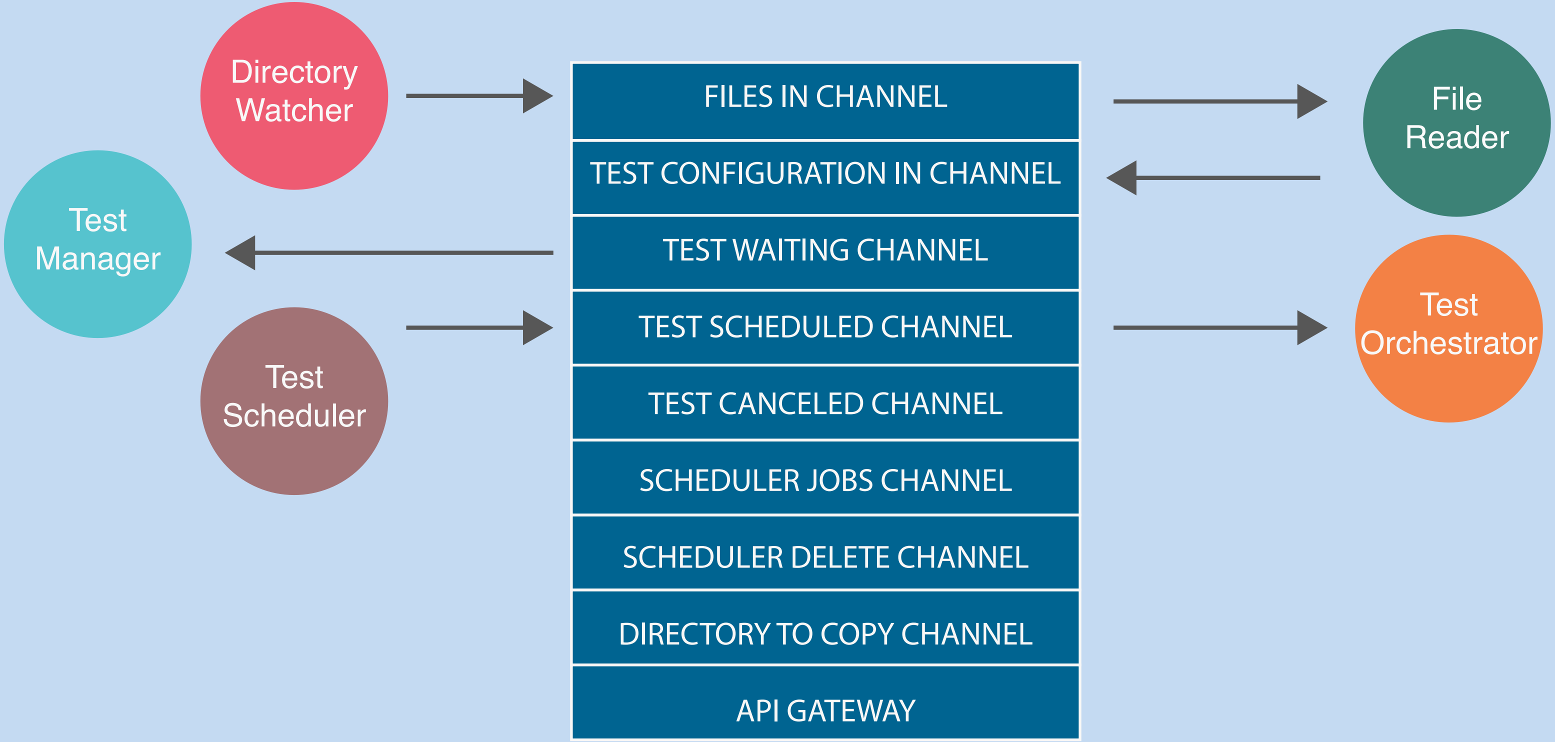
Worker Verticle

Worker Verticle

Blocking multi-threaded


JVM
Event bus
JVM


Log Repository
Web Router
Job Scheduler
Job Repository
Directory Watcher
File Reader
Test Manager
Test Scheduler
WebService Operation
Test Repository
Test Orchestrator
FTP Operation
Search Operation



