FRESSIA: A Testing Framework for VO Applications

Christopher J. Miller, Alvaro Egana, Exequiel Fuentes, and Ricardo Massad

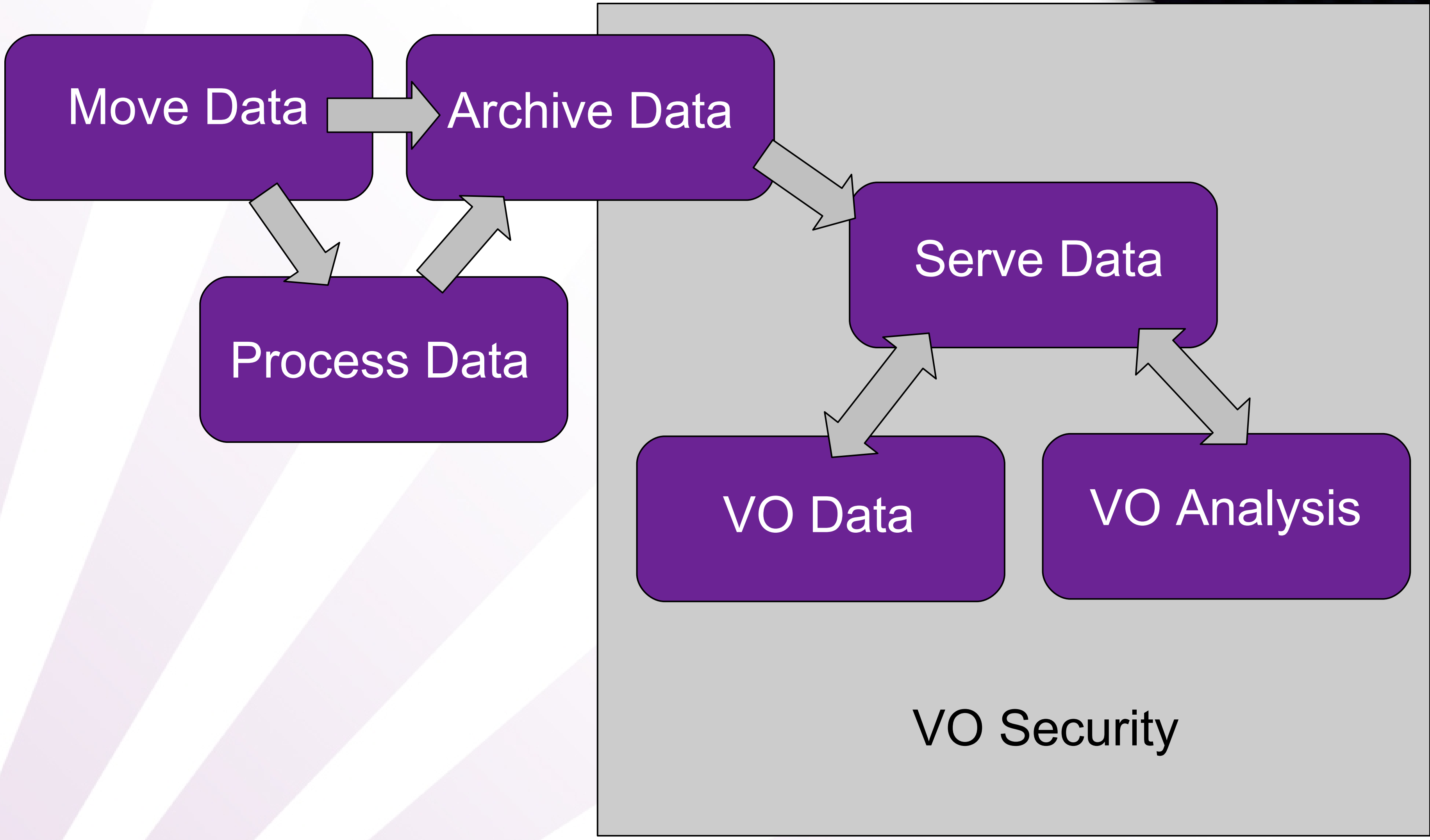
Data Products Program

CTIO/NOAO

VO App Testing

- Web-based testing (client side)
  - So many browsers
  - So many shared services
  - So many features (AJAX, FLASH, etc.)
- More than just the web (server side)
  - SOAP and REST work behind the scenes
  - Internal DBs
  - Security mechanisms
- Users EXPECT a working INTEGRATED system
  - Testing moves beyond simple unit tests and functional tests.....
  - .....and into the realm of true integrated testing frameworks.

