Lukáš Fryč

Visual Testing of Browser Screen Captures using RushEye

Agenda

- Motivation
- Sample Application
- Present Tools
- Introducing RushEye
- Methods of Result Stabilization
- Automation
- Future

Motivation

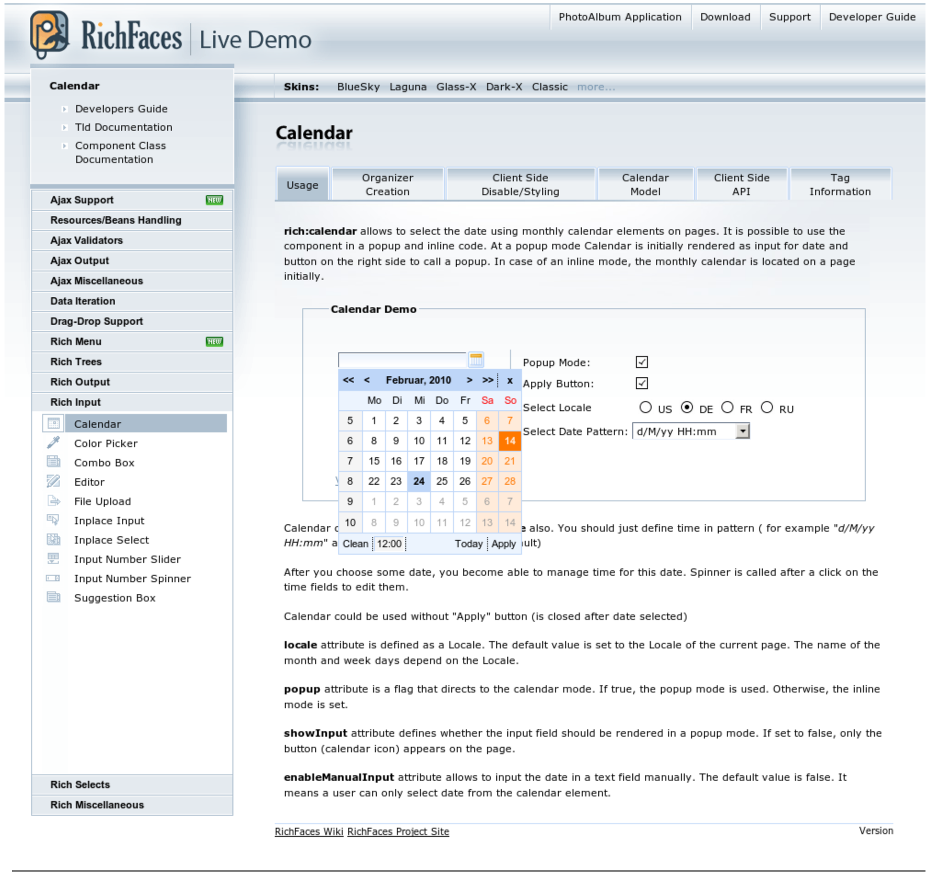
- Visual testing = Subset of manual testing
  - repetitious and so error-prone

- Visual testing can be automated
  - **Automated testing can be more extensive**
  - Testers may **focus on critical areas**

Sample Application

- RichFaces Live Demo
  - http://livedemo.exadel.com/richfaces-demo/index.jsp
  - 77 components
  - 12 skins
  - 3 major browsers (FF, IE7, IE8)
  - 77 * 12 * 3 = 2772

- Functional tests in place



Calendar

rich:calendar allows to select the date using monthly calendar elements on pages. It is possible to use the component in a popup and inline code. At a popup mode Calendar is initially rendered as input for date and button on the right side to call a popup. In case of an inline mode, the monthly calendar is located on a page initially.

Calendar Demo

Popup Mode: ✓
Apply Button: ✓
Select Locale: US [✓ DE] FR RU
Select Date Pattern: d/M/yy HH:mm

Calendar c[...] also. You should just define time in pattern (for example "*d/M/yy HH:mm*" a[...]ult)

After you choose some date, you become able to manage time for this date. Spinner is called after a click on the time fields to edit them.

Calendar could be used without "Apply" button (is closed after date selected)

locale attribute is defined as a Locale. The default value is set to the Locale of the current page. The name of the month and week days depend on the Locale.

popup attribute is a flag that directs to the calendar mode. If true, the popup mode is used. Otherwise, the inline mode is set.

showInput attribute defines whether the input field should be rendered in a popup mode. If set to false, only the button (calendar icon) appears on the page.

enableManualInput attribute allows to input the date in a text field manually. The default value is false. It means a user can only select date from the calendar element.