Directory Watcher

Test Manager

Test Scheduler

FILES IN CHANNEL

TEST CONFIGURATION IN CHANNEL

TEST WAITING CHANNEL

TEST SCHEDULED CHANNEL

TEST CANCELED CHANNEL

SCHEDULER JOBS CHANNEL

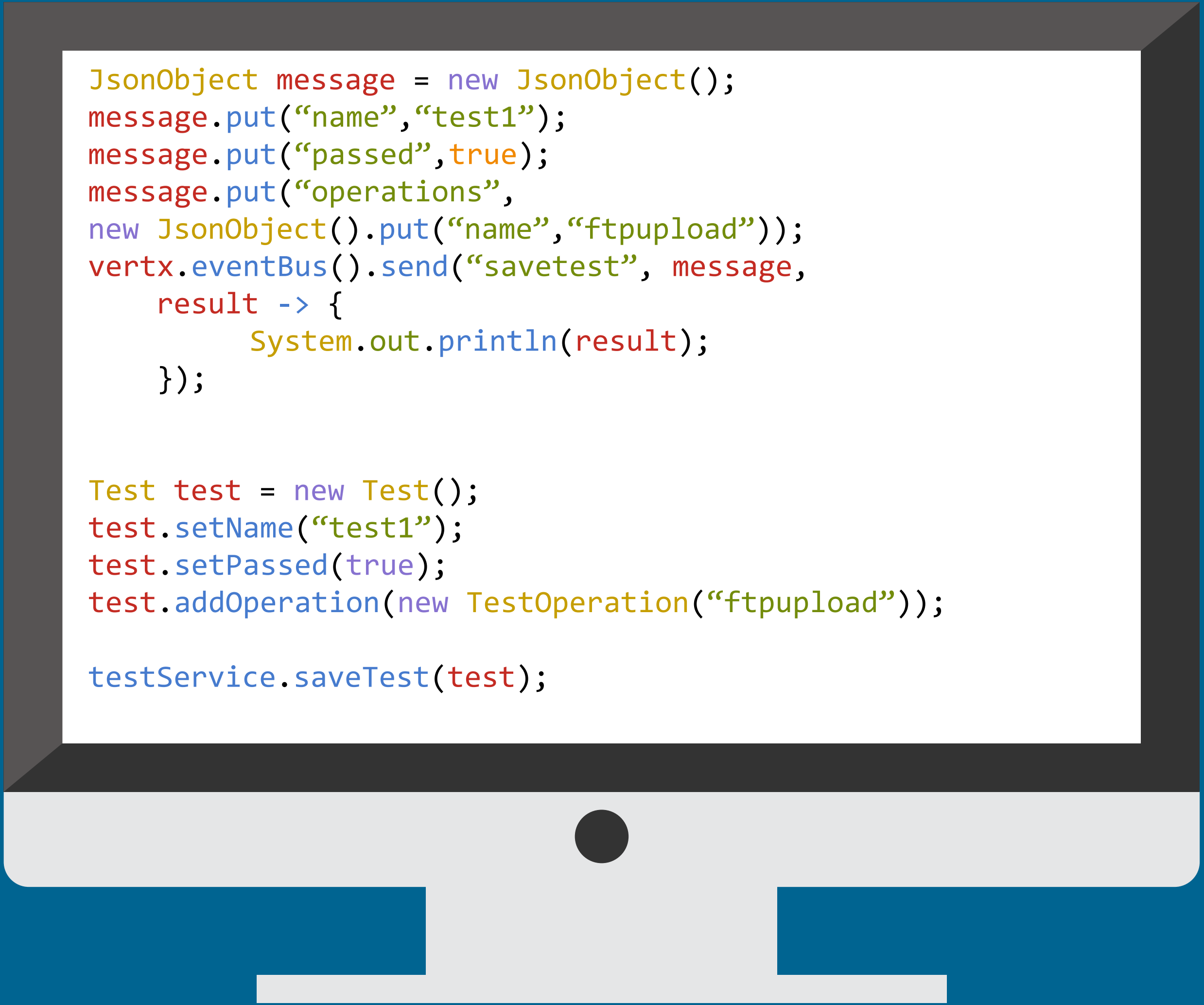
SCHEDULER DELETE CHANNEL

DIRECTORY TO COPY CHANNEL

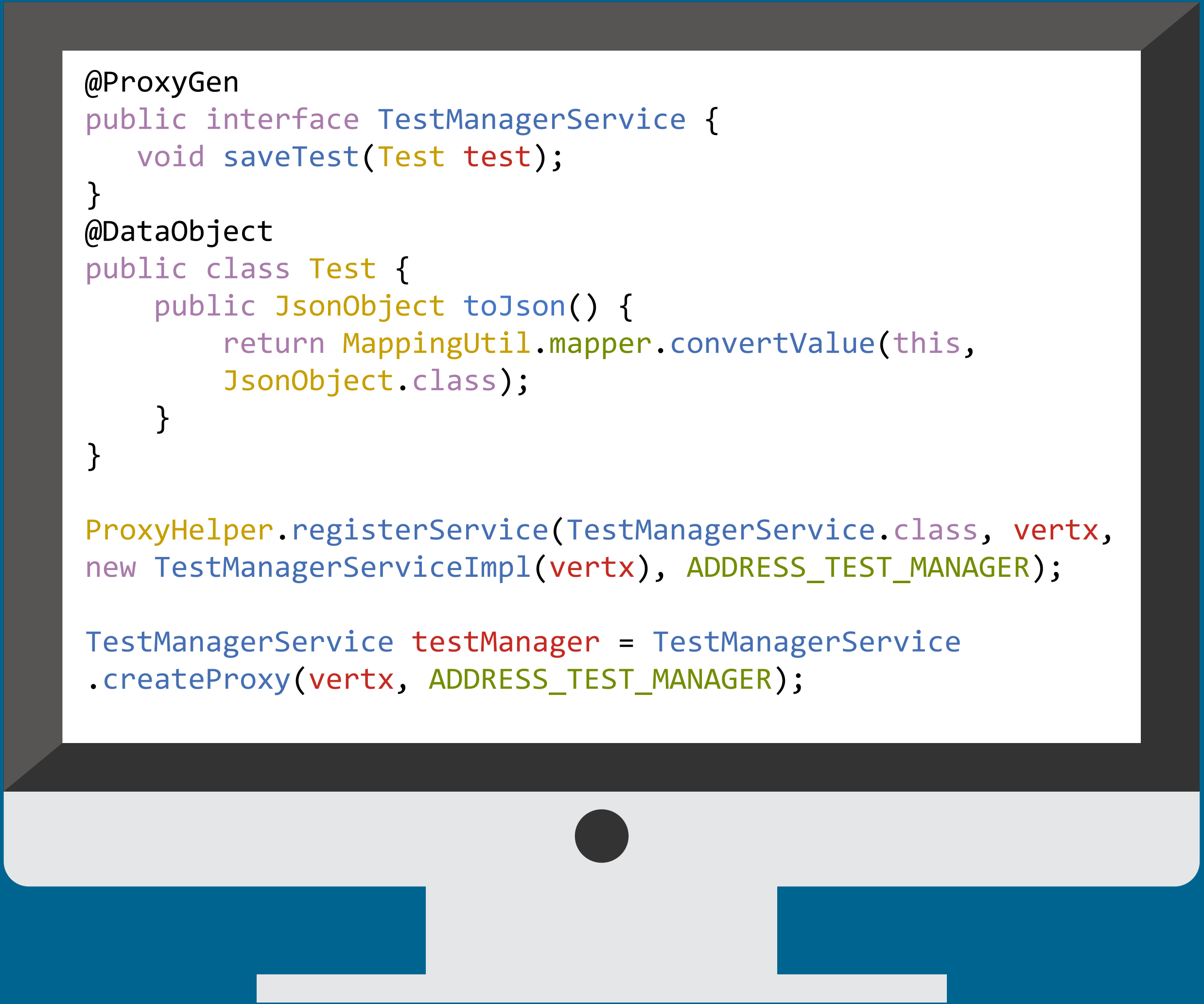
API GATEWAY

File Reader

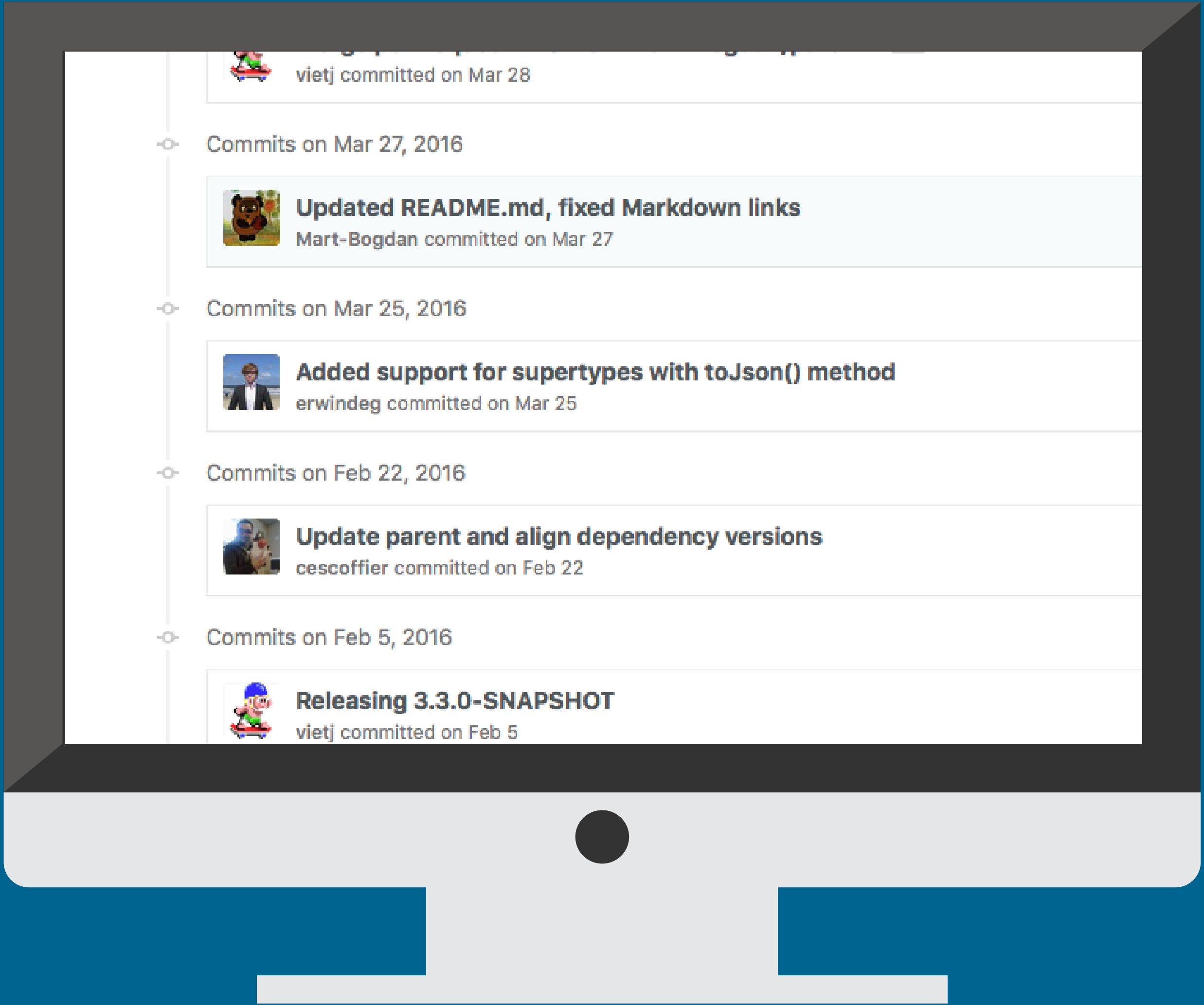
Test Orchestrator



```
JsonObject message = new JsonObject();
message.put(“name”,“test1”);
message.put(“passed”,true);
message.put(“operations”,
new JsonObject().put(“name”,“ftpupload”));
vertx.eventBus().send(“savetest”, message,
    result -> {
          System.out.println(result);
    });


Test test = new Test();
test.setName(“test1”);
test.setPassed(true);
test.addOperation(new TestOperation(“ftpupload”));

testService.saveTest(test);
```



```
@ProxyGen
public interface TestManagerService {
   void saveTest(Test test);
}
@DataObject
public class Test {
    public JsonObject toJson() {
        return MappingUtil.mapper.convertValue(this,
        JsonObject.class);
    }
}

ProxyHelper.registerService(TestManagerService.class, vertx,
new TestManagerServiceImpl(vertx), ADDRESS_TEST_MANAGER);

TestManagerService testManager = TestManagerService
.createProxy(vertx, ADDRESS_TEST_MANAGER);
```



vietj committed on Mar 28

Commits on Mar 27, 2016

Updated README.md, fixed Markdown links
Mart-Bogdan committed on Mar 27

Commits on Mar 25, 2016

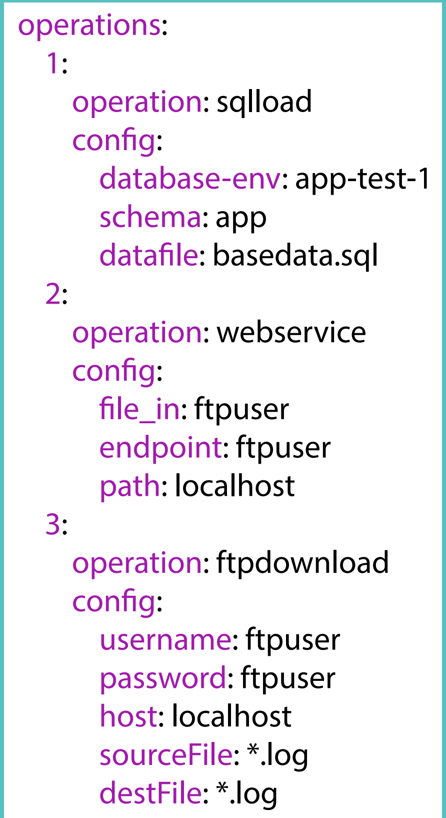
Added support for supertypes with toJson() method
erwindeg committed on Mar 25

Commits on Feb 22, 2016

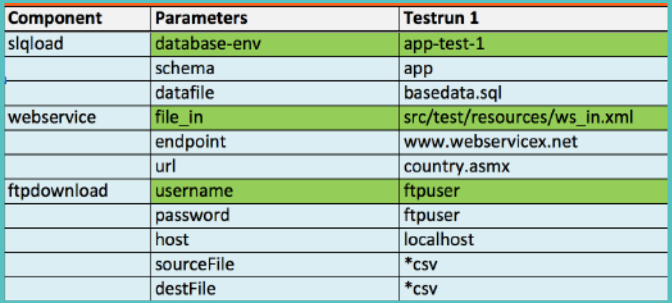
Update parent and align dependency versions
cescoffier committed on Feb 22

Commits on Feb 5, 2016

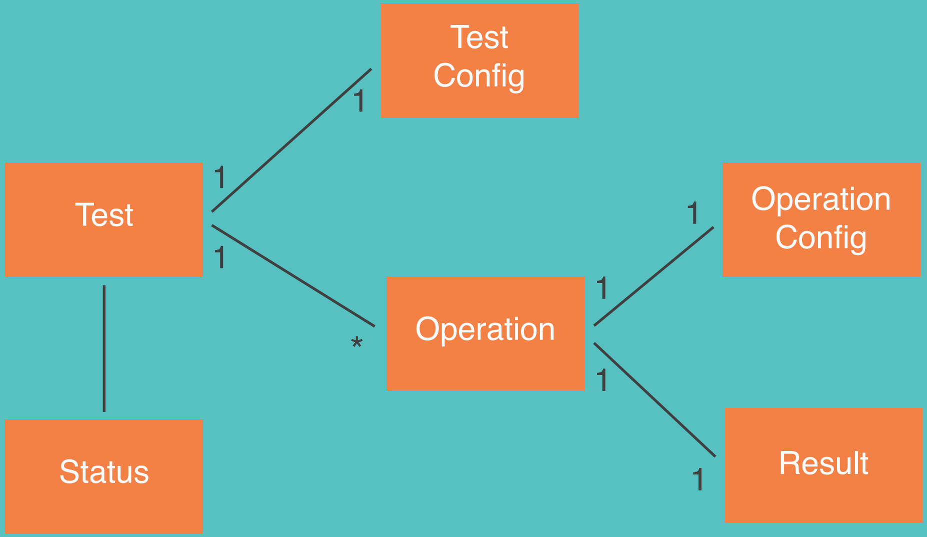
Releasing 3.3.0-SNAPSHOT
vietj committed on Feb 5