SOA-based Distributed Apps



Enter FRESSIA

- What is it?
  - A framework for writing tests for rich applications
  - Written in JAVA
  - Utilizes a simple Domain Specific Language (DSL).
- Who is it for?
  - Anyone who needs integrated application testing (i.e., beyond unit tests and simple functional tests).
  - Anyone whose applications utilize multiple languages (Java, C, IRAF, etc.) or multiple service calls (SOAP, REST, etc.).
  - Scientists!
- Why do we need it?
  - Allows testers to work at the highest application level i.e., the user) and outside of the middleware layers.
  - Eases integrated application Regression, Stress, and Performance testing.

A FRESSIA Test

- Test
  - identifier
  - action block
    - action definition
    - action options
    - action events
  - asserts blok
    - assert definitions

```
suite suite_id {

    test
    test_id
    {
        [action] : <the action>;
        <option> : <value>;
        <type> <cond> [(<argument>)];
    }
    asserts
    {
        <type> <cond> [(<argument>)];
    }

}
```

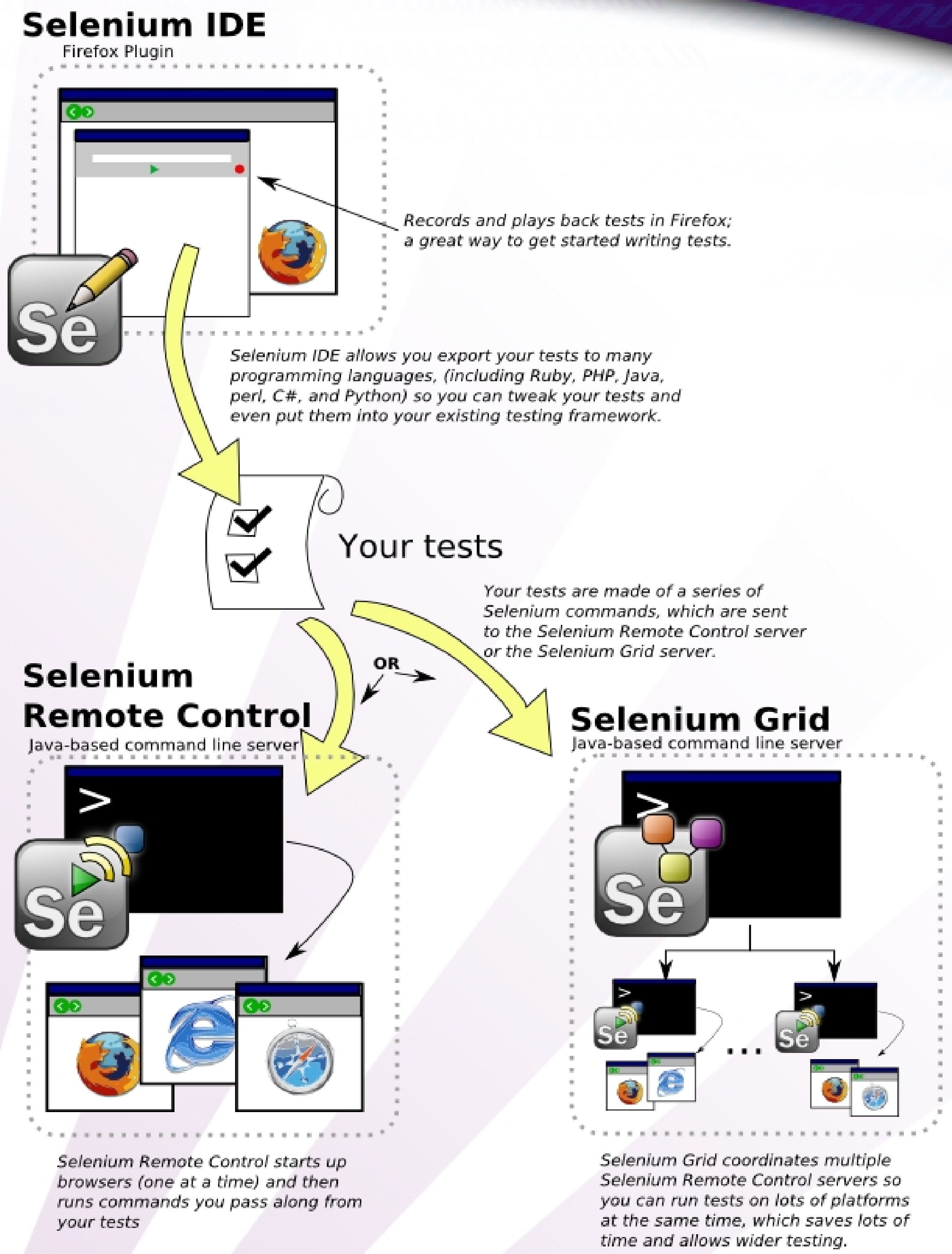
FRESSIA: example

```
suite google_tests {

        $var=”http://www.google.com”;

        test can_call {
                [action] : rest call;
                url : $var;
        }

        test test_html_integrity {
                [action] : rest call;
                url : $var;
        }
        asserts {
                html isValid;
                text contains ("<title>Google</title>");
        }

}
```