	Mo	Di	Mi	Do	Fr	Sa	So
5	1	2	3	4	5	6	7
6	8	9	10	11	12	13	14
7	15	16	17	18	19	20	21
8	22	23	24	25	26	27	28
9	1	2	3	4	5	6	7
10	8	9	10	11	12	13	14

Conditions of Sample App.

- Ready for capturing screen-shots
  - using Selenium
- Idea of automated screen-shot
  - generation
  - comparison
- The problem: existing screen-shot comparison tools

Real Needs

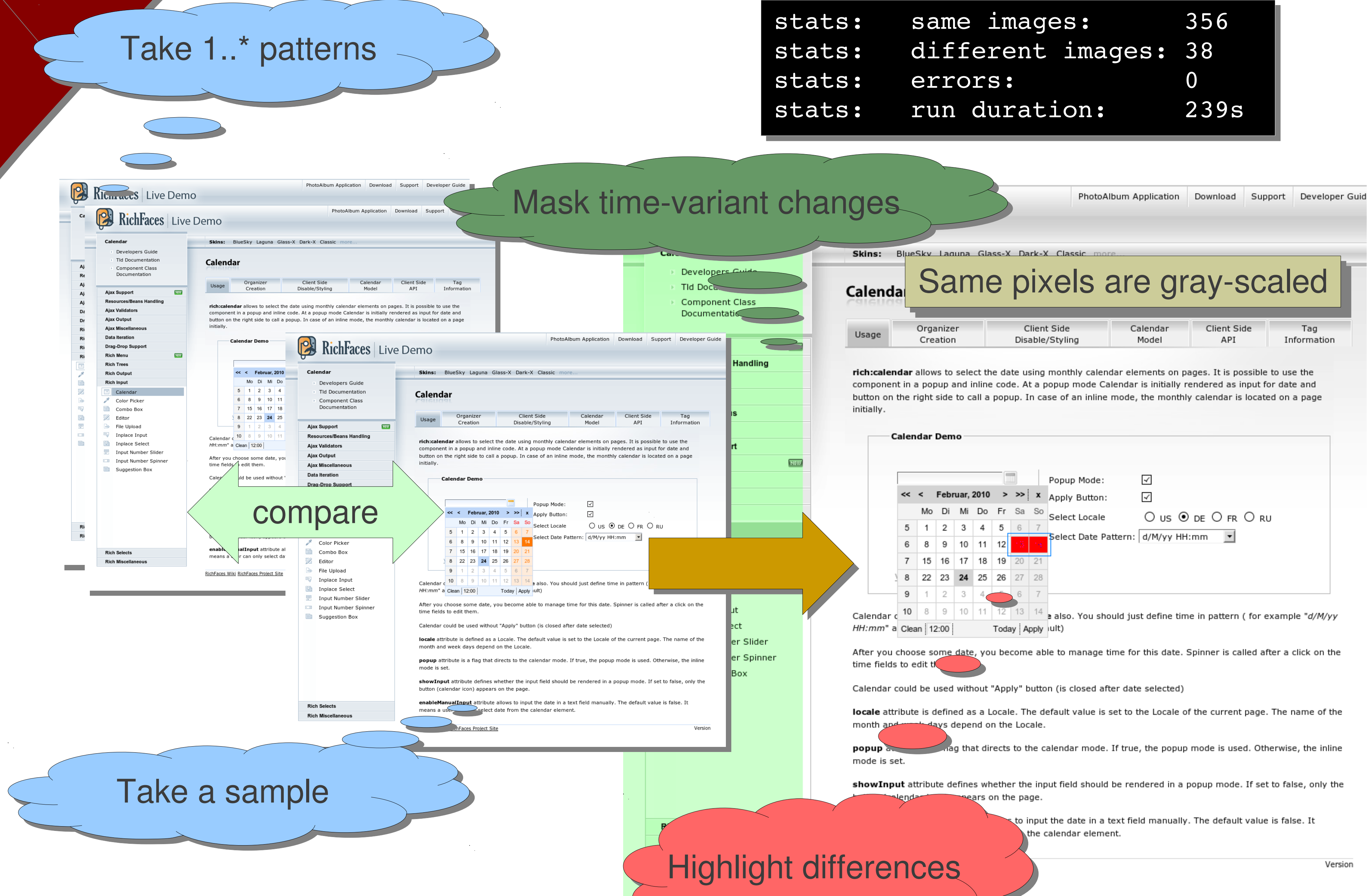
- Static Analysis of 2D images
- Filtering as much wrong hits as possible