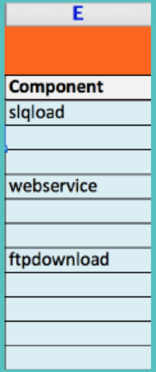


```
operations:
  1:
    operation: sqlload
    config:
      database-env: app-test-1
      schema: app
      datafile: basedata.sql
  2:
    operation: webservice
    config:
      file_in: ftpuser
      endpoint: ftpuser
      path: localhost
  3:
    operation: ftpdownload
    config:
      username: ftpuser
      password: ftpuser
      host: localhost
      sourceFile: *.log
      destFile: *.log
```

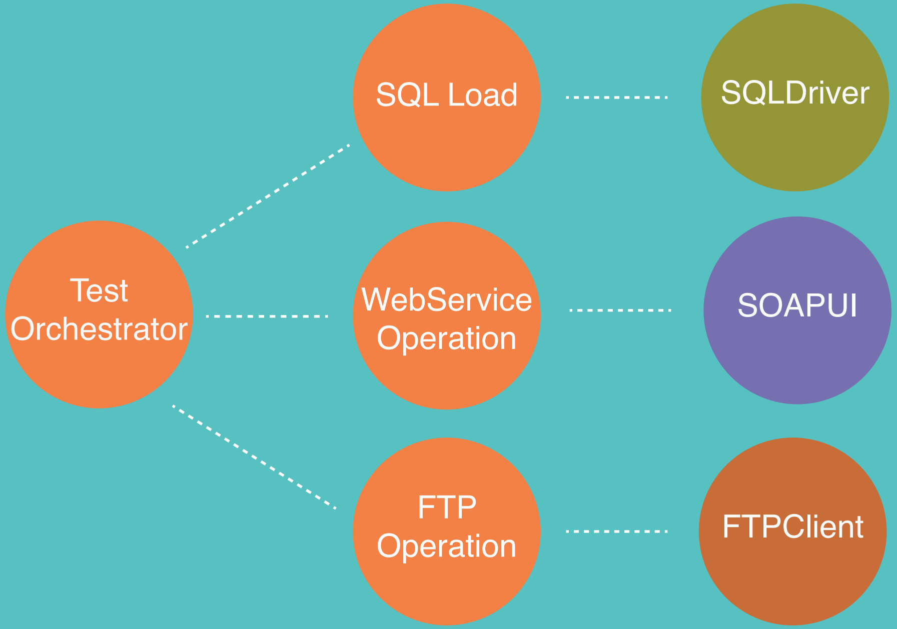


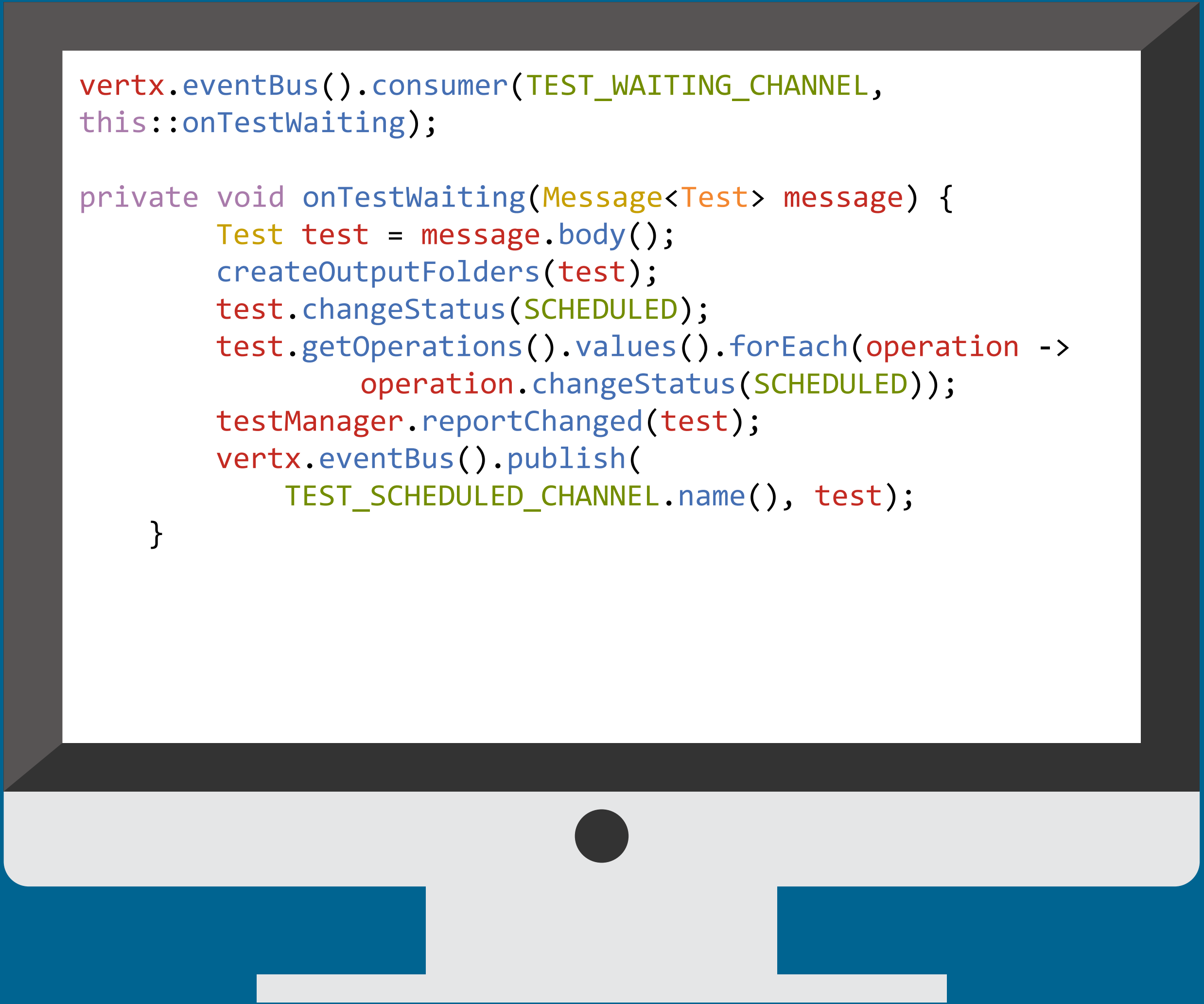
Component	Parameters	Testrun 1
slqload	database-env	app-test-1
	schema	app
	datafile	basedata.sql
webservice	file_in	src/test/resources/ws_in.xml
	endpoint	www.webservicex.net
	url	country.asmx
ftpdownload	username	ftpuser
	password	ftpuser
	host	localhost
	sourceFile	*csv
	destFile	*csv



