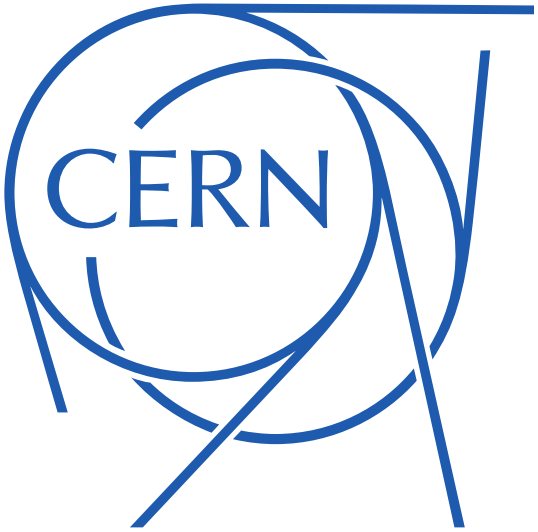

CERN

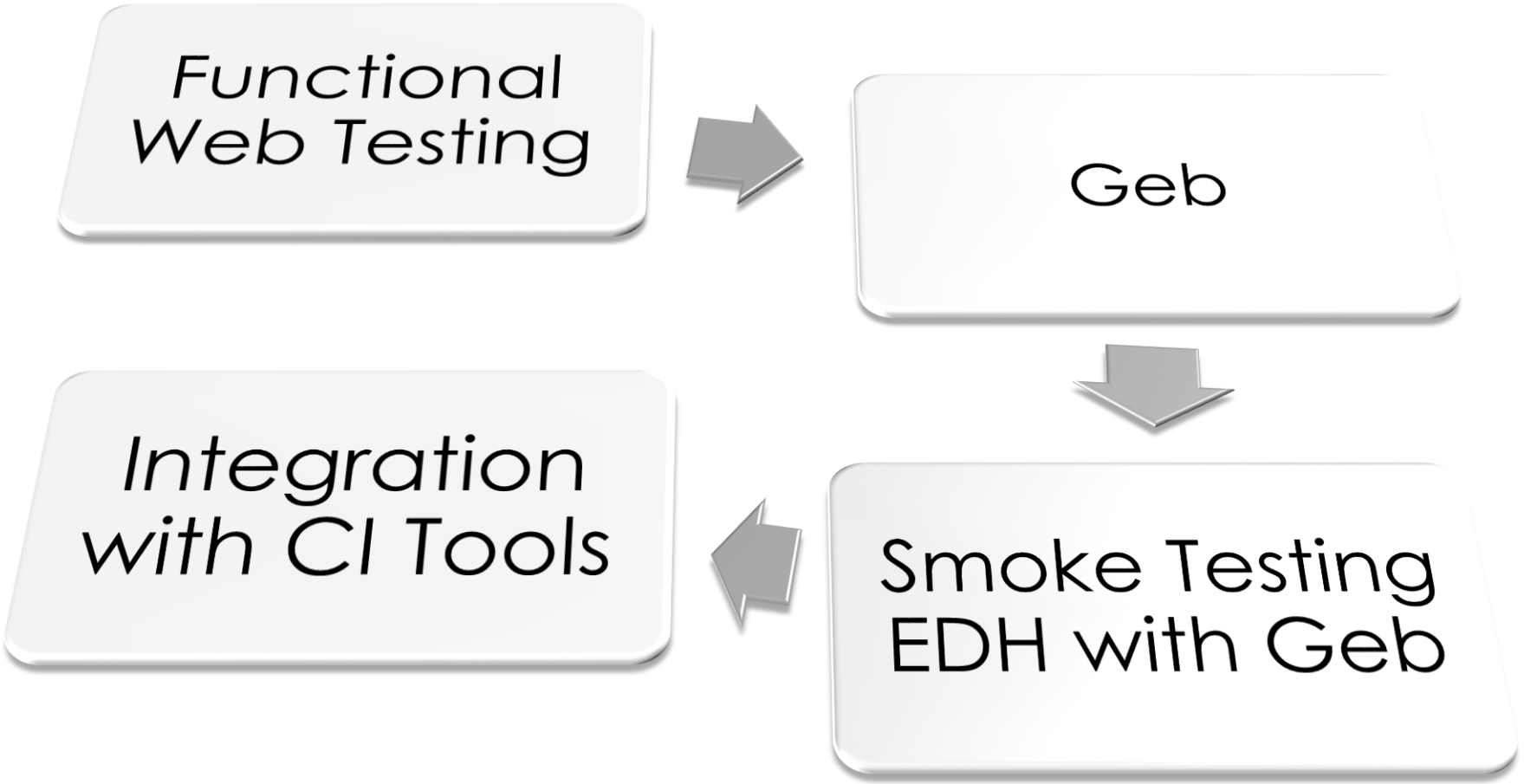
e-Business
Section

Automated Smoke Testing on the JVM with Geb

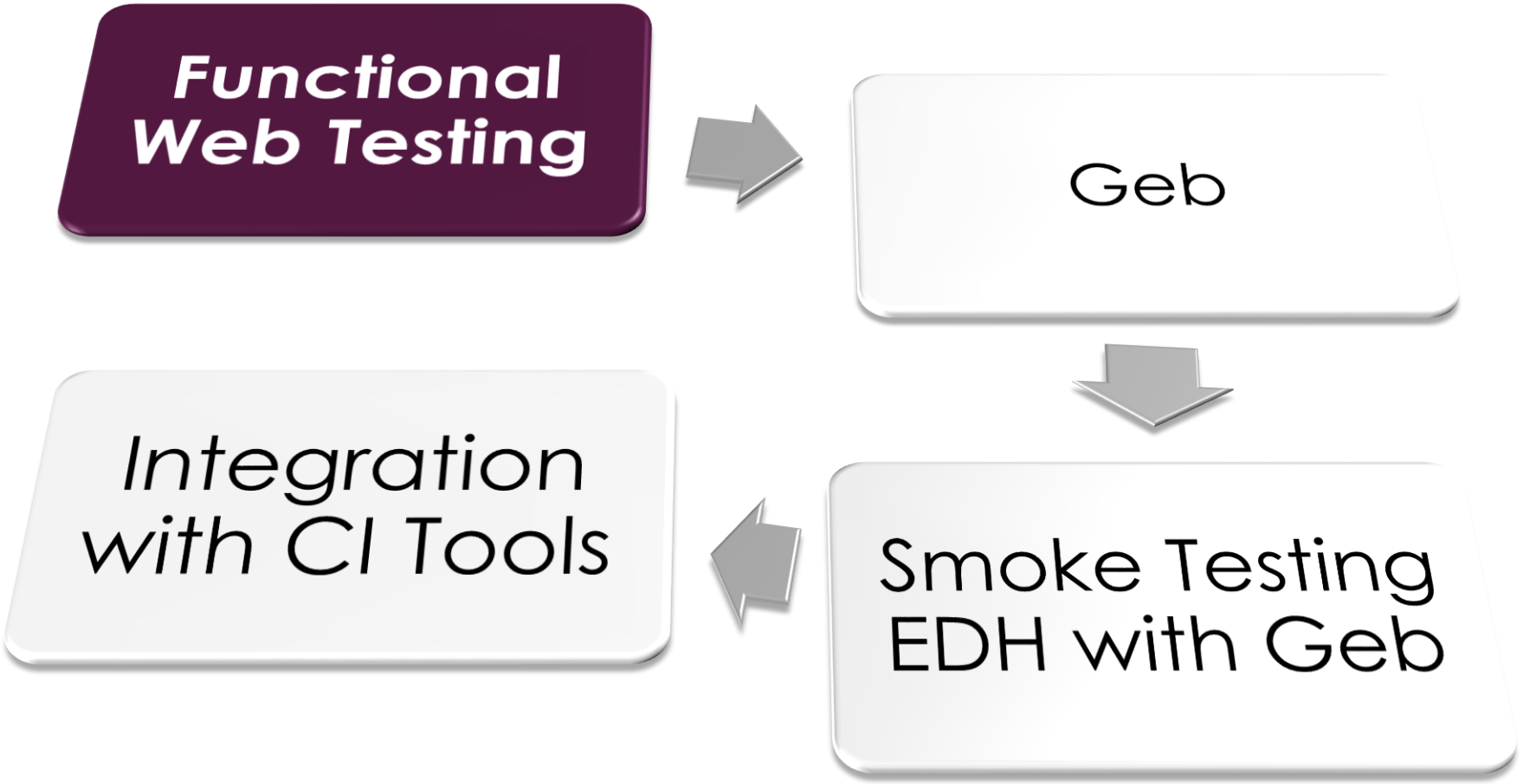
João SILVA (GS-AIS-EB)

1st Developers@CERN Forum

Agenda


Functional Web Testing
Geb
Smoke Testing EDH with Geb
Integration with CI Tools

Agenda


Functional Web Testing
Geb
Smoke Testing EDH with Geb
Integration with CI Tools

Functional Testing

- Type of **black-box** testing
  - Ignores the specifics of the underlying software component under test
- Asserts that providing certain inputs results in certain outputs
- Contrasts with white-box testing (e.g. unit testing)

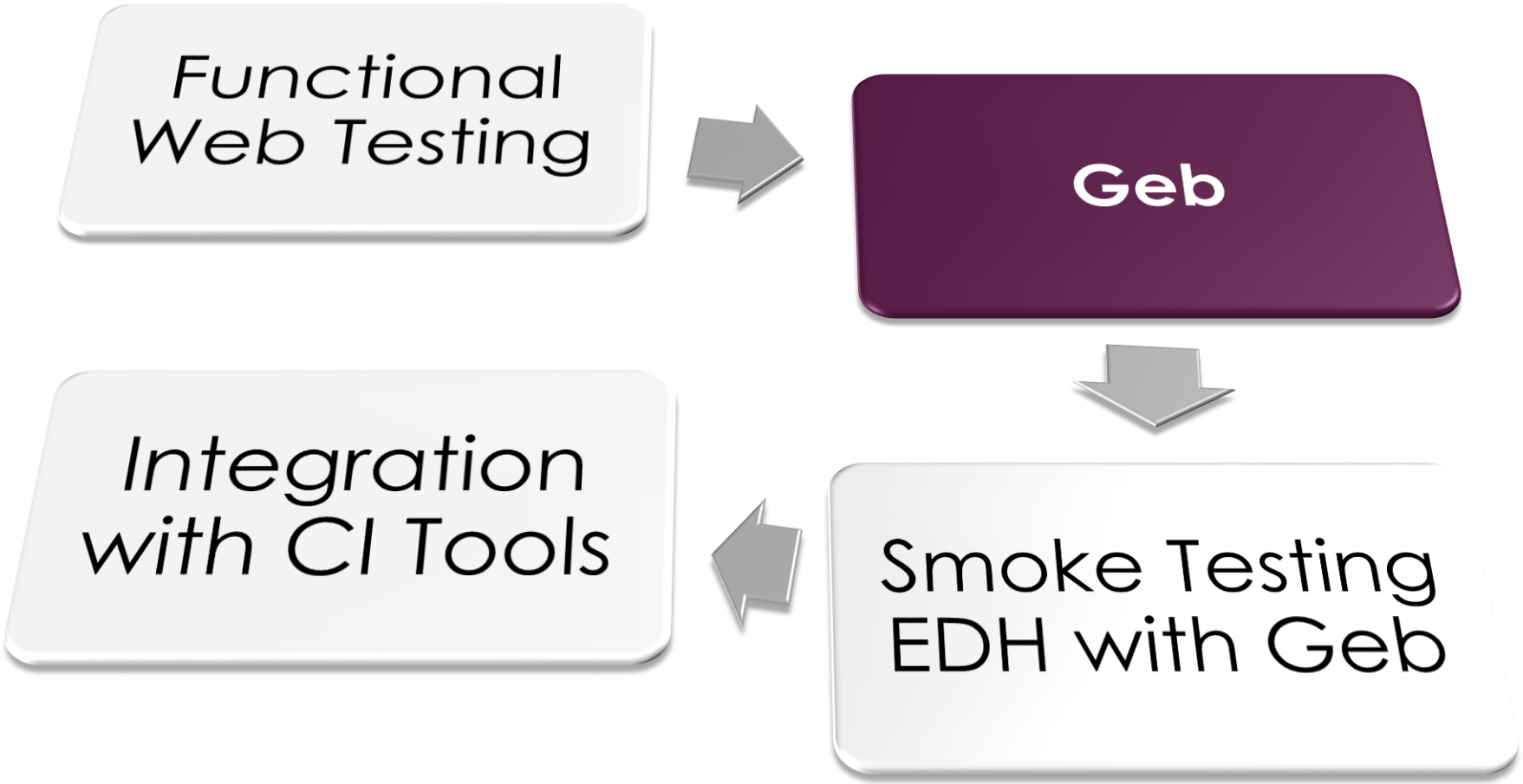
Functional Web Testing

- When developing for the web, **unit testing only goes so far**
  - Unable to test for JavaScript errors in a particular version of a particular browser
  - Unable to identify client-side regressions
  - Unable to test breaking changes in a REST API consumed on the client-side

Functional Web Testing

- Functional web test usually means **programmatically controlling** a web **browser** to simulate the **actions** of a user on a web page
- All application layers are tested at the same time
- Language agnostic

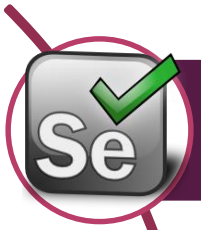
Agenda


Functional Web Testing
Geb
Smoke Testing EDH with Geb
Integration with CI Tools

What is Geb?