Introducing RushEye

- **RushEye**
- is able to output **perceptual statistics**
- is able to process **filtering**

RushEye at Work

```
stats:   same images:      356
stats:   different images: 38
stats:   errors:           0
stats:   run duration:     239s
```

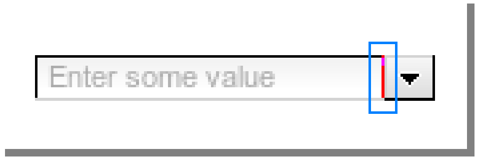


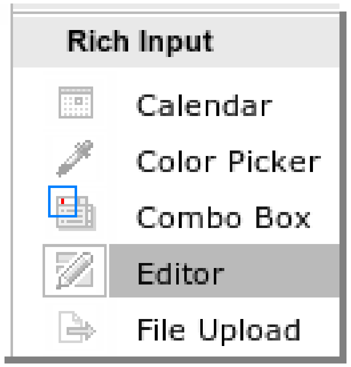
Perceptual Statistics

- How much was pixel in image changed?
  - locally
  - in global measure

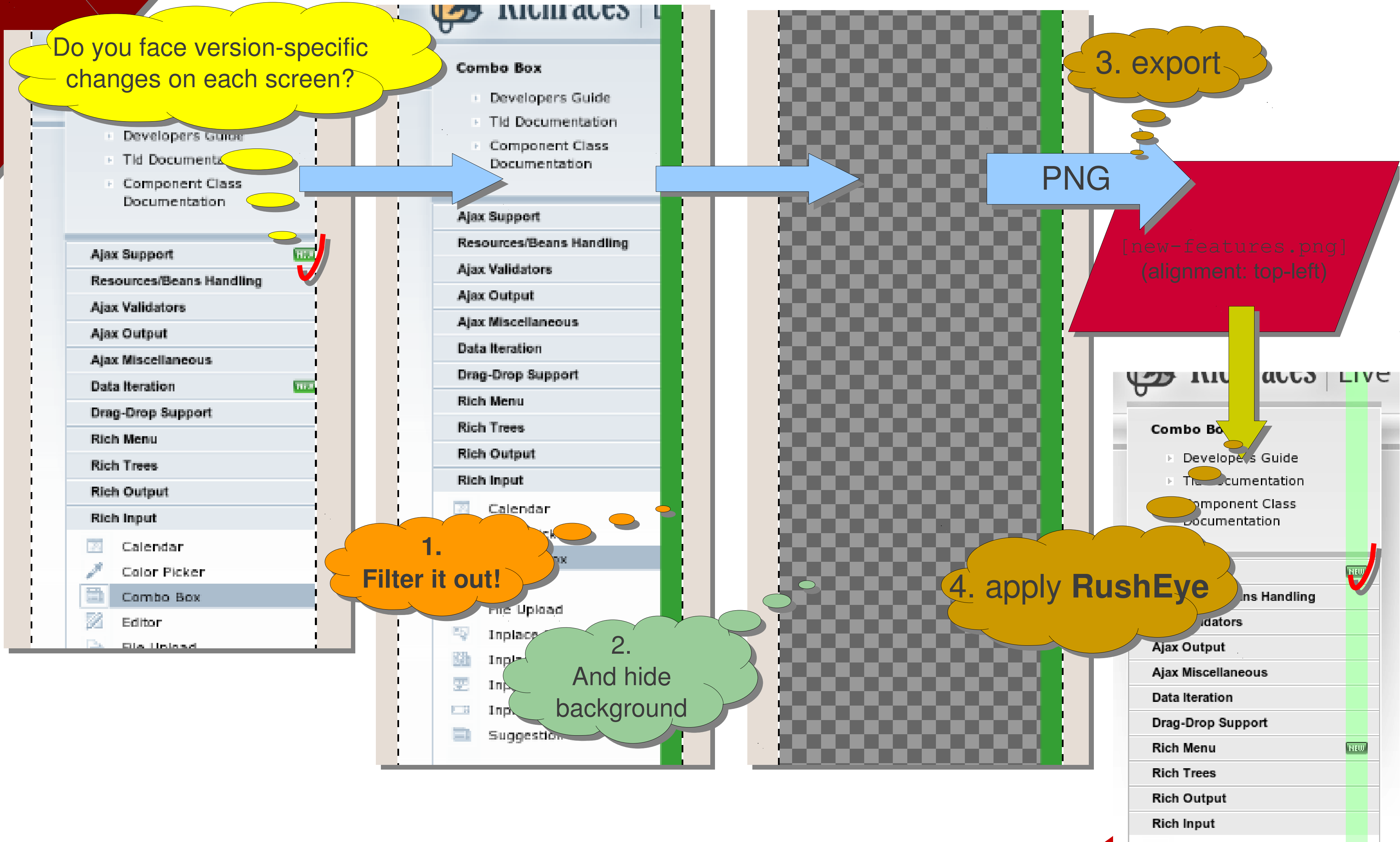
- Accepting images on basis of perceptual settings