ACTION NAME	DESCRIPTION
email reception email	Used to deteremine whether a message has been received (via an SMTP server)
webgui events webgui	Used for browser-based user actions (called events)
blocking command	email reception
command	Used for any command line user action (e.g., compiled binaries)
rest call http call https call	HTTP/REST calls returning either HTML or XML
rest download http download https download	HTTP/REST calls and downloading the result

FRESSIA: WEBGUI Events

EVENT NAME	DESCRIPTION	CALL
click at	Click the mouse at a web page object element ID	**click at (" id ");**
click onLink	Click the mouse on a link in the webpage	**click onLink (" [sw=vm] ; nl ");**
selectFrom	Select from a LIST, OPTION box, or CHECK BOX	**selectFrom list (" id { et1 ; et2 ; ... ; etN } ");**
execute elementVerification	Verify that web page element ID exists	**execute elementVerification (" timew ; id ");**
execute javascript	Run a piece of JS	**execute javascript (" ${ ra } ");**
enter	Enter text into a form box	**enter text (" id { t1;t2;..;tn } ");**


Selenium IDE
Firefox Plugin
Records and plays back tests in Firefox; a great way to get started writing tests.
Selenium IDE allows you export your tests to many programming languages, (including Ruby, PHP, Java, perl, C#, and Python) so you can tweak your tests and even put them into your existing testing framework.
Your tests
Your tests are made of a series of Selenium commands, which are sent to the Selenium Remote Control server or the Selenium Grid server.
OR
Selenium Remote Control
Java-based command line server
Selenium Grid
Java-based command line server
Selenium Remote Control starts up browsers (one at a time) and then runs commands you pass along from your tests
Selenium Grid coordinates multiple Selenium Remote Control servers so you can run tests on lots of platforms at the same time, which saves lots of time and allows wider testing.

FRESSIA's WEBGUI action and its associated events uses the Selenium Remote Control

In practice, the Selenium IDE allows testers to get started, while the RC and Grid are used for “production testing”

FRESSIA: Summary

- DOES NOT replace unit tests, aliveness tests, functional tests, etc.
- Provides a simple Domain Specific Language for Selenium browser-based app testing.
  - The browser-based approach allows for testing in the “user's space” as opposed to the “developer's space”.
  - This same DSL is then enabled for other tests (e.g., unit tests, aliveness tests, functional tests, etc).
- Integrates all forms of application testing into one environment.
  - The environment is based upon a user-friendly (i.e., readable) DSL
  - The environment allows multiple test suites to be strung together
  - The environment enables result reporting on the entire integrated test suite.