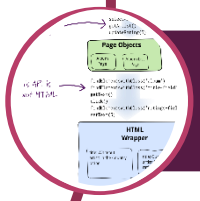
- **Browser automation solution**
- Written in Groovy
- Uses
  - Fuctional/Web/Acceptance **Testing**
  - Scripting
  - Screen scraping
- Current version is 0.12.2
  - But just like winter, version 1.0 is coming!

What's so great about Geb?



Power of **Selenium WebDriver**

Elegance of **jQuery**



Robustness of **Page Object** modelling

Expressiveness of **Groovy**

Power of Selenium WebDriver

- Code -> Driving -> Browser
- De-facto **standard** in the test automation world
- **Cross-browser** support

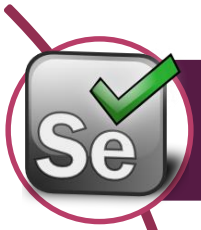
WebDriver Example

```
public class Example {
  public static void main(String[] args) {
    WebDriver driver = new FirefoxDriver();
    // Go to Google
    driver.get("http://www.google.com");

    // Find the text input element by its name
    WebElement element = driver.findElement(By.name("q"));

    // Enter something to search for
    element.sendKeys(“CERN”);

    // Now submit the form.
    element.submit();

    // Check the title of the page
    System.out.println("Page title is: " + driver.getTitle());

    driver.quit();
  }
}
```

WebDriver Example Async

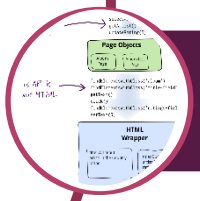
```
public class Example  {
  public static void main(String[] args) {
    WebDriver driver = new FirefoxDriver();

    driver.get("http://www.google.com");

    WebElement element = driver.findElement(By.name("q"));

    element.sendKeys(“CERN”);

    element.submit();

    // Poll until async request is complete
    (new WebDriverWait(driver, 10)).until(new ExpectedCondition<Boolean>()) {
      public Boolean apply(WebDriver d) {
        return d.getTitle().toLowerCase().startsWith("CERN");
      }
    });

    System.out.println("Page title is: " + driver.getTitle());

    driver.quit();
  }
}
```

What's so great about Geb?



Power of **Selenium WebDriver**

Elegance of **jQuery**



Robustness of **Page Object** modelling

Expressiveness of **Groovy**

Geb Example

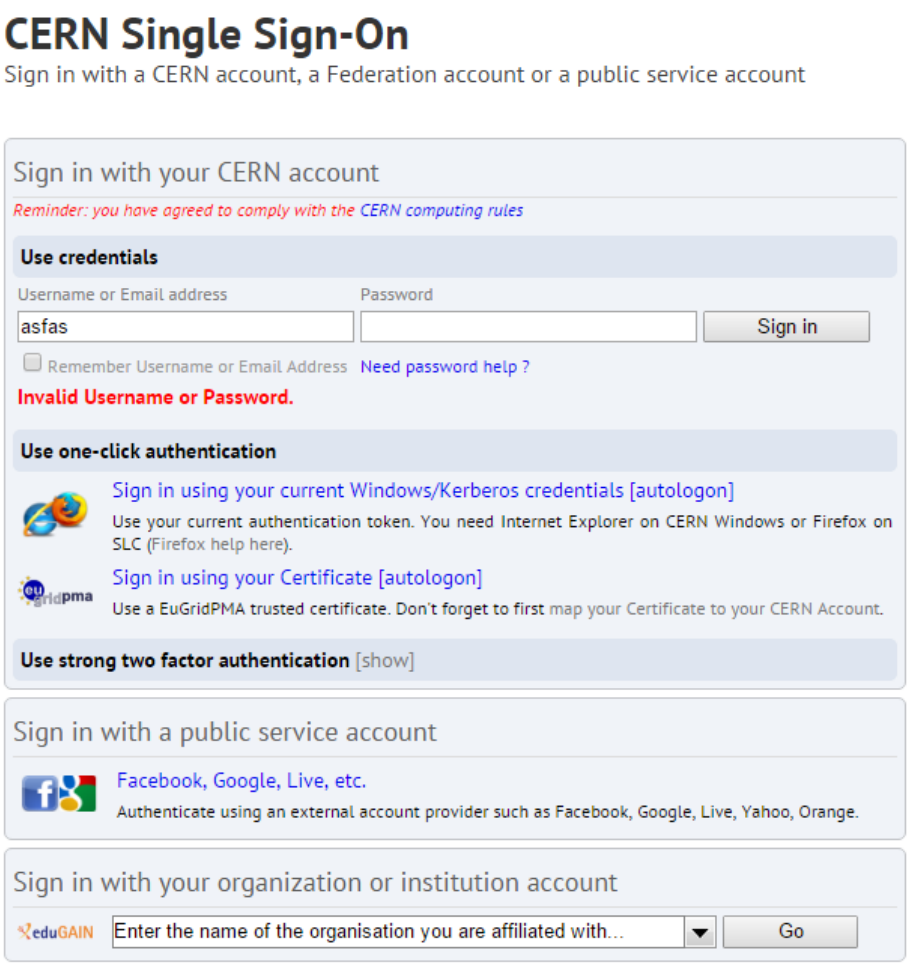
```groovy
Browser.drive("http://www.google.com") {
  def element = $("input", name: "q")
  element << "CERN"
  element << Keys.ENTER

  waitFor { title.startsWith("CERN") }

  def firstResultLink = $("li.g", 0).find("a.l")
  assert firstResultLink.text() == "CERN | Accelerating science"
}
```

With less code, comes less responsability.