- Examples:

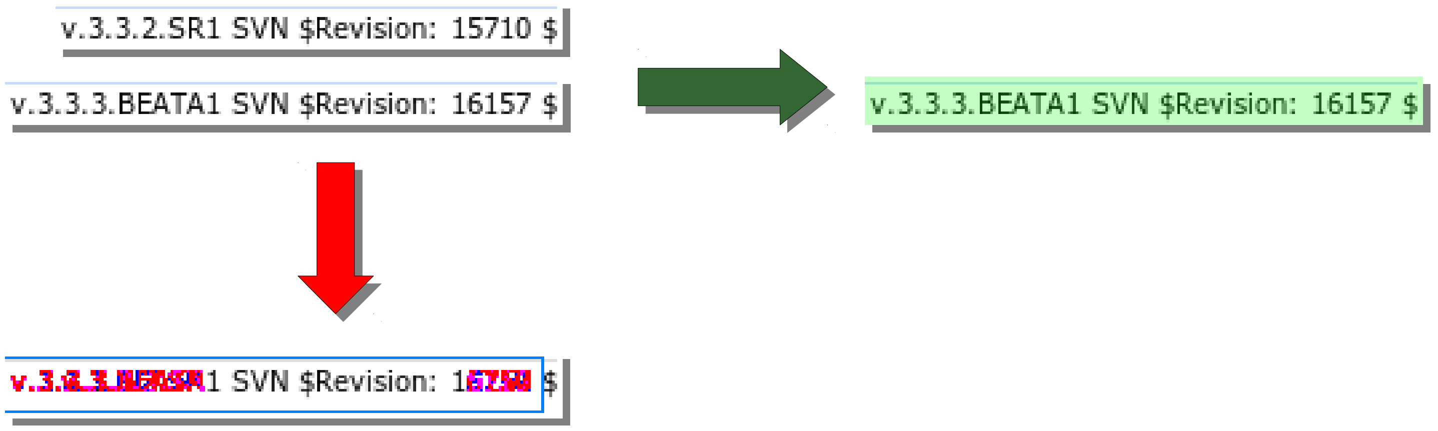




Selective-Alpha in Action



Ignore-Bitmap masks



v.3.3.2.SR1 SVN $Revision: 15710 $

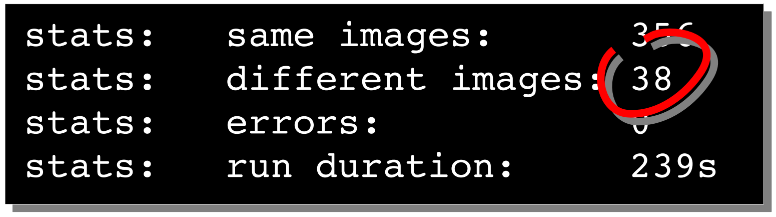
v.3.3.3.BEATA1 SVN $Revision: 16157 $

v.3.3.3.BEATA1 SVN $Revision: 16157 $

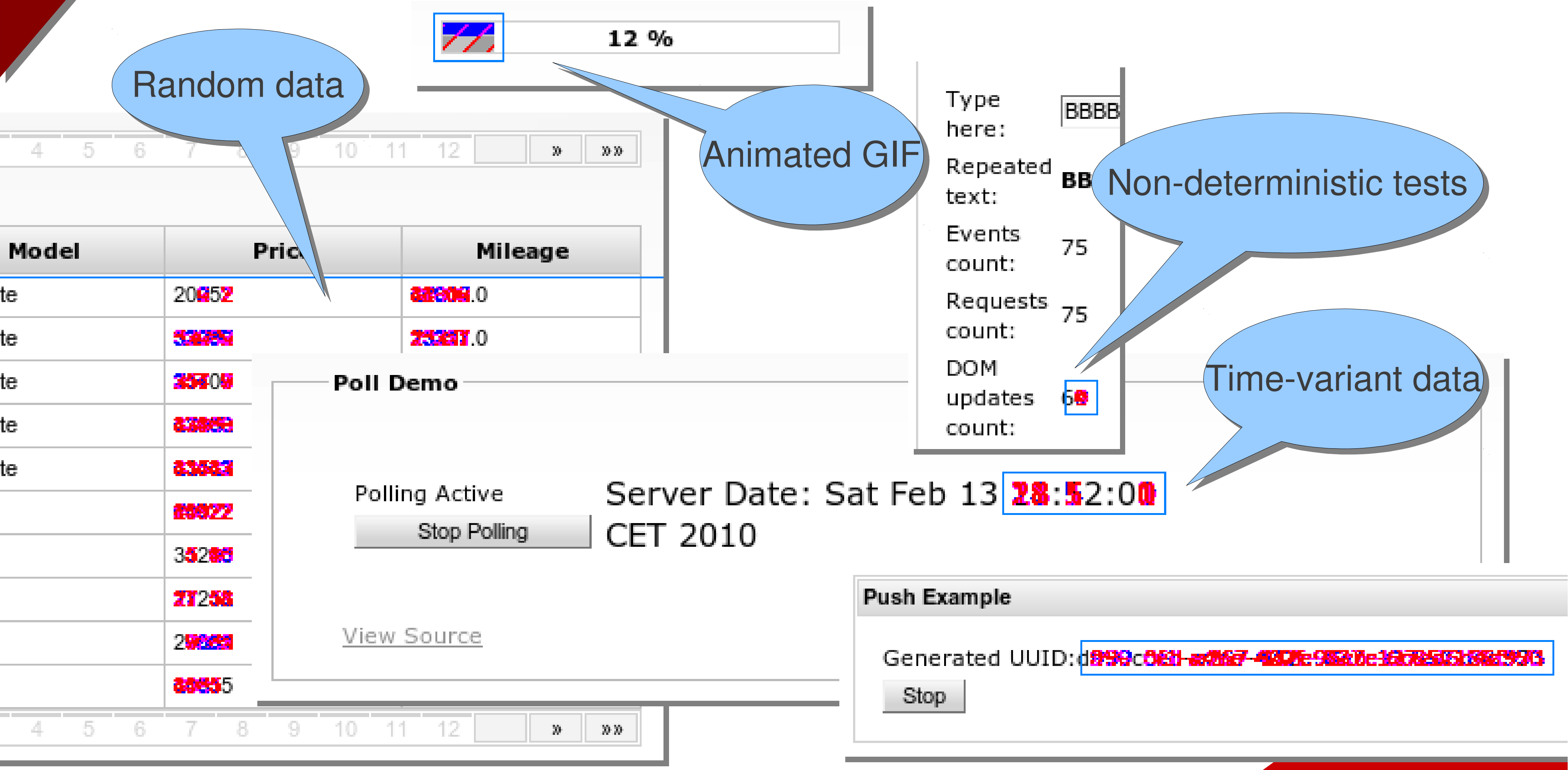
[illegible]1 SVN $Revision: 1[illegible] $

Failures

- But it isn't only successful



```
stats:    same images:          356
stats:    different images:     38
stats:    errors:               0
stats:    run duration:         239s
```



Model	Price	Mileage
te	20[illegible]	[illegible].0
te	[illegible]	[illegible].0
te	[illegible]	
te	[illegible]	
te	[illegible]	
	[illegible]	
	352[illegible]	
	[illegible]	
	2[illegible]	
	[illegible]5	

Type here:	BBBB
Repeated text:	BB
Events count:	75
Requests count:	75
DOM updates count:	6[illegible]

Poll Demo

Polling Active

Server Date: Sat Feb 13 [illegible]:[illegible]2:0[illegible] CET 2010

Push Example

Generated UUID: d[illegible]