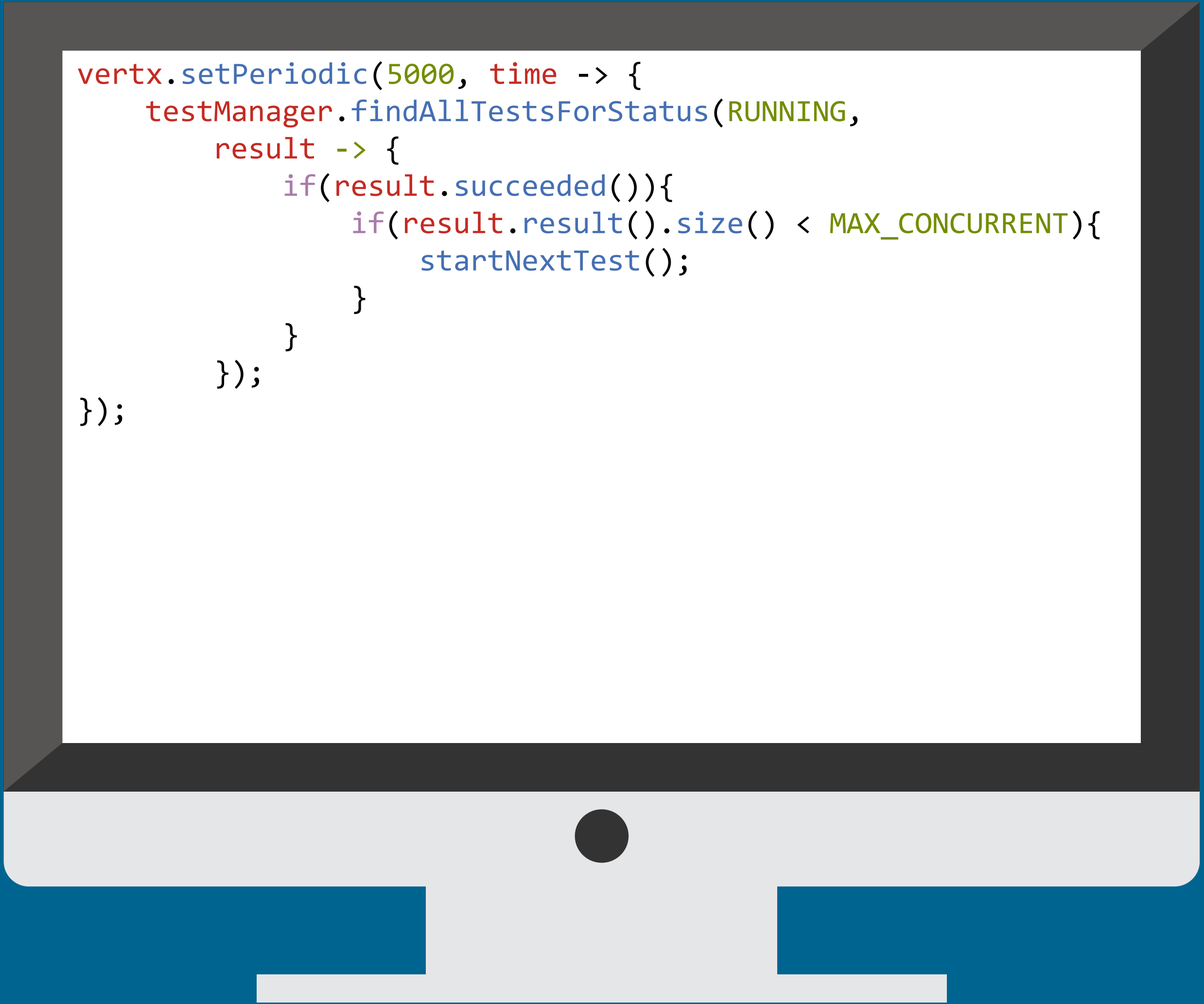


E
Component
slqload
webservice
ftpdownload

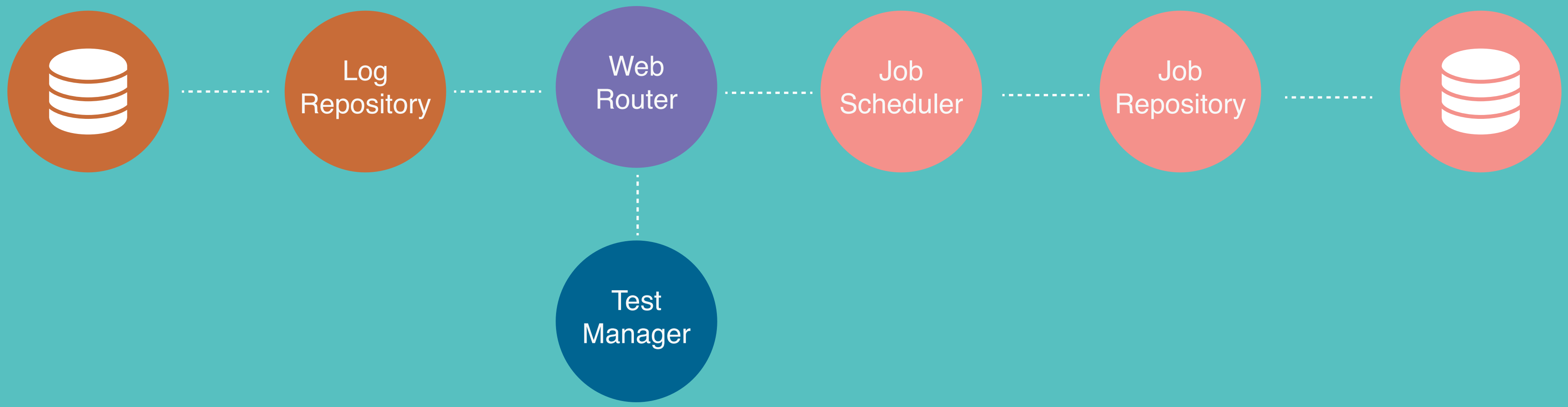


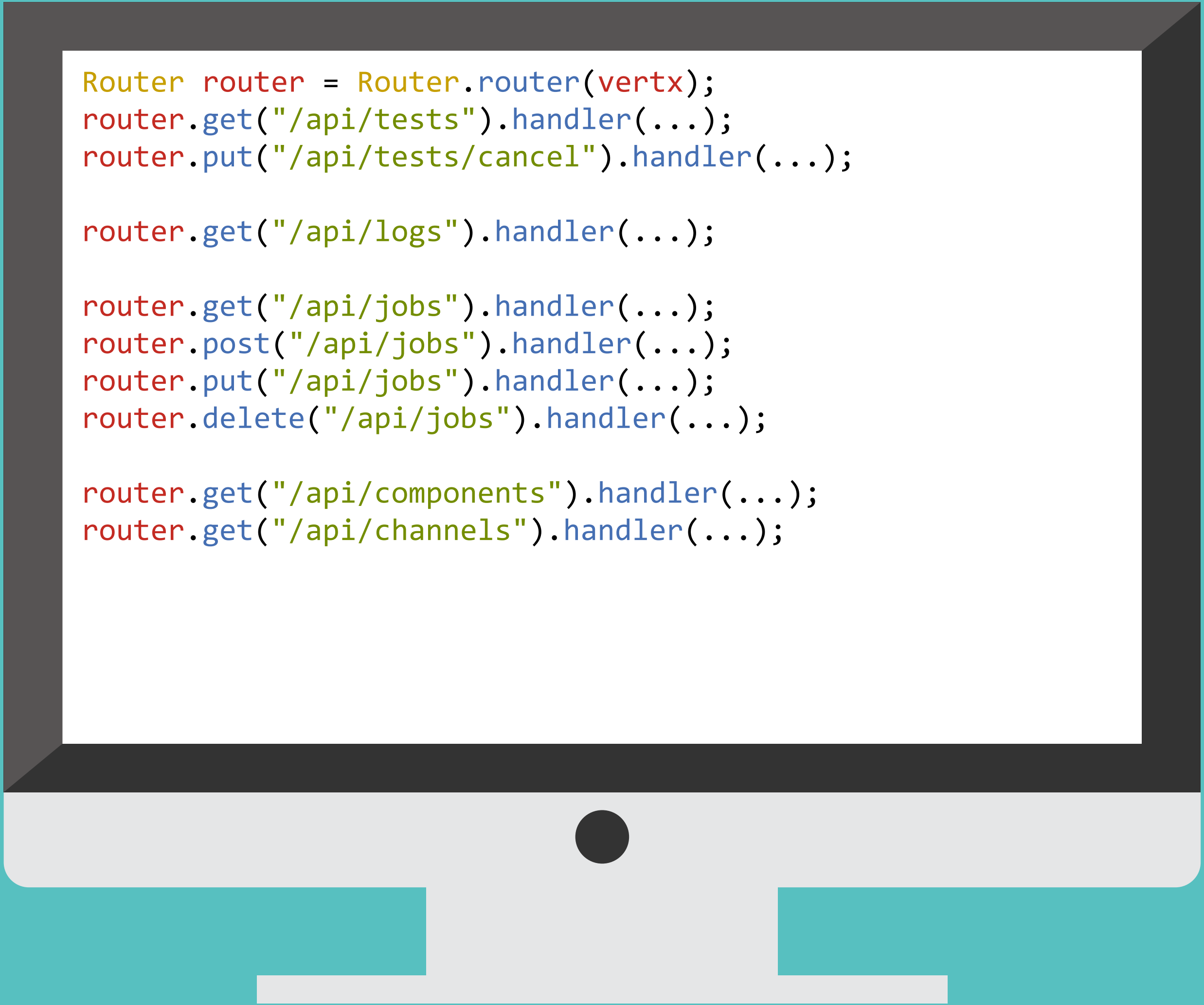


```
vertx.eventBus().consumer(TEST_WAITING_CHANNEL,
this::onTestWaiting);

private void onTestWaiting(Message<Test> message) {
        Test test = message.body();
        createOutputFolders(test);
        test.changeStatus(SCHEDULED);
        test.getOperations().values().forEach(operation ->
                operation.changeStatus(SCHEDULED));
        testManager.reportChanged(test);
        vertx.eventBus().publish(
            TEST_SCHEDULED_CHANNEL.name(), test);
    }
```

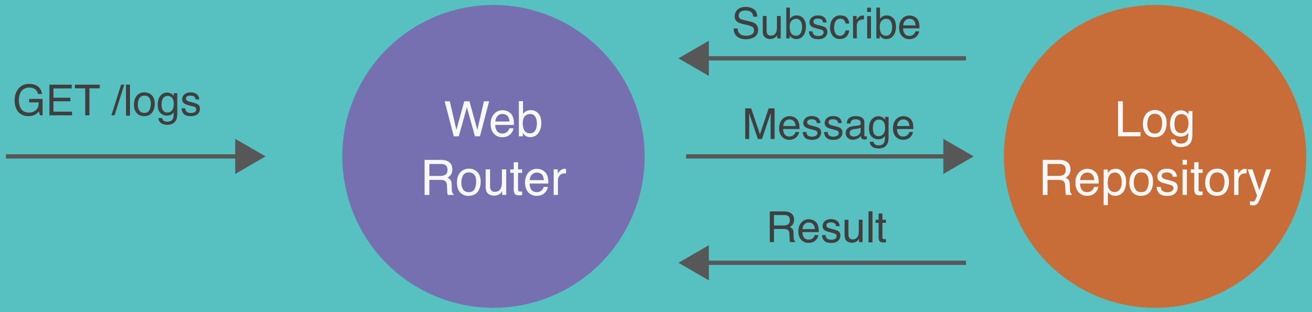


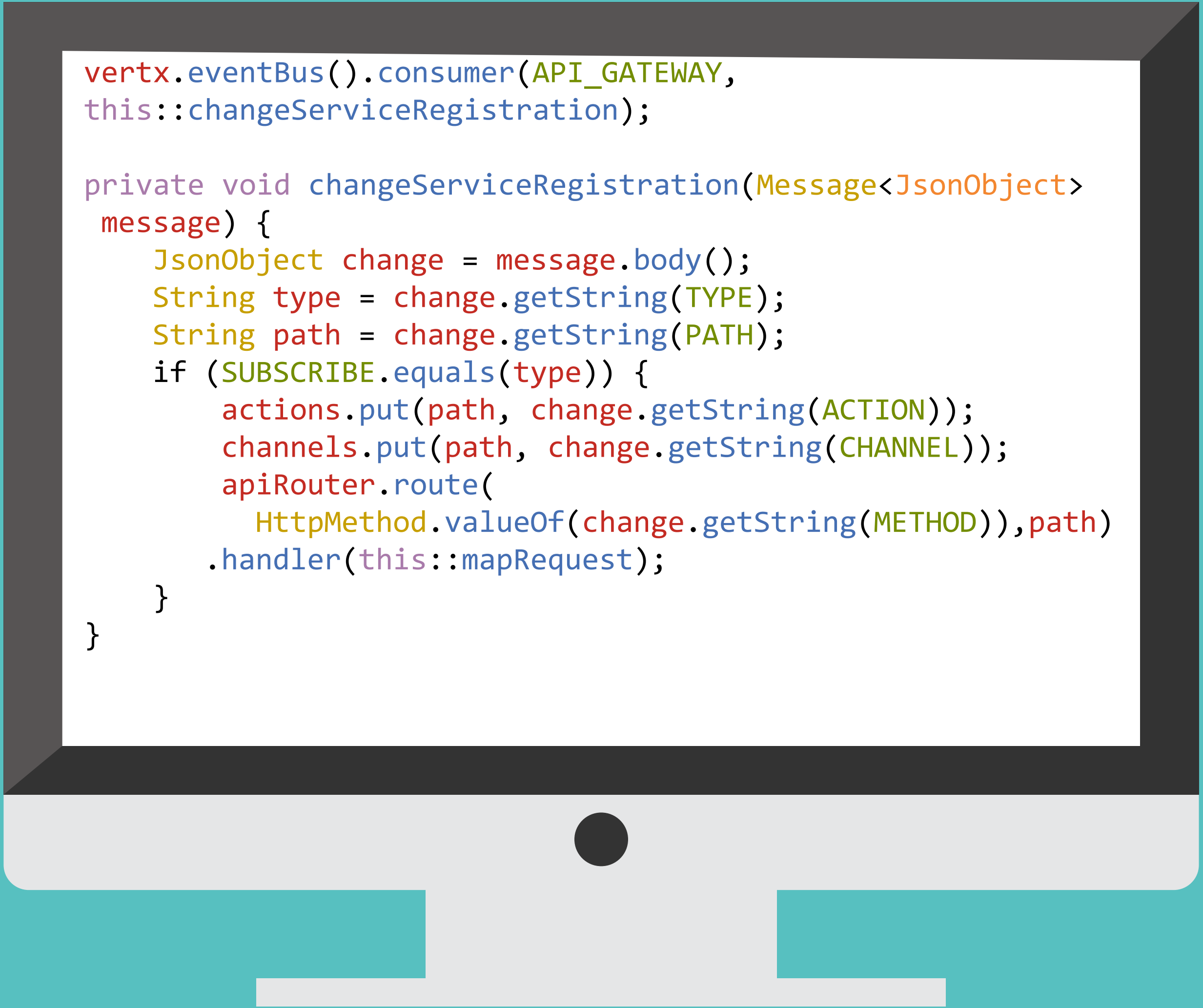
```
vertx.setPeriodic(5000, time -> {
    testManager.findAllTestsForStatus(RUNNING,
        result -> {
            if(result.succeeded()){
                if(result.result().size() < MAX_CONCURRENT){
                    startNextTest();
                }
            }
        });
});
```