Functional Testing with Geb

- Integration with tests frameworks
  - JUnit
  - **Spock**
  - TestNG

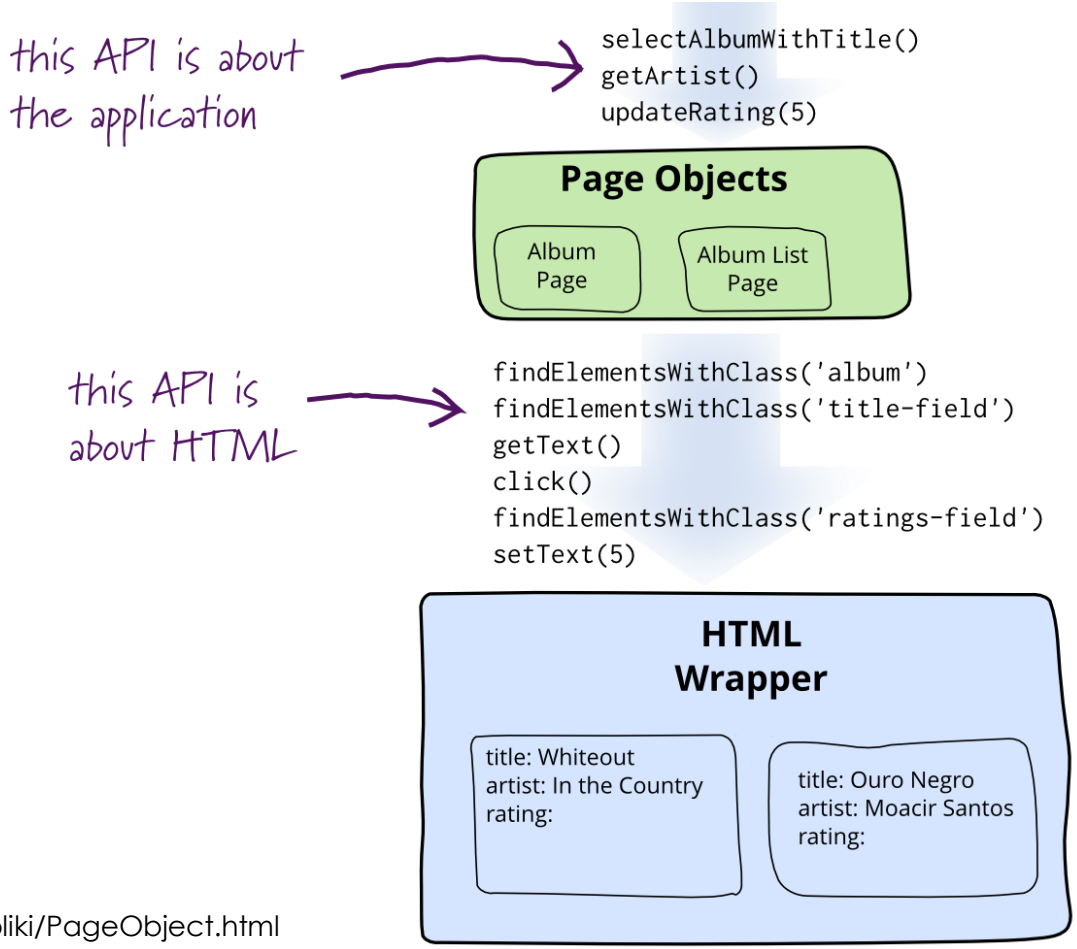
Web Testing CERN SSO Login



CERN SSO Login Spec

```
class CernSsoLoginSpec extends GebSpec {
  def "should redisplay form with an error when password is bad"() {
    when:
    go "https://login.cern.ch"

    then:
    $("#site-name").text() == "CERN Single Sign-On"

    when:
    $("#txtFormsLogin").value("username")
    $("#txtFormsPassword").value("badpassword")
    $("#btnFormsLogin").click()

    then:
    $("#site-name").text() == "CERN Single Sign-On"
    $("#lblFormsError").text() == "Invalid username or password"
  }
}
```

Page Object Pattern



http://martinfowler.com/bliki/PageObject.html

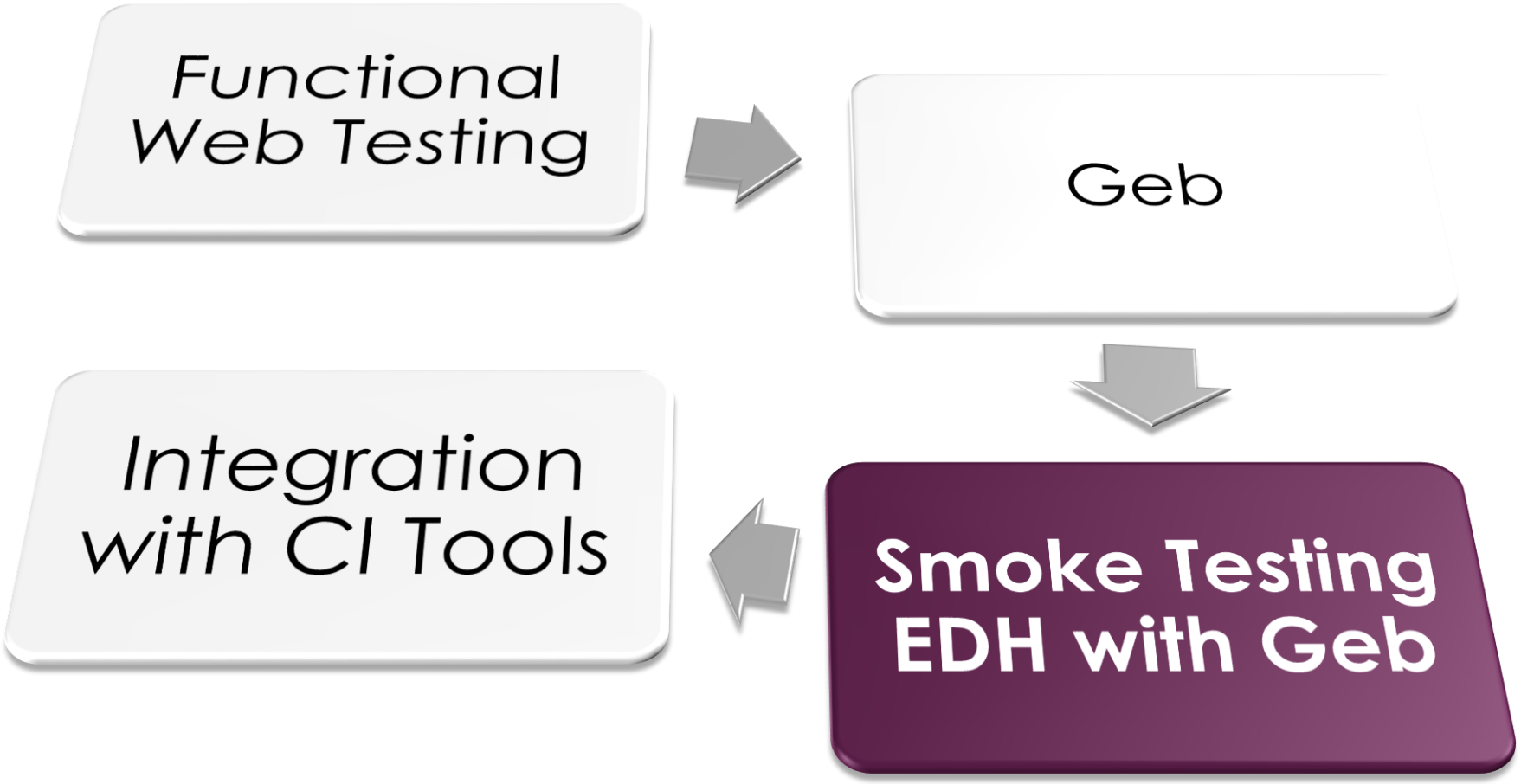
LoginPage

```
class LoginPage extends Page {

  static url = "https://login.cern.ch"

  static at = { $("#site-name").text() == "CERN Single Sign-On" }

  static content = {
    username { $("#txtFormsLogin") }
    password { $("#txtFormsPassword") }
    signIn { $("#btnFormsLogin") }
    errorMessage { $("#lblFormsError") }
  }

  private login(usr, pwd) {
    username = usr
    password = pwd
    signIn.click()
  }
}
```