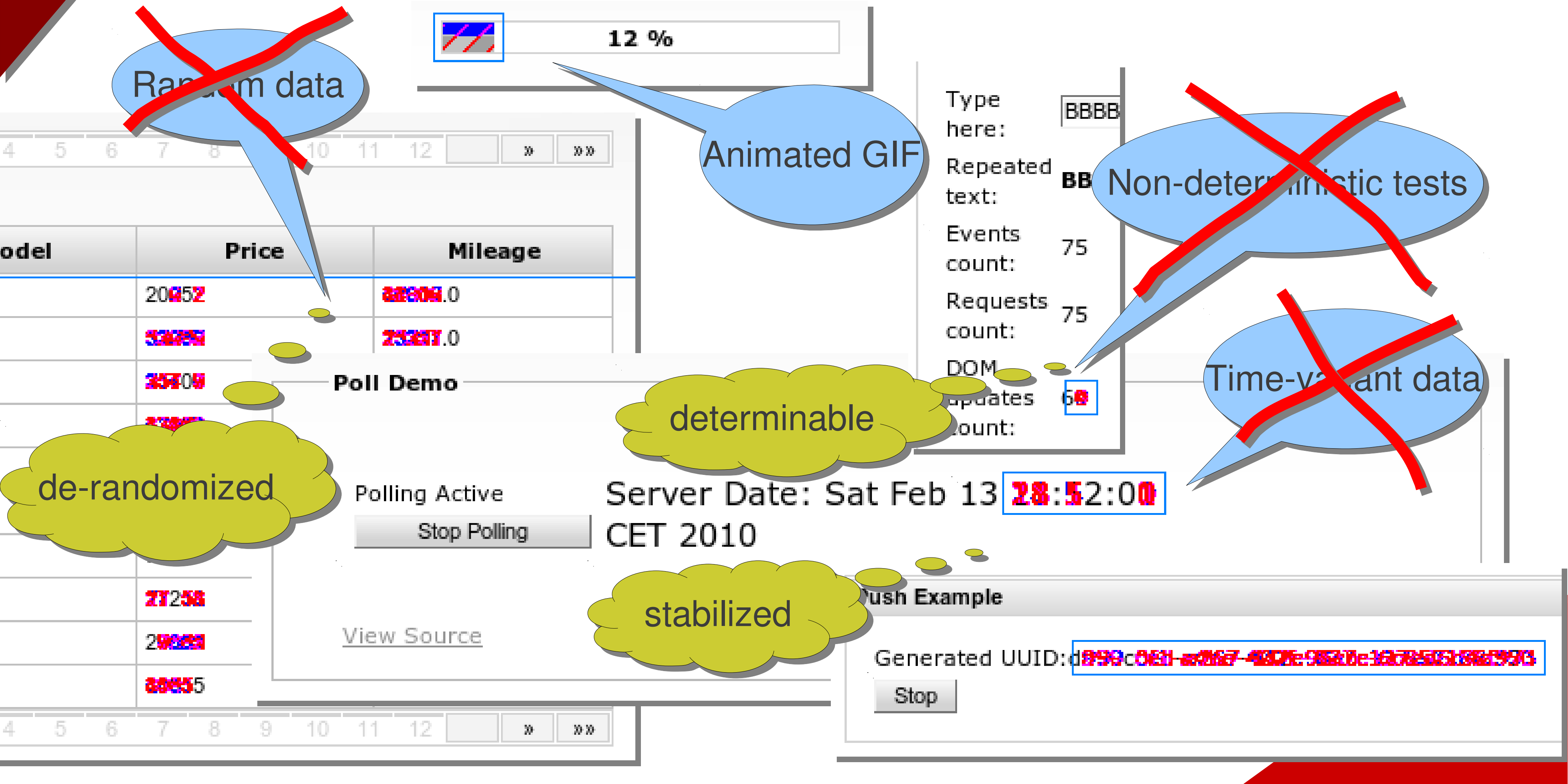
Time-Variant and Random Changes

- Redefining system methods or library features without changes to tested binaries

  - Using BootClassLoader
    - to redefine java.util.Random in order to get non-random data
    - to redefine System.currentTimeMillis()

Stabilized Results

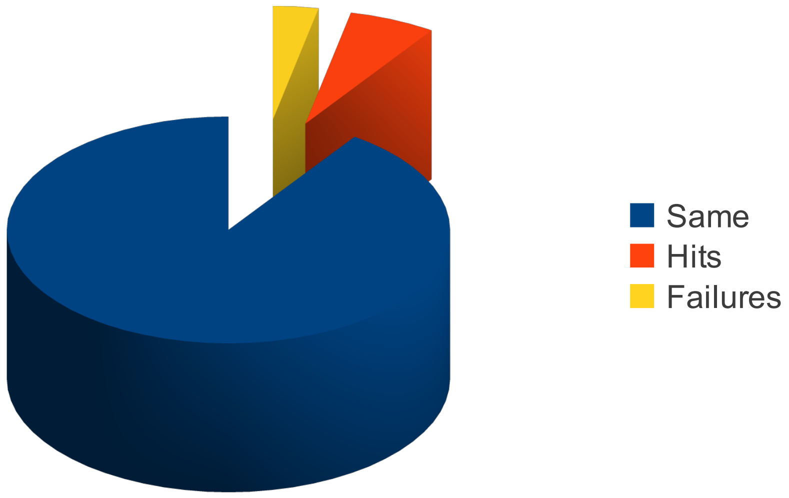
- The only problematic aspect is based on animations