Log Repository
Web Router
Job Scheduler
Job Repository
Test Manager

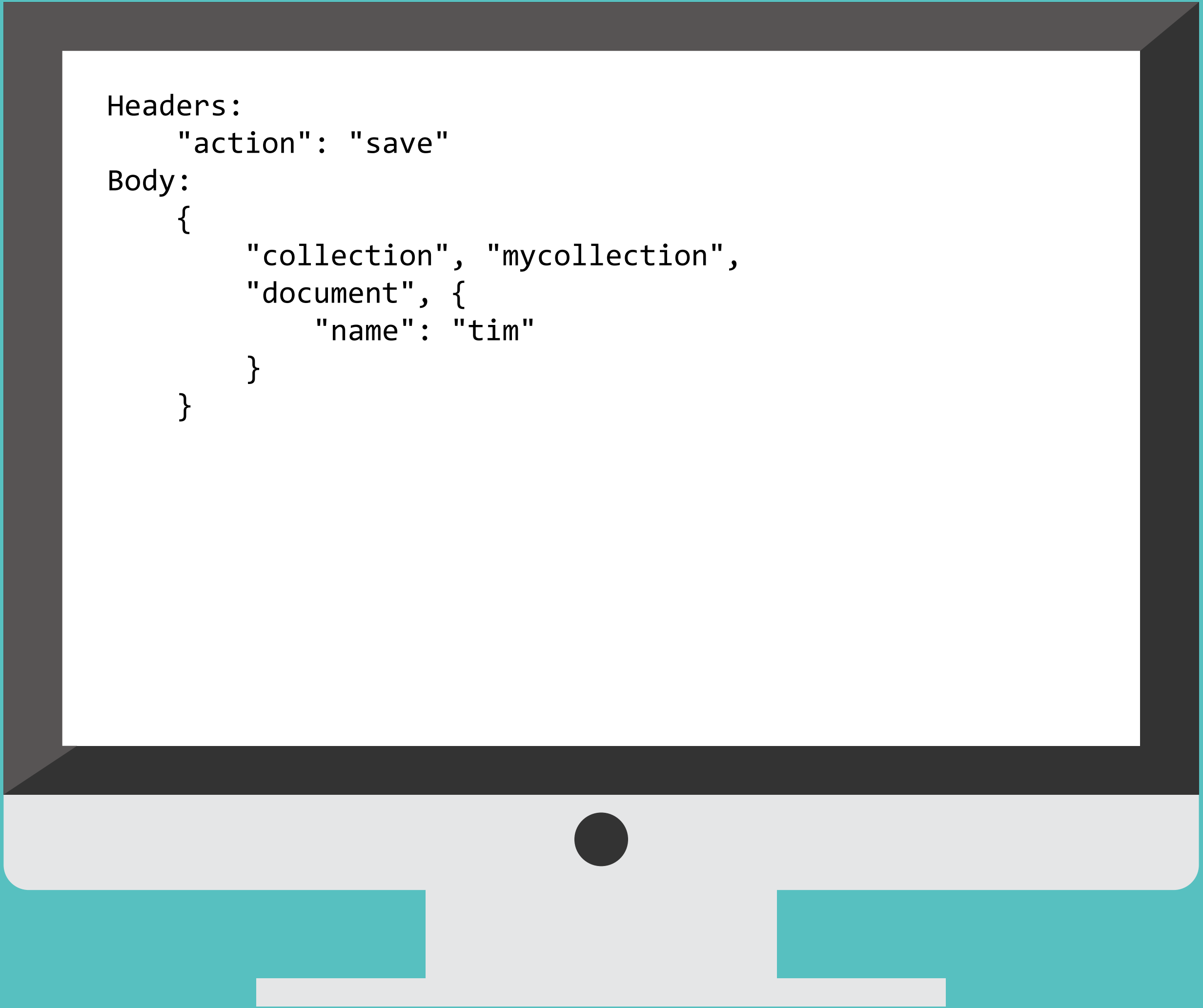


```
Router router = Router.router(vertx);
router.get("/api/tests").handler(...);
router.put("/api/tests/cancel").handler(...);

router.get("/api/logs").handler(...);

router.get("/api/jobs").handler(...);
router.post("/api/jobs").handler(...);
router.put("/api/jobs").handler(...);
router.delete("/api/jobs").handler(...);

router.get("/api/components").handler(...);
router.get("/api/channels").handler(...);
```

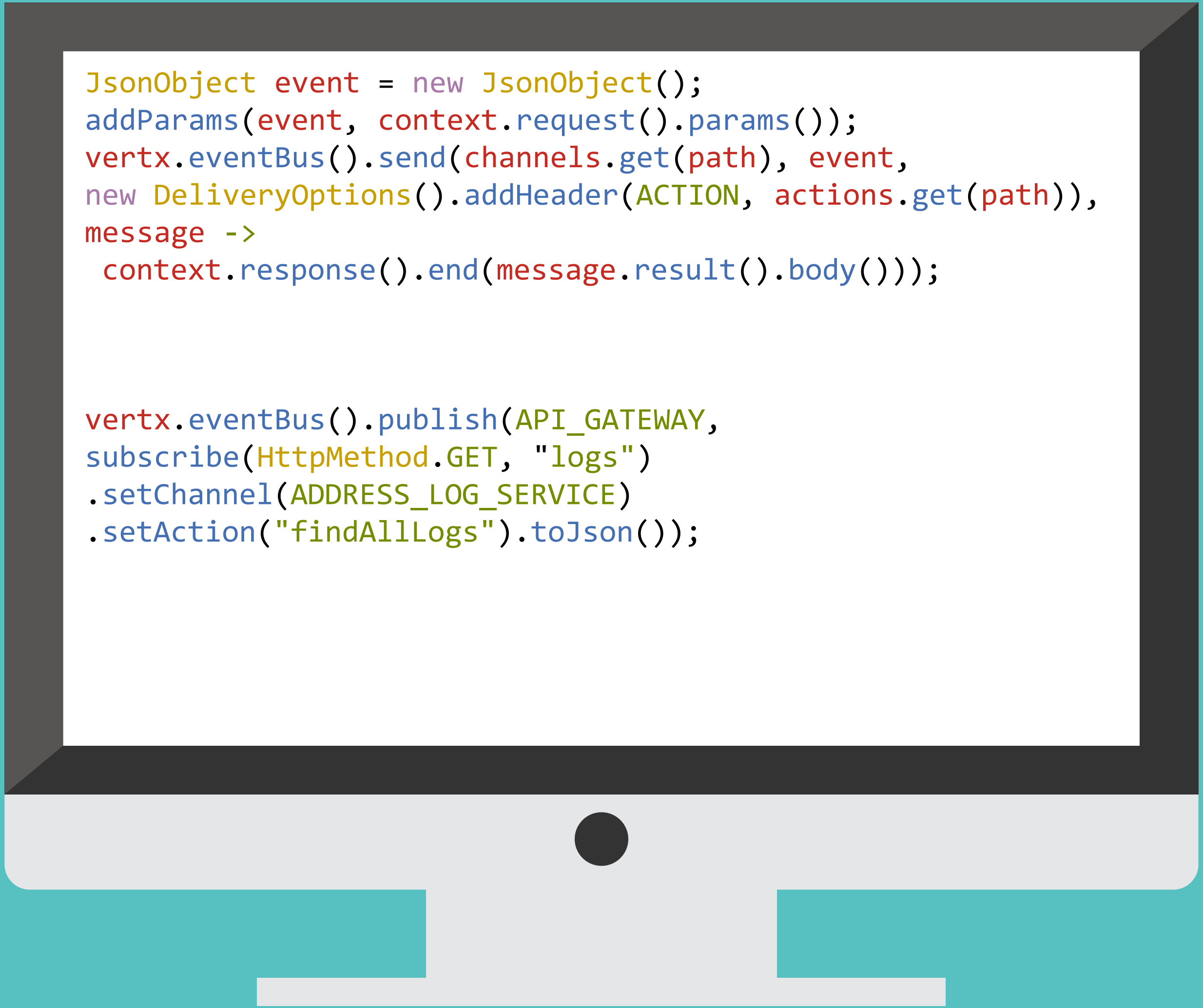

GET /logs
Web Router
Subscribe
Message
Result
Log Repository



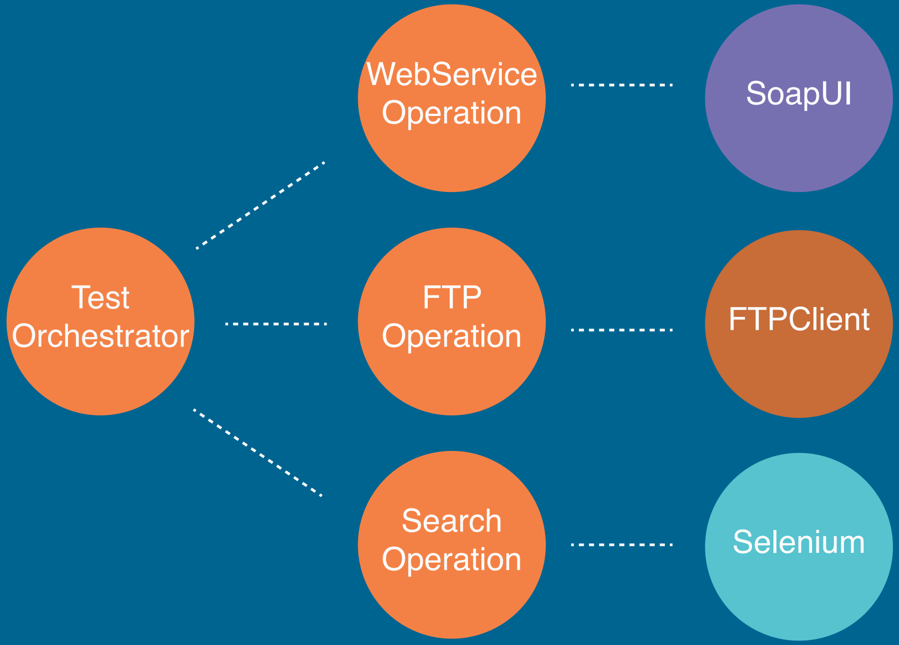
```
vertx.eventBus().consumer(API_GATEWAY,
this::changeServiceRegistration);

private void changeServiceRegistration(Message<JsonObject>
 message) {
    JsonObject change = message.body();
    String type = change.getString(TYPE);
    String path = change.getString(PATH);
    if (SUBSCRIBE.equals(type)) {
        actions.put(path, change.getString(ACTION));
        channels.put(path, change.getString(CHANNEL));
        apiRouter.route(
          HttpMethod.valueOf(change.getString(METHOD)),path)
       .handler(this::mapRequest);
    }
}
```