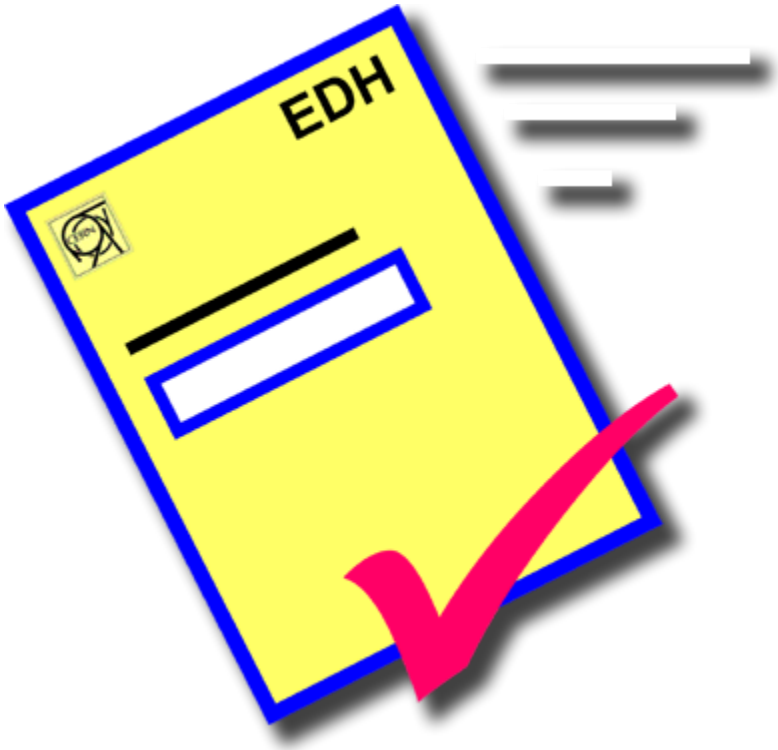
Login Spec with Page Objects

```
class CernSsoLoginSpec extends GebSpec {
  def "should redisplay form with an error when password is bad"() {
    when:
    to LoginPage

    when:
    login("username", "badpassword")

    then:
    at LoginPage
    errorMessage == "Invalid username or password"
  }
}
```