Statistics

- Only **25** from **394** tests were identified as changes in underlying application

- **13** screens still wait for proper method of stabilization



How it works

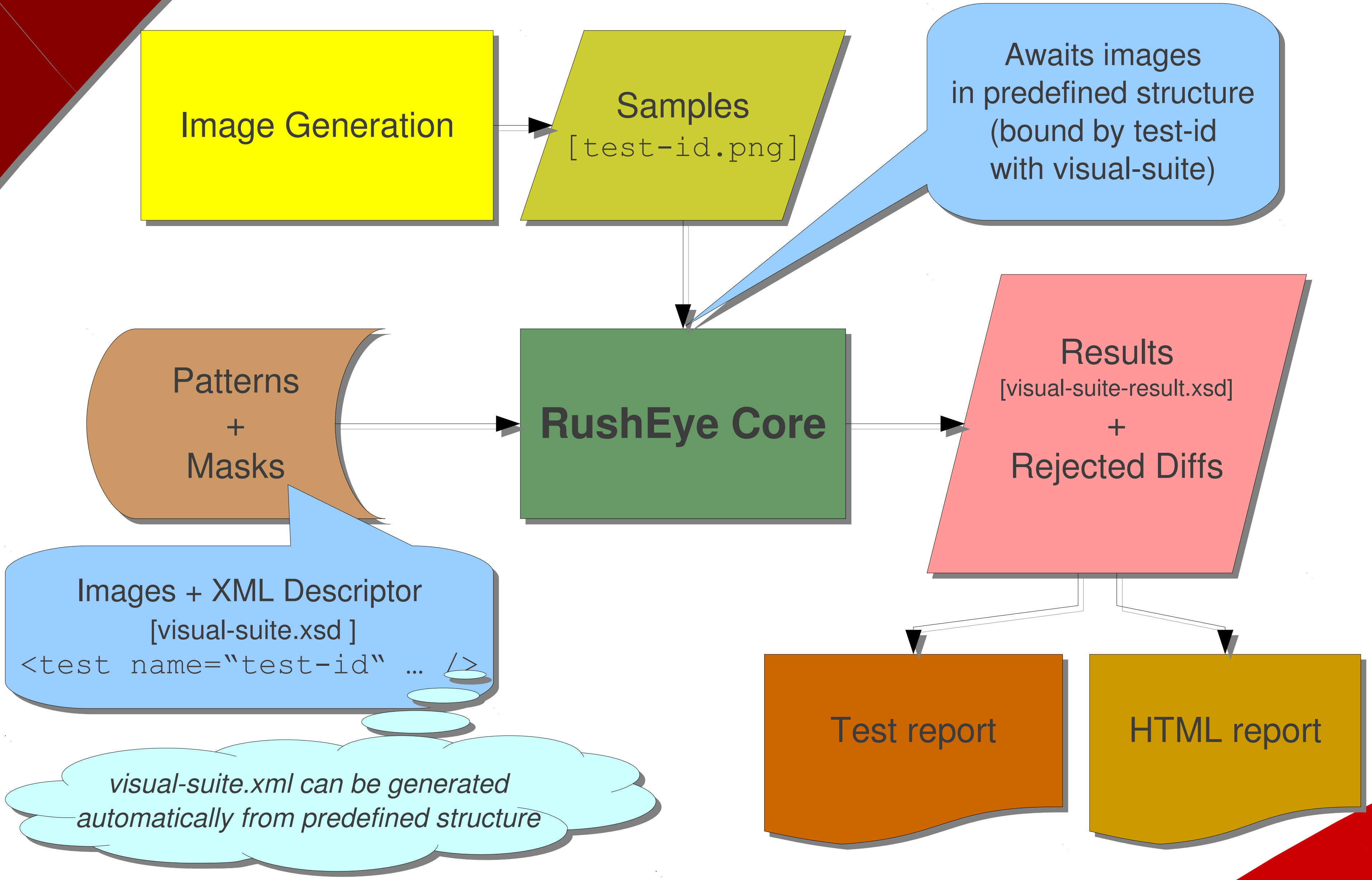
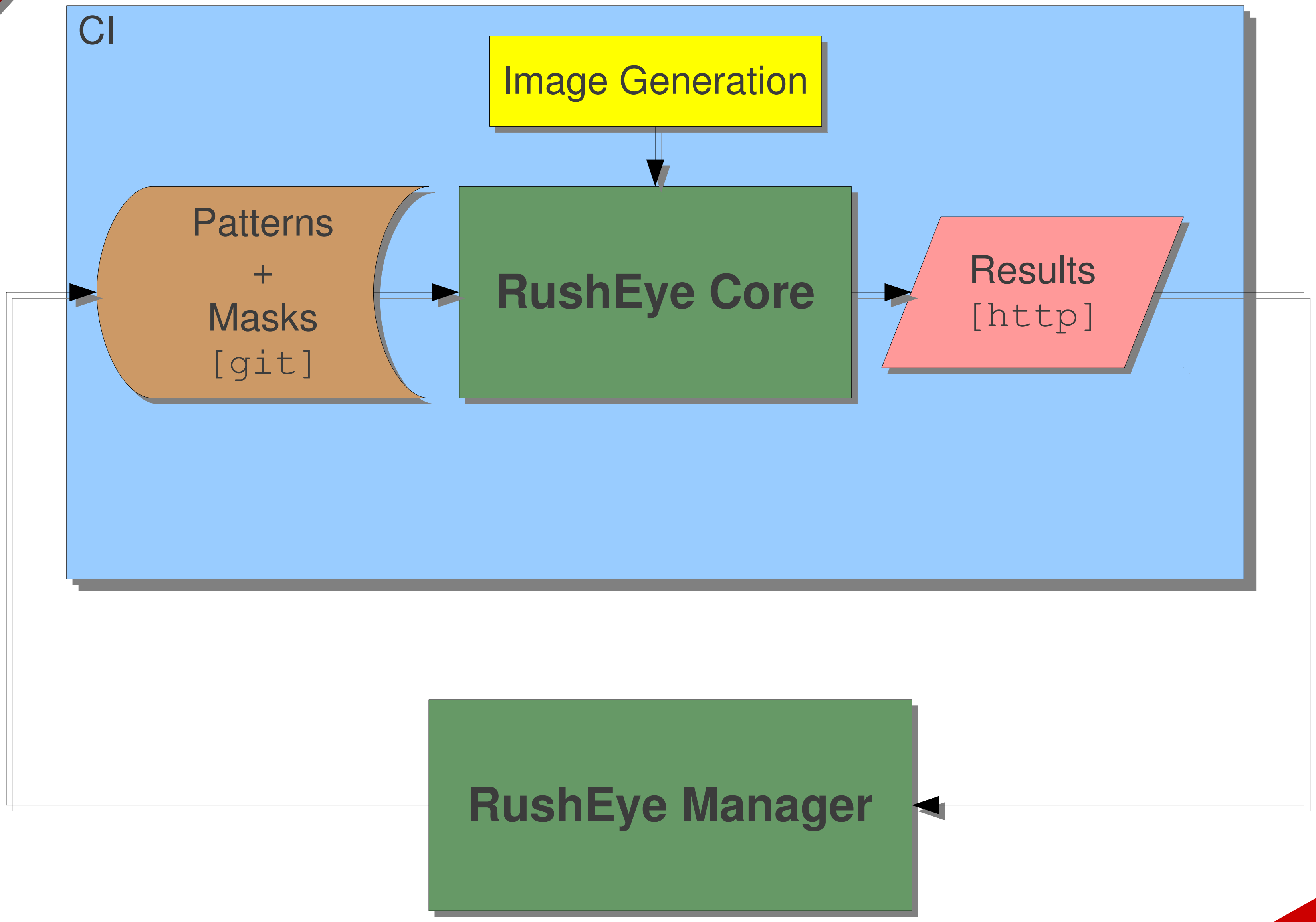


Image Generation
Samples
[test-id.png]
Awaits images in predefined structure (bound by test-id with visual-suite)
Patterns + Masks
RushEye Core
Results
[visual-suite-result.xsd]
+
Rejected Diffs
Images + XML Descriptor
[visual-suite.xsd]
<test name="test-id" … />
visual-suite.xml can be generated automatically from predefined structure
Test report
HTML report

Concept of Automation



Lukáš Fryč

Visual Testing of Browser Screen Captures using RushEye