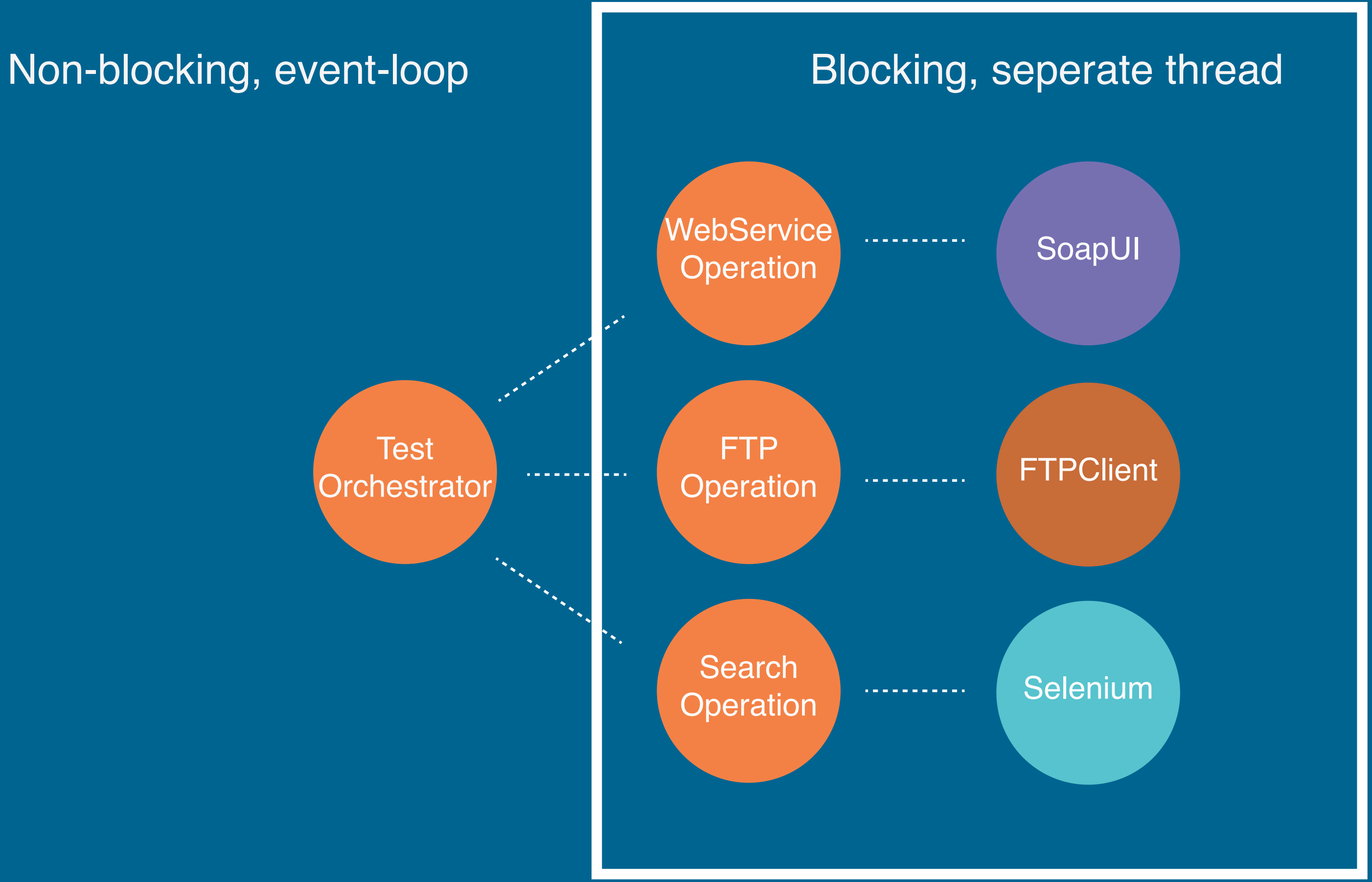


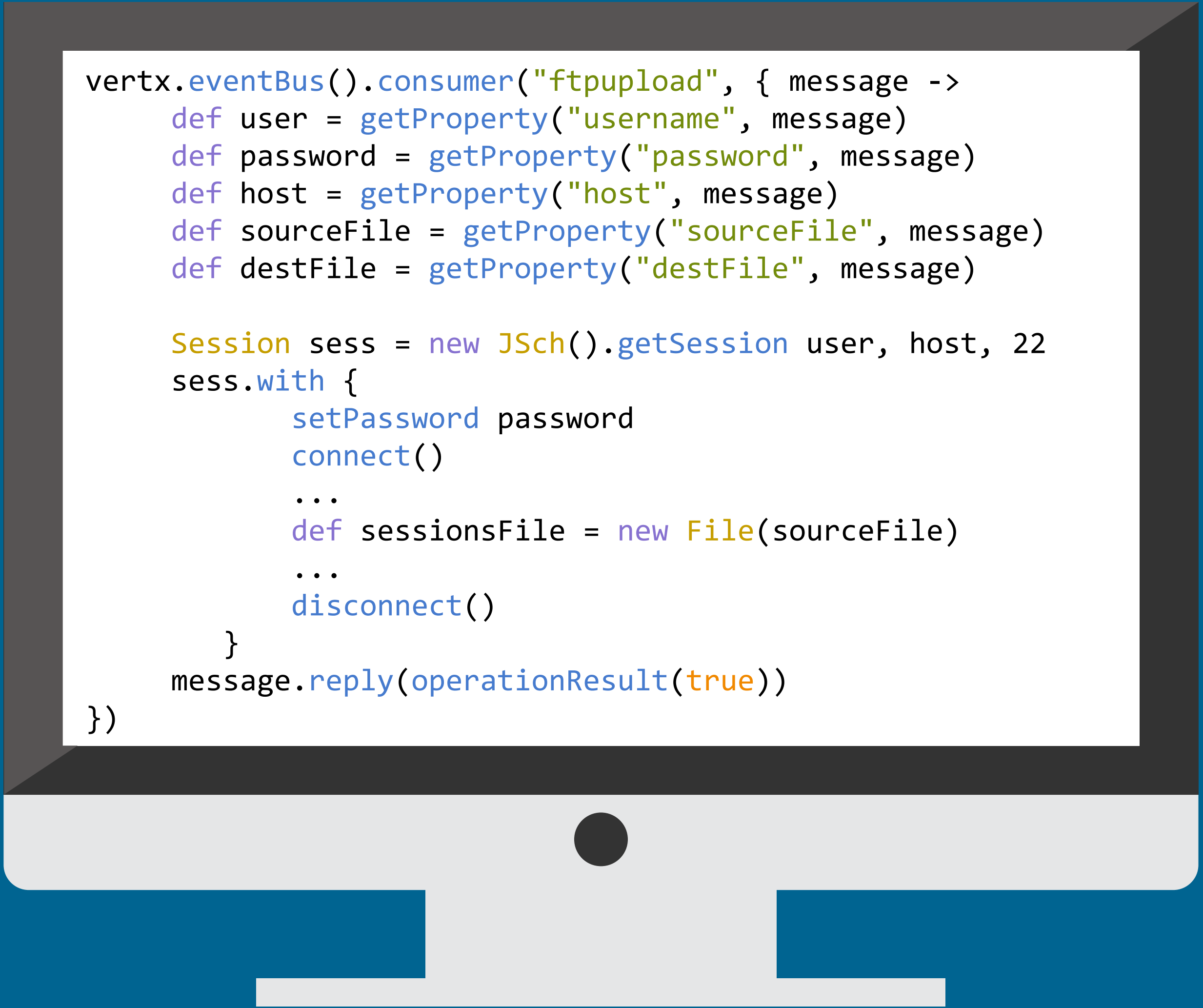
```
Headers:
    "action": "save"
Body:
    {
        "collection", "mycollection",
        "document", {
            "name": "tim"
        }
    }
```