Agenda


Functional Web Testing
Geb
Smoke Testing EDH with Geb
Integration with CI Tools

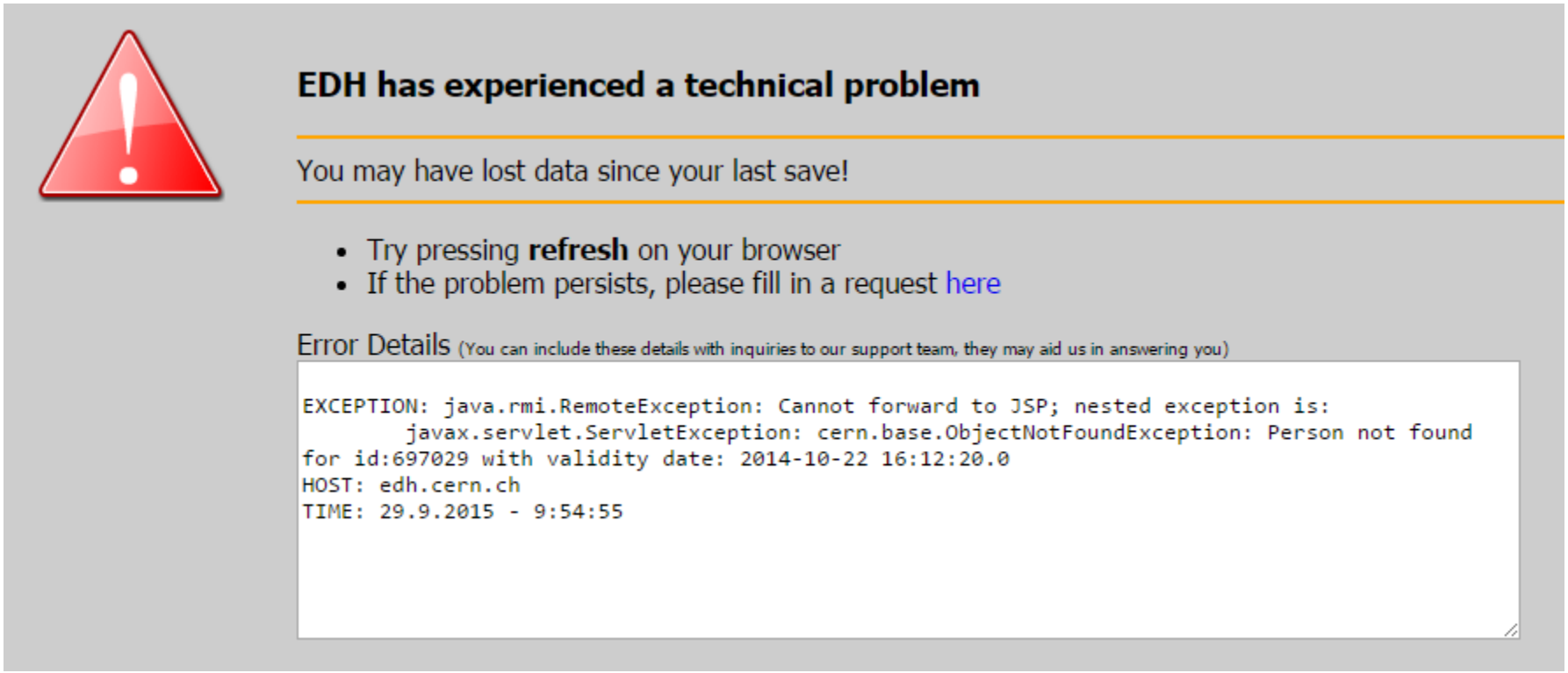
Electronic Document Handling



Electronic Document Handling

- Over **60** official procedures
  - Safety, procurement, finance, HR, etc.
- Over **360k** documents per year
- Over **750k** electronic signatures
- Developed in-house (**1992**), Web (**1998**)
- Over **1 million** lines of Java code

Smoke Testing



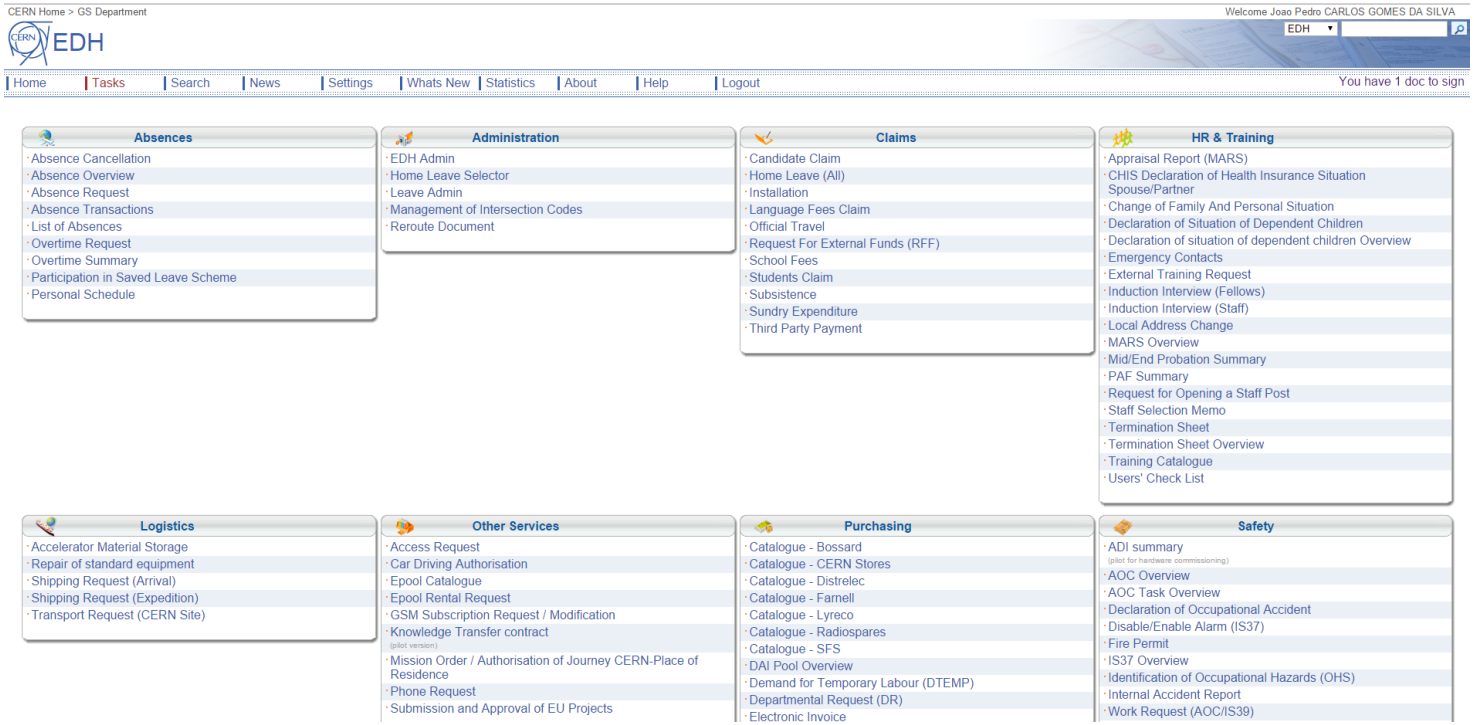
Smoke Testing EDH (1/3)

- Integration with CERN Single Sign-On

```
class EdhBaseSpec extends GebReportingSpec
{
  def "should be redirected to login page and then brought back"() {
    when: "I access EDH Desktop for the first time"
    via DesktopPage

    then: "I'm redirected to CERN SSO-Login"
    at LoginPage

    then: "I login with a service account"
    login()

    then: "I'm redirected back to EDH"
    at DesktopPage
  }
}
```

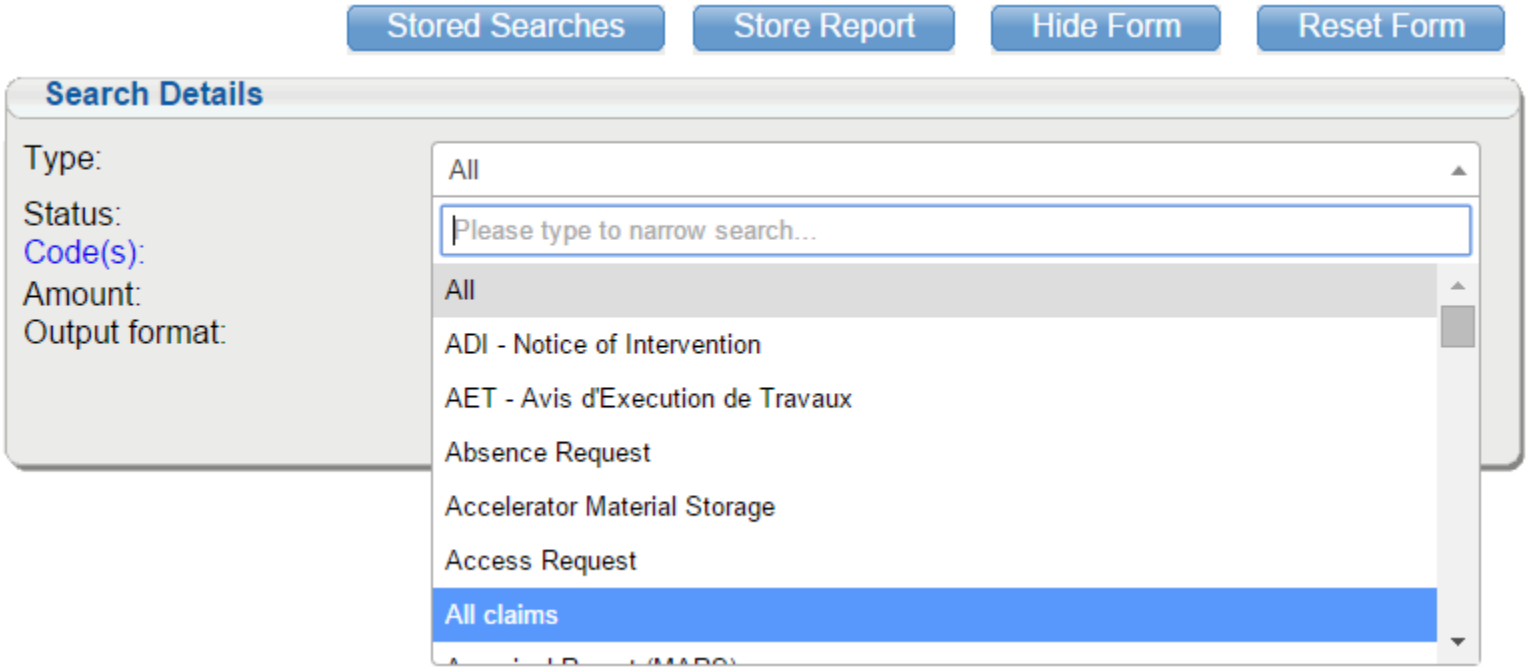
Smoke Testing EDH (2/3)

- Assert that every link/document on EDH Desktop can be opened

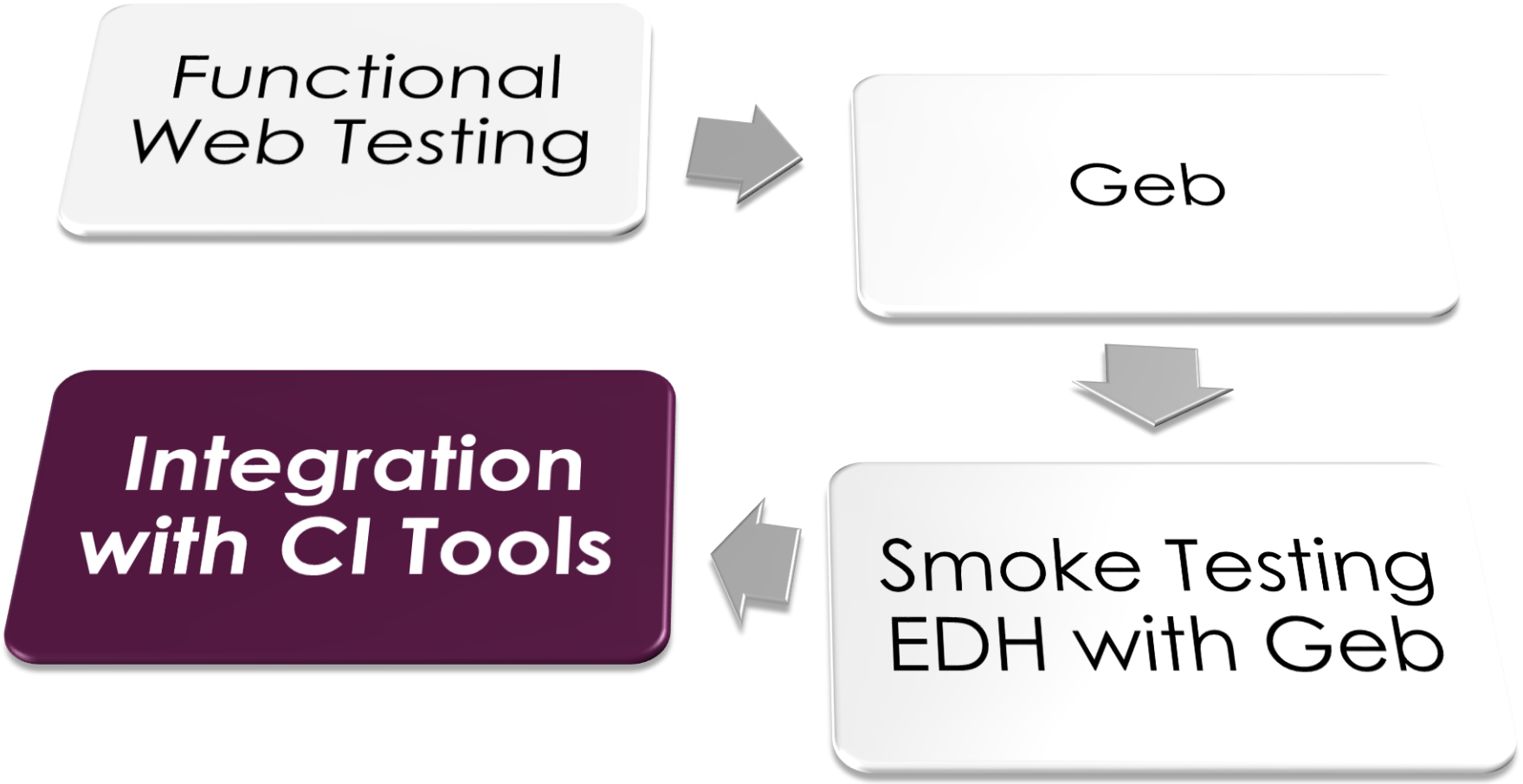


Smoke Testing EDH (3/3)