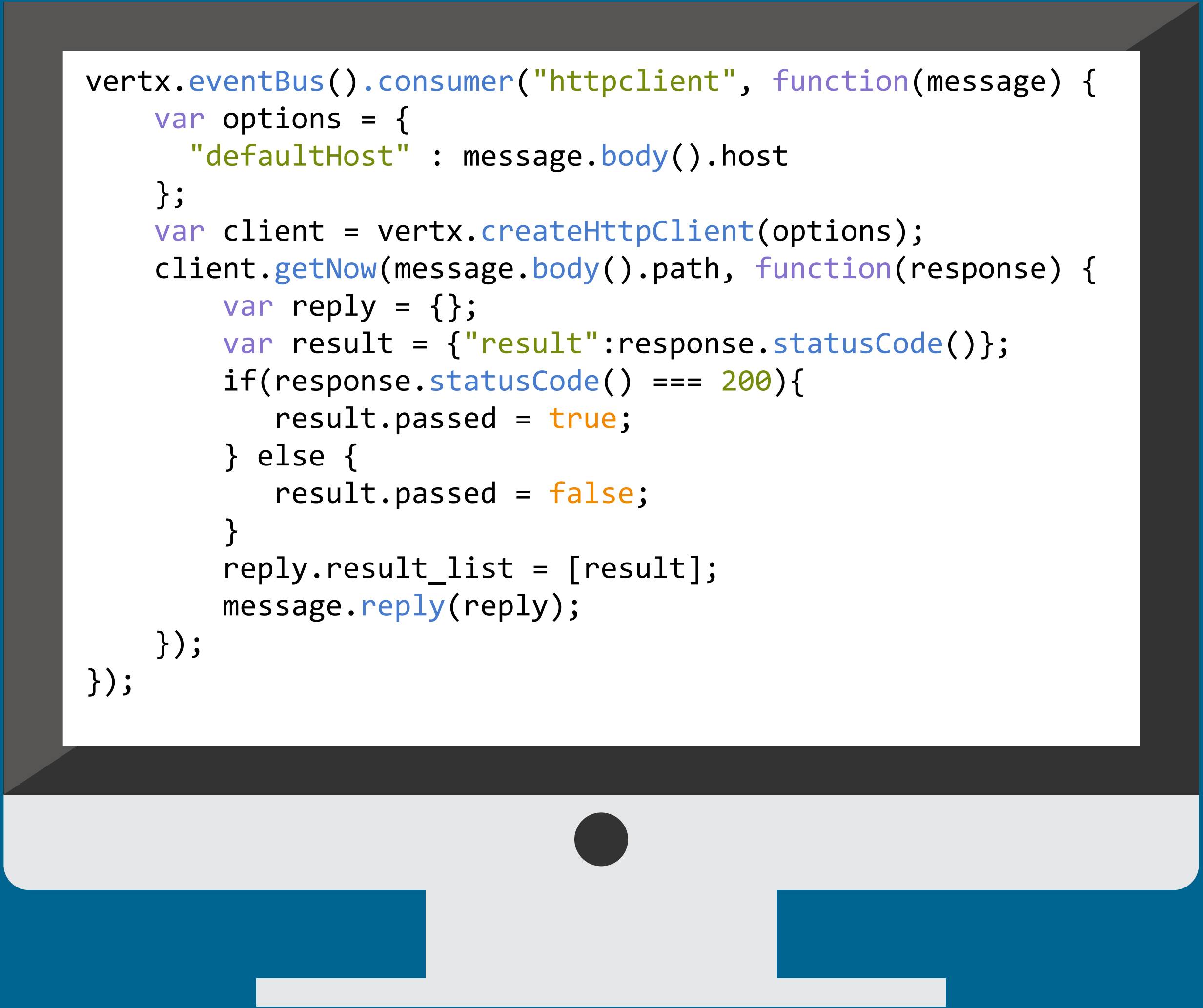
```
JsonObject event = new JsonObject();
addParams(event, context.request().params());
vertx.eventBus().send(channels.get(path), event,
new DeliveryOptions().addHeader(ACTION, actions.get(path)),
message ->
 context.response().end(message.result().body()));



vertx.eventBus().publish(API_GATEWAY,
subscribe(HttpMethod.GET, "logs")
.setChannel(ADDRESS_LOG_SERVICE)
.setAction("findAllLogs").toJson());
```


Test Orchestrator
WebService Operation
FTP Operation
Search Operation
SoapUI
FTPClient
Selenium


Non-blocking, event-loop
Blocking, seperate thread
Test Orchestrator
WebService Operation
FTP Operation
Search Operation
SoapUI
FTPClient
Selenium



```
vertx.eventBus().consumer("ftpupload", { message ->
     def user = getProperty("username", message)
     def password = getProperty("password", message)
     def host = getProperty("host", message)
     def sourceFile = getProperty("sourceFile", message)
     def destFile = getProperty("destFile", message)

     Session sess = new JSch().getSession user, host, 22
     sess.with {
            setPassword password
            connect()
            ...
            def sessionsFile = new File(sourceFile)
            ...
            disconnect()
        }
     message.reply(operationResult(true))
})
```

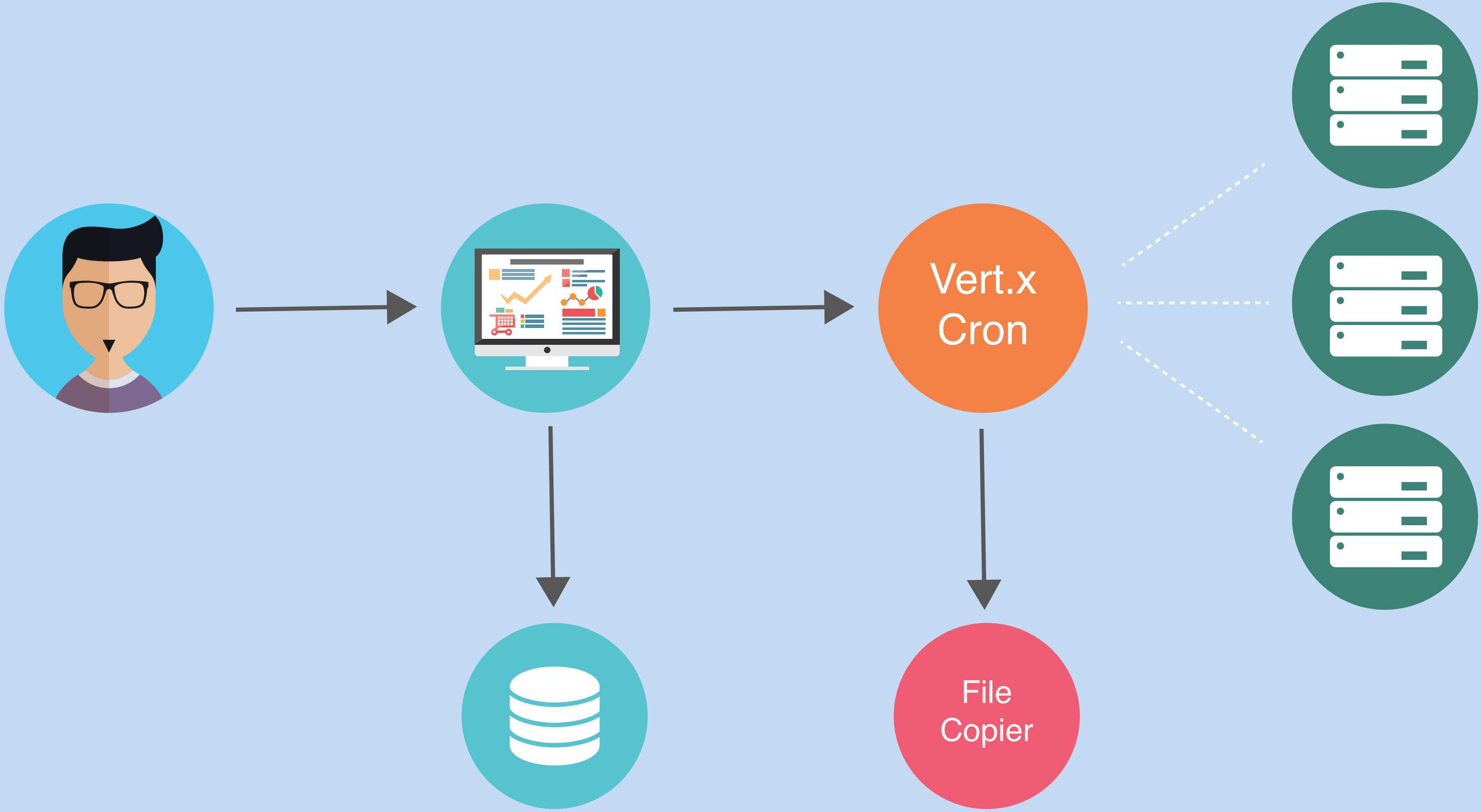


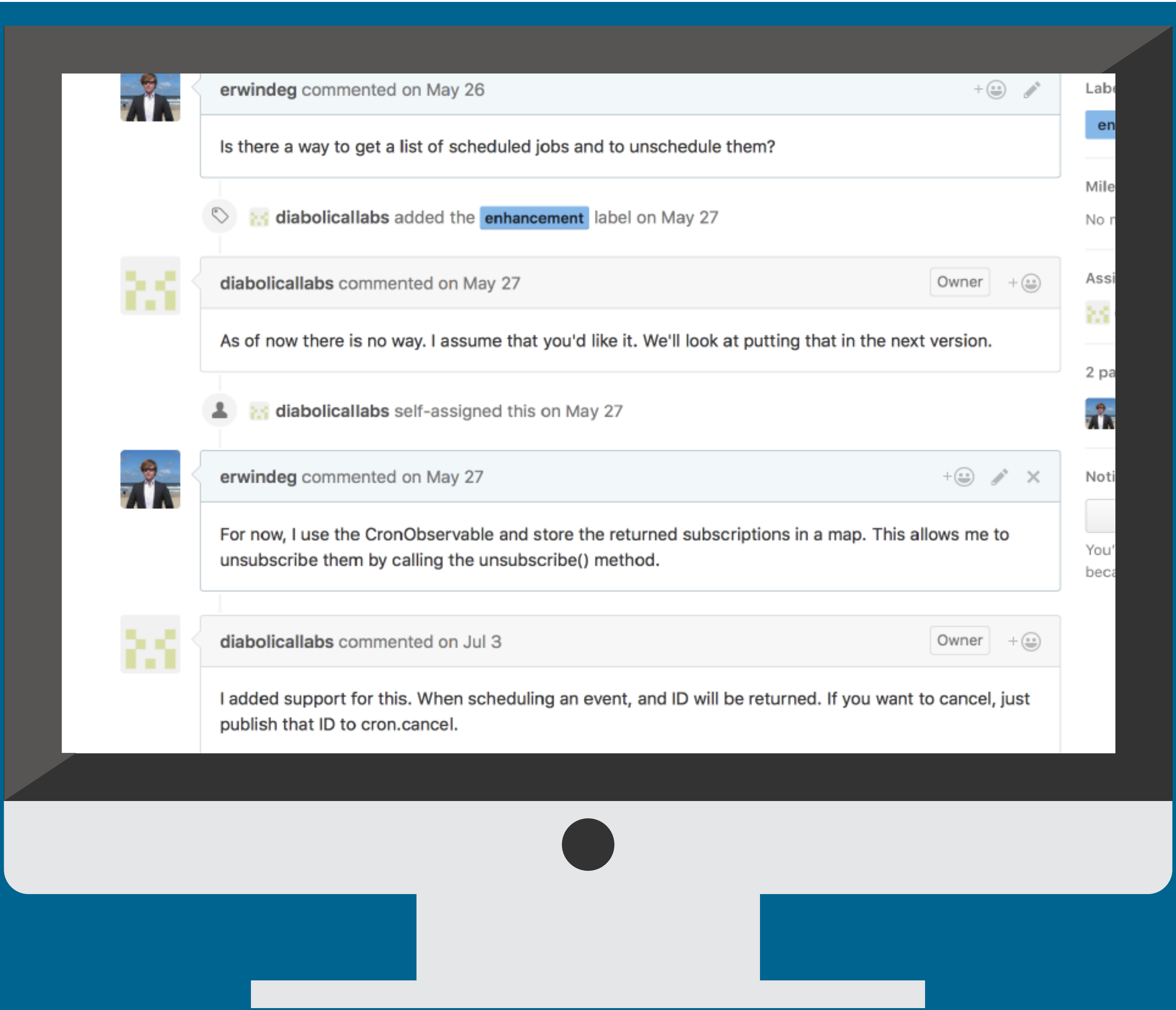
```
vertx.eventBus().consumer("httpclient", function(message) {
    var options = {
      "defaultHost" : message.body().host
    };
    var client = vertx.createHttpClient(options);
    client.getNow(message.body().path, function(response) {
        var reply = {};
        var result = {"result":response.statusCode()};
        if(response.statusCode() === 200){
           result.passed = true;
        } else {
           result.passed = false;
        }
        reply.result_list = [result];
        message.reply(reply);
    });
});
```

File system

JAR

Maven
Repository


Vert.x
Cron
File
Copier



erwindeg commented on May 26

Is there a way to get a list of scheduled jobs and to unschedule them?

diabolicallabs added the **enhancement** label on May 27

diabolicallabs commented on May 27 Owner

As of now there is no way. I assume that you'd like it. We'll look at putting that in the next version.