- Assert that random documents can be opened

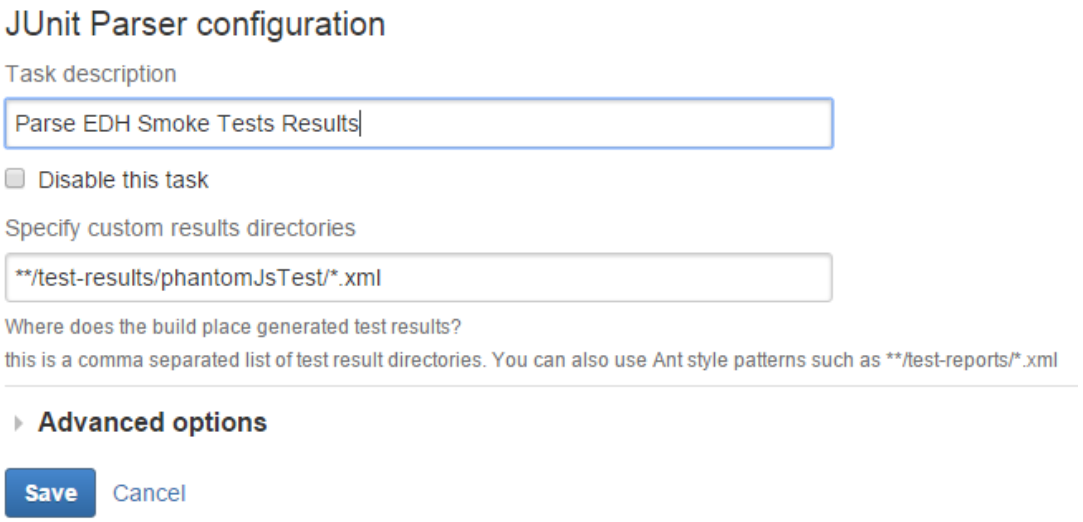


Agenda


Functional Web Testing
Geb
Smoke Testing EDH with Geb
Integration with CI Tools

Geb and CI Tools

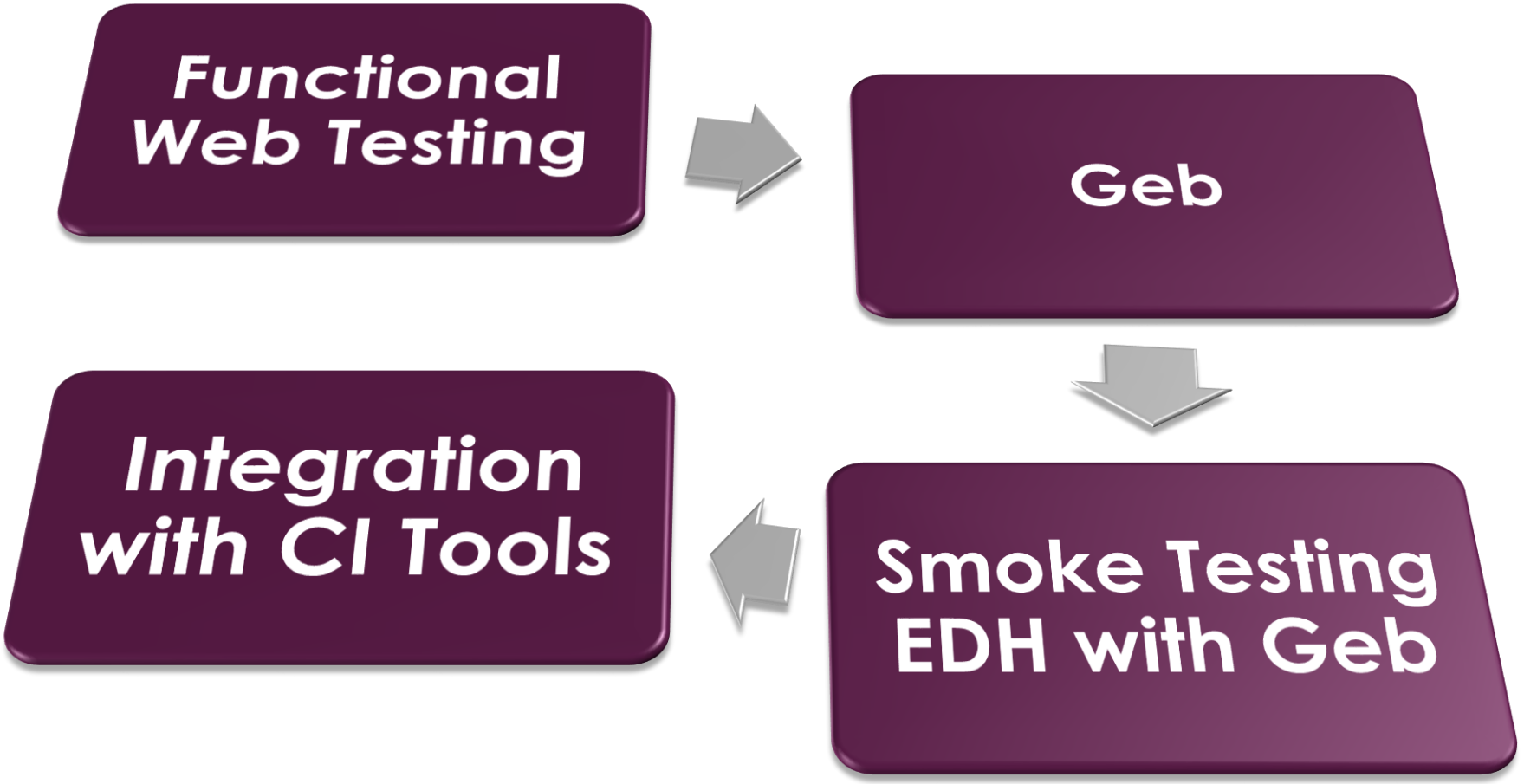
- Geb produces JUnit tests results in XML format
  - Easy to integrate with most CI Tools



Running Geb on a CI Server

- **Problem**
  - CI servers are headless
  - There is no display output for the browser to launch in
- **Solution**
  - Headless browser (e.g. PhantomJS)
  - Virtual framebuffers (e.g. Xvfb)
    - Launch the browser virtually
    - Emulate a graphical display

Agenda


Functional Web Testing
Geb
Smoke Testing EDH with Geb
Integration with CI Tools

Thank you.

@joaopcgsilva

joao.p.silva@cern.ch