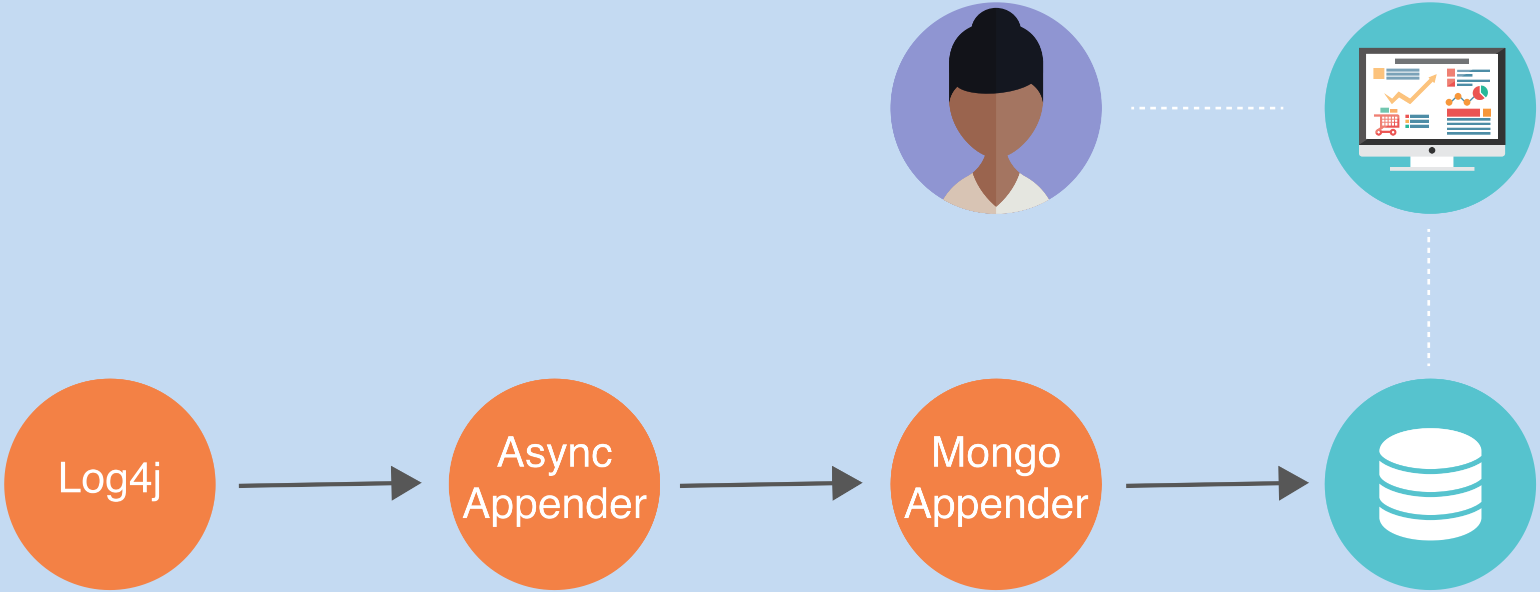
diabolicallabs self-assigned this on May 27

erwindeg commented on May 27

For now, I use the CronObservable and store the returned subscriptions in a map. This allows me to unsubscribe them by calling the unsubscribe() method.

diabolicallabs commented on Jul 3 Owner

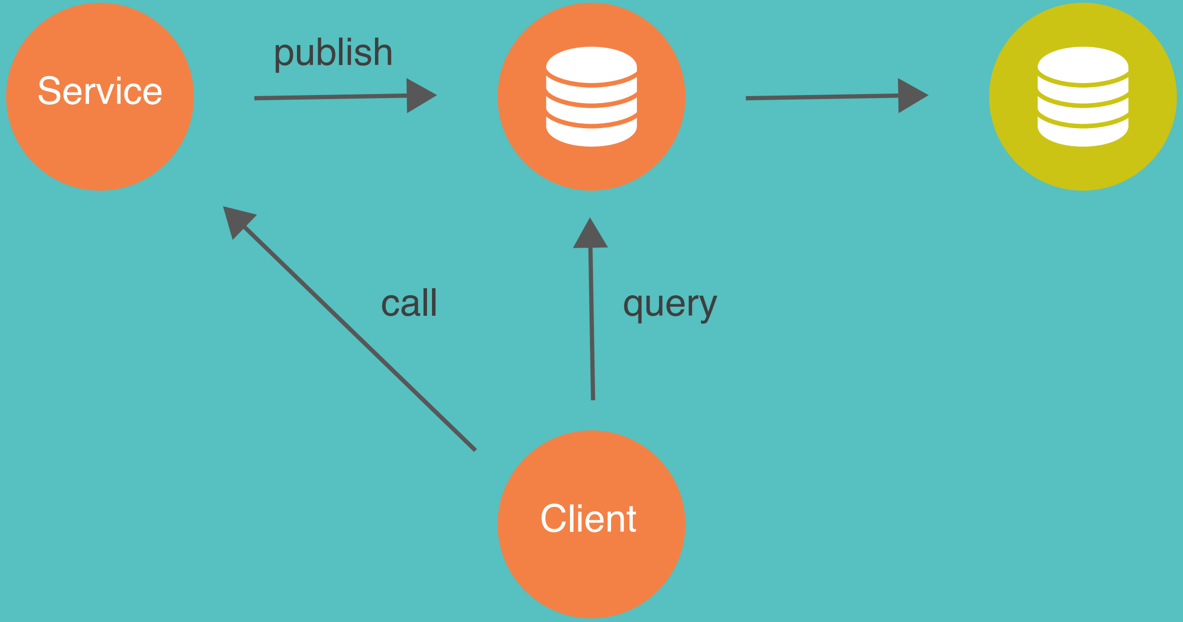
I added support for this. When scheduling an event, and ID will be returned. If you want to cancel, just publish that ID to cron.cancel.


Log4j
Async Appender
Mongo Appender

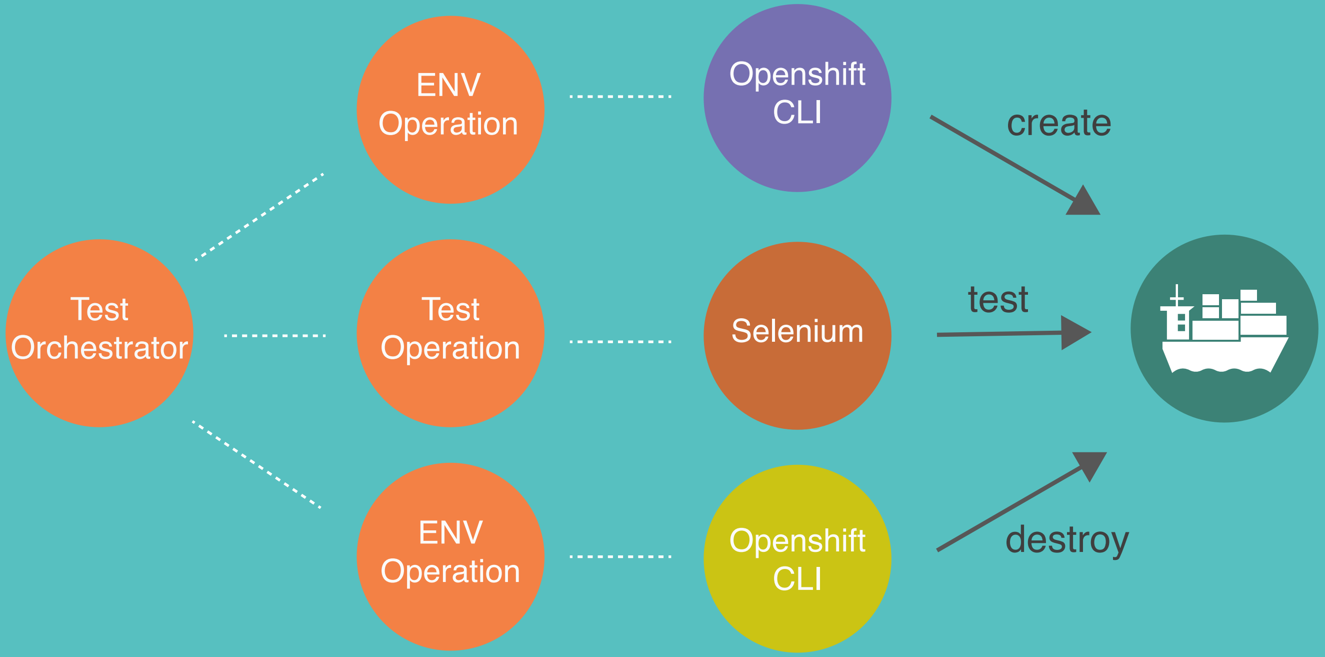
Future Work

- ✓ Given When Then
- ✓ CI/CD integration
- ✓ Service Discovery (Vert.x 3.3)
- ✓ Docker

Vert.x service discovery



Docker Integration



Vert.x
Events
Async
Polyglot

“Do one thing, and do it well.”

- unix philosophy

http://edegier.nl/presentations/
erwin@edegier.nl

Thanks to Vincent Wijnen for his idea for the test automation framework, Sogeti for letting me build it and speak about it and Livia Rickli for her help with the presentation slides.